UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Bauru

Fabricio Martins Batista

Aprendizado Profundo aplicado a Fenotipagem Automática na Piscicultura

Bauru, São Paulo, Brasil

6 de novembro de 2024

Fabricio Martins Batista

Aprendizado Profundo aplicado a Fenotipagem Automática na Piscicultura

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Instituto de Biociências, Letras e Ciências Exatas

Programa de Pós-Graduação em Ciência da Computação

Orientador: Dr. José Remo Ferreira Brega

Bauru, São Paulo, Brasil

6 de novembro de 2024

B333a

Batista, Fabricio Martins
Aprendizado Profundo aplicado a Fenotipagem Automática na Piscicultura / Fabricio Martins Batista. -- Bauru, 2024
97 p.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: José Remo Ferreira Brega

1. Fenotipagem. 2. Segmentação Semântica. 3. Visão Computacional. 4. Piscicultura. 5. Aprendizado Profundo. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE FABRICIO MARTINS BATISTA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.

Aos 21 dias do mês de agosto do ano de 2024, às 09:00 horas, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de FABRICIO MARTINS BATISTA, intitulada **Aprendizado Profundo aplicado a Fenotipagem Automática na Piscicultura**. A Comissão Examinadora foi constituida pelos seguintes membros: Prof. Dr. JOSE REMO FERREIRA BREGA (Orientador(a) - Participação Virtual) do(a) Departamento de Computação / UNESPCâmpus de Bauru, Prof. Dr. MARCELO DE PAIVA GUIMARÃES (Participação Virtual) do(a) Departamento de Realidade Virtual / Universidade Federal de São Paulo, Prof. Dr. KELTON AUGUSTO PONTARA DA COSTA (Participação Virtual) do(a) Departamento de Computação / UNESPCâmpus de Bauru. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final _____ APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

Prof. Dr. JOSE REMO FERREIRA BREGA

Prof. Dr. Jose Remo Ferreira Brega
Vice-diretor da Faculdade de Ciências
Unesp/Bauru

Resumo

A Piscicultura vem ganhando destaque nos últimos anos quase dobrando a produção mundial em um intervalo de dez anos (FAO, 2020). Os produtos da Piscicultura configuram uma importante fonte de proteína em países costeiros ou insulares pois são o recurso natural em maior abundância nessas regiões fazendo parte do cotidiano alimentar de suas populações. Os avanços tecnológicos recentes possibilitam a evolução da prática da Piscicultura através de novas ferramentas como a Inteligência Artificial e Internet das Coisas. Entre as práticas da chamada Piscicultura de Precisão estão o monitoramento inteligente de condições do ambiente através de sensores especializados (AGOSSOU; TOSHIRO, 2021) bem como a automação de tarefas manuais inerentes ao processo produtivo deste tipo de atividade. A automação torna possível a análise de grandes volumes de dados, trazendo escalabilidade ao manejo e reduzindos os custos do processo produtivo como um todo. Entre as tarefas que se destacam, está a fenotipagem de animais cultivados em cativeiro na Piscicultura durante vários estágios do crescimento para avaliação da interação da espécie com o ambiente ou mesmo a prevalência de determinadas características após sucessivas seleções reprodutivas. A fenotipagem geralmente é realizada manualmente através de uma imagem digital tirada por um especialista. A imagem obtida é pós-processada em softwares especializados em medição utilizando um objeto de referência de tamanho conhecido na imagem. Dessa forma, o presente trabalho foi desenvolvido com a finalidade de mapear técnicas de fenotipagem automática a partir de imagens digitais e a partir do levantamento propor uma solução através de Aprendizado Profundo. A finalidade do mapeamento está em obter uma visão geral sobre uma determinada área de pesquisa, e determinar quais e quantas são as evidências relacionadas a essa área. Os resultados demonstram uma tendência no uso de técnicas de Aprendizado Profundo para a automação dessas tarefas na área da Piscicultura. Todavia, ainda há uma prevalência de algumas técnicas de Visão Computacional clássica e Processamento de Imagens como a detecção de bordas e métodos iterativos para obtenção de regiões de interesse em imagens digitais e realizar a fenotipagem a partir das informações obtidas.

Keywords: Fenotipagem. Segmentação Semântica. Visão Computacional. Processamento de Imagens. Piscicultura. Inteligência Artificial. Aprendizado Profundo.

Abstract

Fish farming has been gaining prominence in recent years, almost doubling world production in a ten-year interval (FAO, 2020). Fish farming products are an important source of protein in coastal or island countries as they are the most abundant natural resource in these regions and are part of the daily food of their populations. Recent technological advances enable the evolution of fish farming practice through new tools such as Artificial Intelligence and the Internet of Things. Among the practices of the so-called Precision Fish Farming are the intelligent monitoring of environmental conditions through specialized sensors (AGOSSOU; TOSHIRO, 2021) as well as the automation of manual tasks inherent to the production process of this type of activity. Automation makes it possible to analyze large volumes of data, bringing scalability to handling, reducing costs of the production process as a whole. Among the tasks that stand out is the phenotyping of animals cultivated in captivity in Aquaculture during various stages of growth to evaluate the interaction of the species with the environment or even the prevalence of certain characteristics after successive reproductive selections. Phenotyping is usually performed manually using a digital image taken by a specialist and post-processed in specialized measurement software using an object of known size as a reference in the image. Thus, the present work was developed with the purpose automatic phenotyping techniques from digital images and based on the results obtained, propose a solution through the application of Deep Learning. The purpose of the mapping is to obtain an overview of a particular area of research, and to determine what and how much evidence is related to that area. The results demonstrate a trend in the use of Deep Learning techniques for the automation of these tasks in the fish farming area in recent years, but there is still a prevalence of some classical Computer Vision and Image Processing techniques such as edge detection and iterative methods to obtain regions of interest in digital images.

Keywords: Computer Vision. Phenotyping. Acquaculture. Artificial Intelligence. Deep Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACM	*Association of Computing Machinery*
IoT	*Internet of Things*
IA	Inteligência Artificial
ML	*Machine Learning*
CNN	*Convolutional Neural Networks*
Mask R-CNN	*Masked Region-based Convolutional Neural Networks*
RGB	*Red-Green-Blue*
CIELAB	*Commission Internationale de l´Eclairage* L*a*b
Blob	*Binary Large Object*
MLP	*Multi-Layer Perceptron*
CPU	*Central Processing Unit*
GPU	*Graphical Processing Unit*
OpenGL	*Open Graphical Language*
CUDA	*Compute Unified Device Architecture*
GPGPU	*General Purpose Graphical Processing Unit*
ReLU	*Rectified Linear Unit*
ILSVRC	*Imagenet Large Scale Visual Recognition Challenge*
VGG	*Visual Geometry Group*
NiN	*Network in Network*
Pascal VOC	*Pattern Analysis Statistical Modelling and Computational Learning Visual Object Classification*
HOG	*Histogram of Oriented Gradients*
SIFT	*Scale Invariant Feature Analysis*
SVM	*Support Vector Machines*

R-CNN	*Region-based Convolutional Neural Networks*
SPPNet	*Spatial Pyramid Pooling Network*
Fast R-CNN	*Fast Region-based Convolutional Neural Networks*
RoI Pooling	*Region of Interest Pooling*
RPN	*Region Proposal Network*
NMS	*Non-Max Supression*
IoU	*Intersection over Union*
RoI Align	*Region of Interest Align*
RSL	Revisão Sistemática da Literatura
PICOC	População, Intervenção, Comparação, Saídas e Contexto (do inglês, *Population, Intervention, Comparison, Outcomes, Context*)
IEEE	*Institute of Electrical Engineering and Eletronics*
NPU	*Neural Processing Unit*
FP	*Floating Point*
RAM	*Random Access Memory*
MSE	*Mean Squared Error*
tl	*Total Length*
th	*Total Height*
hh	*Head Height*
hl	*Head Length*
sl	*Standard Length*
pl	*Pelvis Length*

Sumário

1 Introdução

A produção mundial de aquicultura cresceu em média 5,3% ao ano no período 2001-2018, consistindo em 82,1 milhões de toneladas (US$ 250,1 bilhões) de animais aquáticos em 2018. Na América Latina, a aquicultura avançou significativamente na posição global (FAO, 2020). O principal grupo de peixes nativos produzidos no Brasil é composto por espécies de Serrasalmídeos (tambaqui *Colossoma macropomum*, pacu *Piaractus mesopotamicus*, pirapitinga *Piaractus brachypomys* e seus híbridos interespecíficos), correspondendo a cerca de 30% da produção nacional, estimada em 529,6 mil toneladas em 2019 (IBGE, 2020)

Novas tecnologias se fazem necessárias para apoiar o crescente volume de produção de pescado e a demanda por consumo, particularmente em termos de segurança alimentar tema extremamente relevante considerando o cenário geopolítico atual. Portanto, abordagens de ponta na aquicultura podem acelerar o progresso em direção a uma maior produtividade e sistemas de produção sustentáveis (ROSE; CHILVERS, 2018).

Nos últimos anos, a aplicação de Inteligência Artificial vem ganhando destaque em diversas áreas do conhecimento. Os avanços em tecnologias e o rápido desenvolvimento de dispositivos eletrônicos, sensores e conectividade de internet deram origem ao conceito de "Agricultura de Precisão" que tem se expandido rapidamente com automação de manejo, controle de colheita de alta precisão, coleta de informações além de outras aplicações (BONGIOVANNI; LOWENBERG-DEBOER, 2004).

No contexto da Piscicultura, figuram aplicações de monitoramento de indicadores da qualidade da água como Oxigênio dissolvido, temperatura, turbidez, cor da água, pH, dióxido de carbono, alcalinidade, condutividade elétrica, sólidos totais dissolvidos, amônia não sindicalizada, nitrato, nitrito entre outros (AGOSSOU; TOSHIRO, 2021).

Além das aplicações de Internet das Coisas (do inglês *Internet of Things* - IoT), destacam-se também o uso de Visão Computacional, Processamento de Imagens e Inteligência Artificial com diversas aplicações como detecção de doenças (MALIK; KUMAR; SAHOO, 2017) (WALEED et al., 2019) (YASRUDDIN et al., 2022) (DARAPANENI et al., 2022), classificação de espécies (CAO et al., 2021) estimativa de qualidade (TAHERI-GARAVAND et al., 2019), contagem e a fenotipagem (extração de características do trato físico de animais) (WANG et al., 2019a), (NGUYEN et al., 2020). A automação do processo de fenotipagem torna possível a análise de grandes volumes de dados trazendo escalabilidade ao manejo, reduzindo custos do processo produtivo como um todo.

A fenotipagem pode ser definida como o processo de extração de medidas específicas de determinados animais como a pesagem ou medição de regiões específicas de interesse como tamanho da cabeça, comprimento total ou altura. Estes dados são utilizados por

pesquisadores ou produtores para avaliação da interação da espécie com o ambiente ou mesmo a prevalência de determinados características após sucessivas seleções reprodutivas (BLAY et al., 2021), (LEEDS et al., 2016).

A seleção reprodutiva tem por objetivo identificar e selecionar indivíduos que possuem características desejáveis como resistência a doenças ou peso total atingido após a maturação. Através desse processo é possível realizar um refinamento genético com os indivíduos (LHORENTE et al., 2019).

A extração de informações do trato físico de animais geralmente é feita através da captura de uma imagem digital tirada por um especialista. Então, esta imagem digital é pós-processada manualmente em softwares especializados em medição utilizando um objeto de referência de tamanho conhecido na imagem (ANDRIALOVANIRINA et al., 2020).

O presente estudo foi criado com o objetivo de apresentar uma Revisão Sistemática da Literatura (RSL) sobre a utilização de técnicas computacionais para a fenotipagem de animais na Piscicultura. A finalidade do mapeamento está em obter uma visão geral sobre esta área de pesquisa, e determinar quais e quantas são as evidências relacionadas a essa área. A RSL é destinada a temas específicos e por isso, foi a abordagem escolhida para um dos capítulos deste trabalho.

Este trabalho tem por objetivo investigar quais as abordagens utilizadas recentemente na automação da fenotipagem de animais através de uma imagem digital. Assim, o foco foi encontrar trabalhos na literatura que tivessem a aplicação de técnicas computacionais de fenotipagem a partir de imagens digitais. Após a investigação foi selecionada a técnica de segmentação de imagens Mask R-CNN (HE et al., 2017) para realizar a fenotipagem de duas espécies diferentes de peixes: o Colossoma Macropomum (Tambaqui) e o Piaractus Mesopotamicus (Pacu).

A disposição deste trabalho foi realizada da seguinte forma: além desta introdução, no Capítulo 2 Fundamentação Teórica são apresentados os principais conceitos relacionados a processamento de imagens de maneira a contextualizar as técnicas apresentadas; o Capítulo 3 Redes Neurais Convolucionais, a partir dos conceitos apresentados no capítulo anterior, são discutidos os blocos e operações elementares que compõe as Redes Neurais Convolucionais; o Capítulo 4 Arquiteturas de Redes Neurais Convolucionais é dedicado a um levantamento histórico em ordem cronológica das arquiteturas mais relevantes da literatura, no contexto desse trabalho, bem como suas contribuições para as técnicas de Visão Computacional modernas; no Capítulo 5 Segmentação de Imagens e Mask R-CNN é apresentado de forma mais detalhada o problema de Segmentação Semântica e a evolução da família de arquiteturas de segmentação baseada em regiões, as R-CNNs; no Capítulo 6 Revisão Sistemática da Literatura (RSL) é apresentada a metodologia de pesquisa da literatura e os resultados obtidos através da RSL; no Capítulo 7 Experimentos

e Resultados é apresentada a abordagem proposta e a discussão relacionada ao tema, com base nos experimentos conduzidos. Por fim o Capítulo 8 Conclusão conclui o texto com uma discussão sobre a relevância da abordagem proposta para a prática de Fenotipagem com aplicações a Seleção Reprodutiva na Piscicultura.

2 Fundamentação Teórica

Neste capítulo estão apresentados os principais conceitos relacionados a este estudo. O primeiro conceito é o de Visão Computacional que pode ser definido como uma área da Computação Gráfica que tem por objetivo modelar e utilizar sistemas capazes de processar e extrair informações relevantes a partir de imagens ou vídeos digitais de maneira análoga ao sistema de processamento visual humano. Neste capítulo, são revisitados alguns conceitos elementares da área de Visão Computacional e os conceitos da área de Genética relacionados a este trabalho.

2.1 Fenotipagem

A fenotipagem pode ser definida como o processo de extração de medidas específicas de determinados animais como a pesagem ou medição de regiões específicas de interesse como tamanho da cabeça, comprimento total ou altura. Estes dados são utilizados por pesquisadores ou produtores para avaliação da interação da espécie com o ambiente ou mesmo a prevalência de determinados características após sucessivas seleções reprodutivas (BLAY et al., 2021), (LEEDS et al., 2016).

A automação do processo de fenotipagem torna possível a análise de grandes volumes de dados trazendo escalabilidade ao manejo, reduzindo custos do processo produtivo como um todo.

A extração de informações do trato físico de animais geralmente é feita através da captura de uma foto digital tirada por um especialista. Esta foto digital é pós-processada manualmente em softwares especializados em medição utilizando um objeto de tamanho conhecido como referência na imagem (ANDRIALOVANIRINA et al., 2020).

Esse tipo de processo é manual e bastante custoso, estando sujeito a viéses dos pesquisadores responsáveis por realizar as medições. Como o volume de animais pode ser bastante grande, realizar esse tipo de processo manualmente se torna inviável em aplicações comerciais.

Conforme a tendência do uso de técnicas de aprendizado profundo aplicadas a problemas de fenotipagem na Psicultura conforme observado no Capítulo 6. Nesse trabalho foi discutida a aplicação de um método de segmentação de imagens aplicado ao problema de fenotipagem.

Entre as medidas obtidas pelo sistema construído estão a altura total (th), comprimento total (tl), comprimento padrão (sl), largura da pelve (pl), altura da cabeça (hh), largura da cabeça (hl), além da área em pixels de cada região obtida conforme a Figura 1

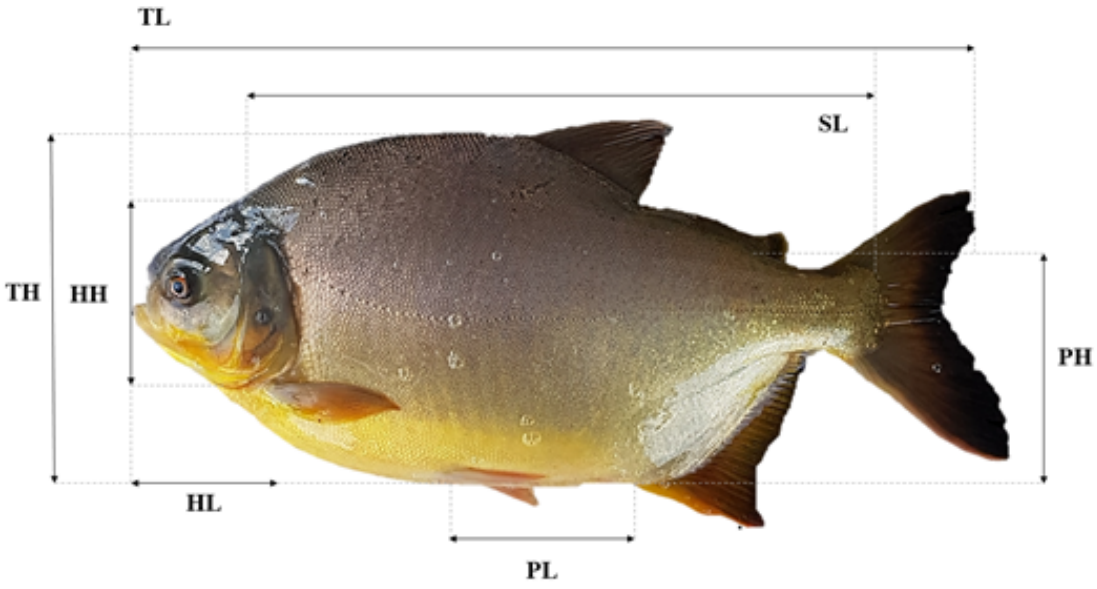


Figura 1 – Medidas a serem obtidas pelo sistem proposto neste trabalho.

2.2 Seleção Reprodutiva

A seleção reprodutiva, por sua vez, tem como objetivo identificar e selecionar indivíduos que possuem características desejáveis como resistência a doenças ou peso total atingido após a maturação. Através desse processo é possível realizar um refinamento genético com os indivíduos (LHORENTE et al., 2019).

A seleção reprodutiva para melhoramento genético se concentra basicamente na escolha de indíviduos que atingem pesos maiores ou que possuem alto ganho diário de peso. Apesar disso, no contexto da Piscicultura comercial no Brasil o melhoramento genético ainda não é realizado devido as dificuldades e o intenso trabalho manual necessário para realização desses programas em larga escala. Para resolver este problema, é necessário a viabilização de maneiras de eliminar parte do trabalho manual envolvido no processo.

Este trabalho apresenta um método para alcançar esse objetivo e coletar informações granulares dos indíviduos da espécie para incorporação das informações em programas de melhoramento genético das espécies *Piaractus Mesopotamicus (Pacu)* e *Colossoma macropomum (Tambaqui)*.

2.3 Segmentação Semântica

A tarefa de classificação consiste em atribuir uma ou mais categorias a uma imagem. Já a detecção de objetos consiste em localizar e classificar um objeto presente na imagem de entrada estimando as coordenadas de uma caixa delimitadora que contenha o objeto. Para o processo de fenotipagem apresentado neste trabalho, uma outra tarefa é necessária, além da detecção de objetos: a classificação dos pixels pertencentes ao objeto detectado. Através da área em pixels obtida, é possível estimar outras medidas interessantes além das medidas descritas em (FREITAS et al., 2023) como por exemplo o peso dos subprodutos obtidos a partir dos animais como a costela e o lombo (ARIEDE et al., 2023a). Sob esta ótica, a segmentação pode ser vista como um problema de classificação aplicada pixel a pixel na imagem. A Figura 2 ilustra a tarefa de segmentação.

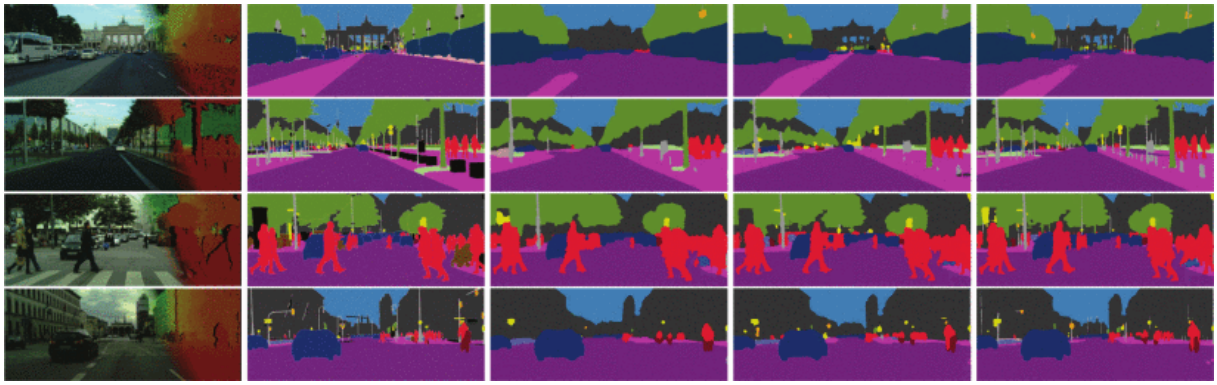


Figura 2 – Exemplo do conjunto Cities Dataset - Fonte: (CORDTS et al., 2016)

A R-CNN é um método de predição de múltiplas etapas que pode ser utilizado para segmentação de imagens. As etapas são: seleção, extração e classificação. O nome R-CNN se deve a divisão das imagens em regiões, do inglês *Region-based Convolutional Neural Networks*. Cada uma dessas regiões então é redimensionada para ser compatível com as dimensões de entrada de CNNs pré-treinadas no ImageNet. Já a detecção da caixa delimitadora dos objetos é modelada como um problema de regressão a partir dos mapas de características extraídos a partir das regiões iniciais.

A Mask R-CNN (HE et al., 2017) é um método que realiza todas as tarefas descritas anteriormente através de uma função multi-objetivo que é otimizida simultaneamente durante o treinamento da arquitetura. Além disso, a Mask R-CNN evita o alto custo computacional requerido pela operação de busca seletiva, do inglês *Selective Search* (UIJLINGS et al., 2013) presente na R-CNN através de uma RPN *Region Proposal Network*.

2.4 Imagens Digitais e Espaços de Cores

Em processamento de sinais uma imagem digital pode ser definida através de uma estrutura bi-dimensional. A Figura 3, apresenta uma imagem e o ponto de origem (x = 0; y = 0) de uma imagem digital. Mais dimensões podem ser utilizadas na representação de uma imagem digital a depender do espaço de cores escolhido para representação da imagem.

Figura 3 – Estrutura bi-dimensional da imagem

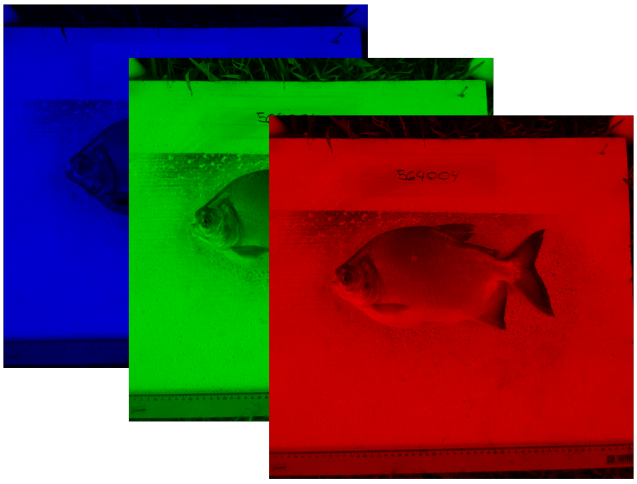
Fonte - Elaborado pelo autor

Uma imagem em escala de cinza necessita de apenas um valor por pixel (ou canal) que se refere a quão próximo o valor está do limite inferior, com o 0 correspondendo a cor preta ou do limite superior 255 cor branca. Já para a representação de cores em imagens digitais é necessário estabelecer um espaço de cores. Um espaço de cores amplamente utilizado é o RGB *(Red-Green-Blue)*. As divisões no espaço de cores RGB são chamadas canais. Essa representação consiste em um modelo aditivo de 3 cores elementares para formação de um espectro cromático amplo.

Assim, a imagem pode ser representada por uma estrutura bidimensional replicada 3 vezes. Cada uma das 3 camadas representa uma cor elementar, também referida como canal. A cor final de cada pixel é calculada através da combinação dos pixels de cada

camada conforme apresentado na Figura 4.

Figura 4 – Canais RGB



Fonte - Elaborado pelo autor

Existem outros espaços de cores que podem ser úteis a depender da análise pretendida como o CIELAB (L*a*b) (ROBERTSON, 1977), bastante popular na análise de qualidade de alimentos (MENDOZA; DEJMEK; AGUILERA, 2006),(DU; SUN, 2005),(JHA; CHOPRA; KINGSLY, 2007).

2.5 Processamento de Imagens

O Processamento de Imagens digitais consiste em aplicar operações matemáticas e uma imagem com o objetivo de torná-la mais adequada ao tipo de análise desejada (SZELISKI, 2022).

Dessa forma, é possível realçar ou filtrar determinadas características da imagem que possam nos auxiliar a cumprir determinada tarefa como: classificar uma imagem, localizar um objeto, detectar bordas ou regiões de interesse. As operações matemáticas mencionadas, também são referidas na literatura como filtros e podem ser classificadas em lineares e não lineares.

2.5.1 Operadores Pixel a Pixel

Uma das operações mais simples possível é a aplicação de uma operação matemática em cada pixel. O valor final do pixel após a transformação depende apenas do valor original do pixel (SZELISKI, 2022).

Tais operações são denominadas operadores de ponto em Visão Computacional. Esse tipo de transformação possibilita a alterar a cor ou luminosidade dos pixels de uma

determinada imagem.

2.5.2 Operadores Baseados em Área

Outro conjunto de operações possíveis são os Operadores Baseados em Área. Nesta classe de operadores, o valor final do pixel depende de um conjunto de pixels relacionados de alguma forma ao pixel sendo processado. Os pixels podem se relacionar um ao outro na imagem através da adjacência ou distância entre o pixel sendo processado e sua vizinhança (SZELISKI, 2022).

Um representante deste conjunto de operações é a equalização de histograma de uma imagem (PIZER et al., 1987). O objetivo desse tipo de operação é ajustar o contraste de uma imagem através da redistribuição das intensidades de cada pixel usando o cálculo do histograma de cada um dos canais de cores, e então ajustar a intensidade de cada um desses pixels para estar dentro de um determinado intervalo de frequências calculado através do histograma.

2.5.3 Filtros

Além do ajuste de contraste possível através das transformações baseadas em área, um outro conceito fundamental em Visão Computacional é a filtragem. Entre os exemplos de filtros estão o esmaecimento, realçamento de detalhes, detecção de bordas e cantos entre outros. A filtragem pode ser dividida em 2 categorias, filtragem linear e não linear.

2.5.3.1 Filtros Lineares

A filtragem linear se trata da aplicação de uma operação matemática chamada de convolução entre uma imagem e um filtro também referido na literatura como kernel ou máscara. A representação matemática de um filtro é realizada através de uma matriz de valores.

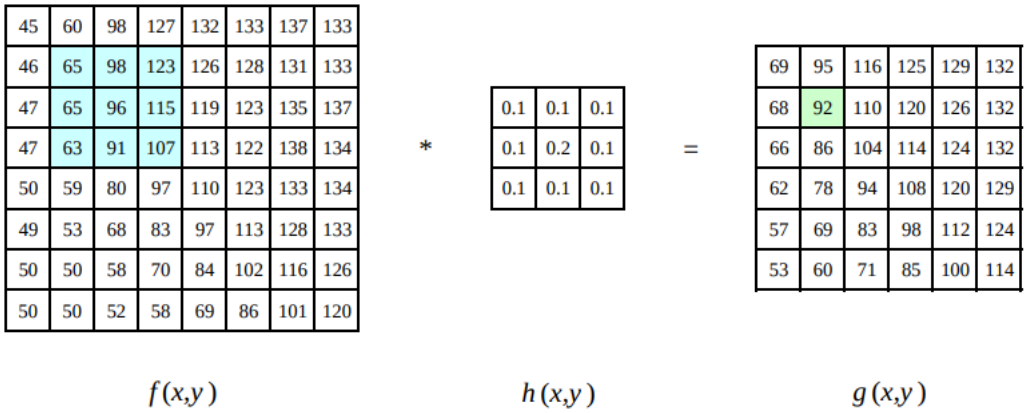
A operação de convolução é definida como a superposição do filtro em cada pixel da imagem a ser processada a fim de obter um novo valor de saída para o pixel através da ponderação e soma dos valores dos pixels de uma pequena vizinhança do pixel sendo processado (SZELISKI, 2022).

2.5.3.2 Filtros Não-lineares

A filtragem não-linear é similar a filtragem linear exceto que ela não pode ser implementada através de uma convolução. Um representante deste tipo de operação é a filtragem baseada na mediana. A Figura 5 ilustra uma operação de convolução.

A filtragem não-linear tem aplicações em remoção de ruídos e é uma importante ferramenta na área de Processamento de Imagens (SZELISKI, 2022). A ideia central de

Figura 5 – Operação de Convolução



Fonte: (SZELISKI, 2022)

um filtro não-linear é que ele tende a se adaptar a região em que está sendo aplicado, portanto, muda constantemente conforme o processamento avança.

2.5.4 Detecção de Bordas

A detecção de bordas está relacionada a forma como o mundo visual é percebido a nossa volta. Através das informações de bordas dos objetos é possível identificar objetos além da percepção de profundidade.

A detecção de bordas é uma importante tarefa de Visão Computacional e pode ser implementada através da filtragem linear apresentada no item 2.5.3.1. Uma borda pode ser definida como uma abrupta alteração de intensidade em uma pequena região da imagem.

Matematicamente, se faz necessária uma operação capaz de quantificar a taxa de mudança de intensidade dos pixels.

No cálculo, a derivada de primeira ordem é uma operação capaz de identificar a taxa de mudança de determinada função. A imagem sendo uma estrutura bidimensional, a derivada parcial em relação a coordenada X pode trazer informações acerca de bordas verticais. A derivada parcial em relação a coordenada Y, por sua vez, trás informações acerca das bordas horizontais (HILDRETH;; BRENNER, 1980).

Para identificar se uma região mapeada através dessas operações como uma borda de fato, é necessário definir um valor limiar para que a mudança de intensidade na imagem seja considerada uma borda. A definição desse valor depende da aplicação e das imagens disponíveis e deve ser determinada de maneira empírica.

Diversos filtros foram propostos na literatura se baseiam no conceito de derivada de primeira ordem, também chamados de operadores de gradiente, entre eles estão os filtros

Roberts (ROBERTS, 1965), Prewitt (PREWITT, 1970) e Sobel (SOBEL; FELDMAN, 1973).

Um exemplo de aplicação do filtro de Prewitt é apresentado na Figura 6. Além da derivada de primeira ordem, a detecção de bordas também pode ser implementada através da derivada de segunda ordem de uma imagem. Tais operações são classificadas na literatura como operadores de Laplace. A partir das definições do cálculo, a derivada de

Figura 6 – Derivadas parciais em relação a X, em relação a Y e o resultado final do Operador de Prewitt

Fonte - elaborado pelo autor

segunda ordem informa sobre pontos de inflexão em uma função. Quando são considerados os operadores de Laplace, procura-se regiões da imagem onde a derivada segunda vale zero, referido na literatura como *"zero-crossings"* (HILDRETH;; BRENNER, 1980).

Toda vez que a derivada parcial de segunda ordem de uma imagem atravessa o zero, indica a presença de uma borda. Canny, um dos detectores de borda mais bem sucedidos em Visão Computacional (CANNY, 1987) combina as abordagens dos operadores de Gradiente e operadores de Laplace.

2.5.5 Detecção de *Blobs*

De maneira similar a detecção de bordas, outra operação importante no contexto de Processamento de Imagens é a detecção de *blobs*. *Blobs* podem ser definidos como regiões de interesse na imagem que podem conter objetos ou texturas (DANKER; ROSENFELD, 1981).

Assim como a detecção de bordas, a detecção de *blobs* se baseia na taxa de mudança de intensidade dos pixels em uma determinada região da imagem. A detecção de *blobs* tem aplicações na detecção e rastreamento de objetos em imagens digitais.

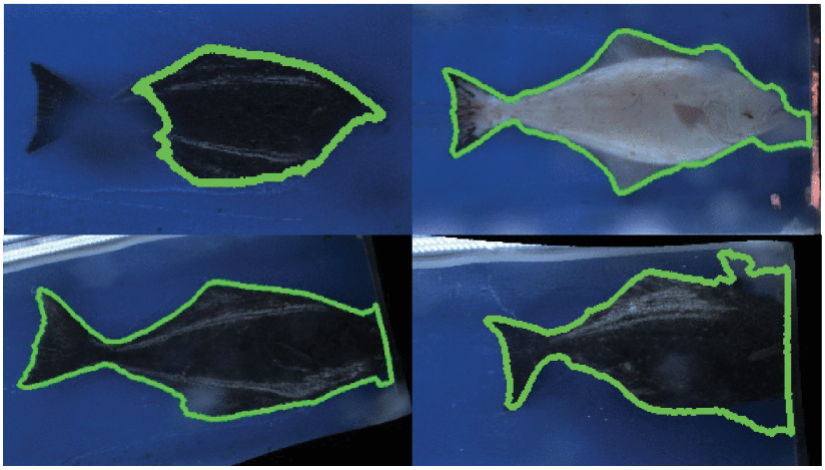
2.5.6 Active Contours

A técnica de *Active Contours* também chamada de *snakes* (KASS; WITKIN; TERZOPOULOS, 1988) é um arcabouço de Processamento de Imagens utilizado para delinear objetos em uma imagem. Essa técnica se baseia em definir uma silhueta inicial para determinado objeto e iterativamente ajustar esse contorno através da minimização de uma função objetivo.

A minimização dessa função busca atrair os contornos iniciais, de forma que, a silhueta inicial se aproxime das bordas de um objeto. Então, torana-se possível segmentar esse objeto em uma imagem. A técnica depende da interação com um usuário, haja visto que, é necessário a definição de um contorno previamente definido a ser ajustado a um objeto.

Então, a partir de uma silhueta pré-definida para um objeto de interesse, é possível aplicar uma técnica iterativa como o método de mínimos quadrados reponderados (WANG et al., 2019a) até se obter o resultado desejado. Um exemplo aplicado a segmentação de peixes é apresentado na Figura 7.

Figura 7 – Silhueta adaptada a um peixe da espécie Halibut



Fonte: (WANG et al., 2019a)

2.6 Aprendizagem de Máquina

A Aprendizagem de Máquina é uma área da Inteligência Artificial e tem por objetivo desenvolver algoritmos capazes de aprender de maneira análoga a mente humana. A aprendizagem de máquina pode ser dividida em três categorias: aprendizagem supervisionada, aprendizagem semi-supervisionada e aprendizagem não supervisionada.

Para o presente estudo é considerado a aprendizagem supervisionada onde procura-se aproximar um padrão de saída através de um estímulo de entrada. Em Aprendizagem

de Máquina é denominada tarefa a formalização de um problema em particular. As tarefas relevantes para o presente estudo são:

- **Classificação:** A tarefa de classificação consiste em identificar a qual categoria ou subpopulação determinado elemento pertence. Dado um conjunto de categorias existentes, busca-se atribuir uma categoria a um elemento a partir de suas características;
- **Regressão:** A tarefa de regressão consiste em aproximar um ou mais valores contínuos a partir de um conjunto de características disponíveis;
- **Detecção de Objetos:** A tarefa de detecção de objetos consiste identificar e localizar em qual região de uma imagem determinado objeto se encontra;
- **Segmentação Semântica:** A tarefa de segmentação semântica consiste em particionar os pixels de uma imagem em segmentos contíguos, de forma que, esses segmentos pertençam a um objeto em particular; e
- **Estimativa de pontos-chave:** A tarefa de estimação de pose consiste em identificar pontos chave em uma imagem, como por exemplo articulações de uma pessoa, e pode ser formulado como um problema de regressão com múltiplas saídas simultâneas;

2.6.1 Aprendizagem Supervisionada

Um problema de aprendizagem supervisionada pode ser formalizado como um conjunto de N exemplos de treinamento na forma $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, x_i sendo um vetor de entrada e y_i a saída desejada para cada vetor de entrada apresentado ao algoritmo.

Uma possível abordagem algorítmica para aproximar uma função arbitrariamente complexa entre esses padrões de entrada e saída, é a aplicação de uma técnica chamada de Aprendizado Profundo, (GOODFELLOW; BENGIO; COURVILLE, 2016a).

Em uma de suas formas mais básicas, um modelo de Aprendizado Profundo consiste em uma composição de funções não-lineares mapeando o padrão de entrada x_i para uma saída y_i.

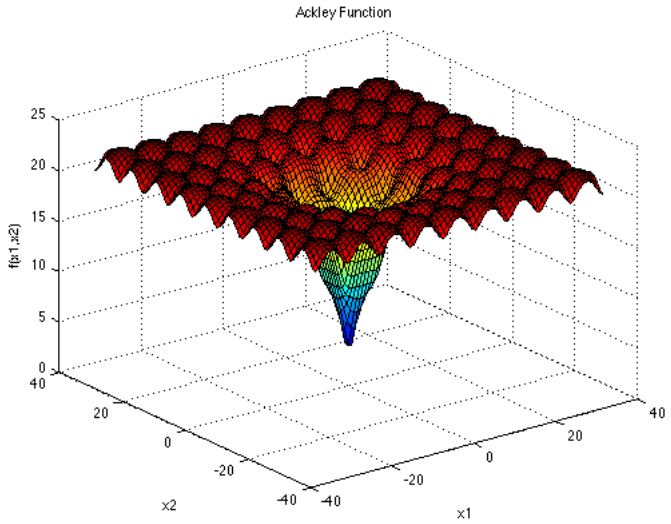
2.6.1.1 Aprendizado Profundo

Em 1989, Cybenko demonstrou que um modelo composto por número arbitrariamente grande de unidades logísticas poderiam compor aproximadores universais de funções (CYBENKO, 1989).

A descoberta motivou a exploração de modelos com um número maior de camadas escondidas e parâmetros. Na maioria dos casos, a função a ser otimizada é uma superfície não-convexa como a função de Ackley, apresentada na Figura 8.

Tais funções podem ser extremamente difíceis de otimizar pela presença de múltiplos mínimos locais, o que dificulta o treinamento de modelos muito profundos pois o algoritmo de otimização pode ficar preso em um desses *plateaus* convergindo para uma solução sub-ótima para o problema.

Figura 8 – Função de Ackley



Fonte - elaborado pelo autor

O processo de treinamento em Redes de Aprendizado Profundo consiste em otimizar uma função f que mapeia um valor de entrada para um valor de saída dado por $f(x) = y$. Com tal composição de funções é possível aproximar soluções para problemas de regressão ou de classificação. Tais funções podem apresentar complexidade arbitrária com diversos mínimos locais.

A função f é formada através de outras funções que representam as camadas da Rede Neural Profunda. No entanto, quanto mais complexa for a composição de funções de mapeamento maior será o número de camadas, consequentemente um número maior de parâmetros será necessário para representá-la.

2.6.1.2 Backpropagation

Devido a essa natureza composicional uma técnica em específico normalmente é empregada para treinar essa classe de algoritmos. Tal técnica é denominada *backpropagation* (WERBOS, 1974; RUMELHART; HINTON; WILLIAMS, 1986) que consiste na propagação dos erros no sentido contrário da composição de funções usando o gradiente descendente como método de otimização.

O trabalho de Rumelhart é considerado seminal sendo um dos mais importantes no reaparecimento do interesse da comunidade científica em modelos de Aprendizagem de Máquina. Para efetuar a propagação dos erros é necessário um esforço computacional alto, devido a necessidade de operações de derivação obedecendo a regra da cadeia aplicada a composição de funções.

O treinamento pode ser efetuado propagando-se os erros obtidos após a comparação do valor de saída aproximado $\hat{y}$ com o valor real y (HAYKIN, 2009). Esse sinal é derivado em relação aos parâmetros de camada que o precede e assim sucessivamente até alcançar a função mais externa da composição (RUMELHART; HINTON; WILLIAMS, 1986).

As derivadas parciais então são multiplicadas e podem ser utilizadas para atualização dos parâmetros de cada camada de funções compostas. O processo deve ser repetido até que um determinado critério de convergência ou critério de parada seja satisfeito.

2.6.2 Métodos de Aprendizagem

Em relação ao modo como os exemplos são apresentados ao algoritmo supervisionado existem dois métodos de aprendizagem: *aprendizagem estocástica* e *batch* (HAYKIN, 2009), (RUMELHART; HINTON; WILLIAMS, 1986).

A aprendizagem estocástica é caracterizada pelo cálculo do gradiente e atualização dos paramêtros do modelo exemplo a exemplo. Neste método o algoritmo esta sujeito ao ruído gerado por variações entre diferentes exemplos do conjunto de treinamento o que pode levar alguns modelos a um tempo maior até atingir a convergência.

No método de *batch*, são apresentados todos os exemplos de treinamento disponíveis antes de calcular o gradiente e executar uma atualização no modelo. Este método pode acelerar o processo de convergência.

No entanto, se o conjunto de treinamento for muito extenso pode ser impraticável apresentar todos os exemplos em uma única iteração por restrições de recursos computacionais disponíveis atualmente.

O método de *batch* possui uma variação: o método de *mini-batch* que consiste na subdivisão do conjunto de treinamento em k subconjuntos disjuntos. As atualizações no modelo serão aplicadas somente após apresentação de cada um dos k subconjuntos ao algoritmo.

Dessa forma, é possível ajustar o tamanho dos *mini-batches* de forma que seja tratável com os recursos computacionais disponíveis ao mesmo tempo em que é aliviada a questão da alta taxa de variação entre os exemplos.

Isso resulta em um tempo de convergência potencialmente menor e um uso mais eficiente dos recursos computacionais disponíveis.

2.6.3 Generalização

No contexto de aprendizagem supervisionada, geralmente o conjunto de exemplos é dividido em ao menos dois conjuntos disjuntos: treinamento e teste. O conjunto de treinamento é utilizado para ajustar os parâmetros do modelo através de uma técnica de otimização como a descrita na Subseção 2.6.1.2.

O segundo conjunto é utilizado para avaliar o desempenho do modelo através de uma determinada métrica, aplicando-o a um conjunto de exemplos ainda não vistos pelo modelo durante a fase de treinamento.

Quando um modelo tem um desempenho adequado na métrica de avaliação no conjunto de testes é possível quantificar a capacidade do modelo de generalizar as informações contidas no conjunto de treinamento para o conjunto de testes.

2.7 Considerações Finais

Neste capítulo foram apresentados conceitos centrais sobre Processamento de Imagens, Visão Computacional e Aprendizagem de Máquina a fim de contextualizar o entendimento das técnicas e conceitos relacionadas a este estudo.

Na Seção 2.5 foram abordados conceitos básicos de Processamento de Imagens como a formalização da estrutura de representação de imagens digitais e espaços de cores, o uso de filtro e sua relação com operadores matemáticos nessas estruturas para extração de características relevantes para as análises pretendidas.

A Seção 2.5.4 revisa o problema de detecção de bordas e as principais abordagens para este problema como o uso de derivadas parciais de primeira e segunda ordem para extrair os limites dos objetos representados em imagens.

No Item 2.5.5, é apresentado o problema de detecção de regiões de interesse através do arcabouço *Active Contours* em uma imagem utilizando métodos iterativos para ajustar uma silhueta inicial aos contornos de um objeto em uma imagem digital.

Na Seção 2.6, são apresentadas as diversas tarefas relacionadas a este estudo na área de Aprendizagem de Máquina. Também foram revisados conceitos centrais da Aprendizagem de Máquina como a formalização do problema de aprendizagem supervisionada.

O Item 2.6.1.1 discute os conceitos básicos sobre Aprendizado Profundo, bem como a dificuldade em otimizar esses algoritmos pela presença de regiões de convergência sub-ótimas.

Por fim, o Item 2.6.1.2 apresenta o *backpropagation* um importante método para treinamento de modelos largamente utilizado em Aprendizado Profundo.

Os conceitos apresentados neste capítulo tem por objetivo facilitar o entendimento

da análise subsequente. Dessa forma, no Capítulo a seguir são definidos os blocos e operações básicos de uma das técnicas mais bem sucedidas na área de Visão Computacional: as Redes Neurais Convolucionais.

3 Redes Neurais Convolucionais

3.1 Introdução

De maneira geral, até o final da década de 90 as abordagens mais comuns a tarefa de classificação de imagens era subdividir o problema em duas etapas (LECUN et al., 1989). A primeira etapa consistia em desenhar extratores de características manualmente através da aplicação sucessiva de filtros como os definidos nas Seções 2.5.4 e 2.5.5.

Nessa fase, as informações mais relevantes são extraídas da imagem e informações menos relevantes para a classificação são filtradas. As informações então são projetadas para um espaço de menor dimensionalidade, efetivamente comprimindo a informação contida na imagem original.

A segunda fase consiste em desenhar um classificador para das informações extraídas na primeira etapa. Isso pode ser feito através de abordagens de reconhecimento de padrões clássico como as redes neurais multicamadas ou máquinas de vetores de suporte (CORTES; VAPNIK, 1995).

Todavia, com essa abordagem em duas fases para cada problema diferente o extrator de características, e.g. a sequência de filtros a serem aplicados, precisa ser redesenhado, o que se configura como um processo bastante tedioso.

Em um trabalho seminal da década de 90 (LECUN et al., 1989), um novo modelo é proposto chamado de Redes Neurais Convolucionais capaz de realizar a tarefa em uma única fase de aprendizagem utilizando backpropagation como método de treinamento.

3.2 Propriedades de Modelos de Classificação de Imagens

Antes de iniciar a discussão sobre redes neurais convolucionais é fundamental revisar propriedades desejáveis de um modelo para processamento de dados imagéticos.

Uma dessas propriedades é a capacidade de classificação invariante a transformações lineares da imagem de entrada. Ou seja, o modelo precisa ser capaz de identificar um elemento em uma imagem mesmo que esse elemento apareça em diferentes posições. O modelo também precisa ser capaz de ser capaz de lidar com imagens onde o objeto de interesse esteja a diferentes distâncias do dispositivo de captura (FUKUSHIMA, 1980).

Um modelo de classificação de cachorros precisa ser capaz de identificar o cachorro da Figura 9, mesmo quando a imagem é capturada de perspectivas diferentes. A aplicação de redes neurais profundas diretamente nos pixels das imagens de entrada resulta em um

Figura 9 – Cachorro visto de diferentes perspectivas

Fonte - Elaborado pelo Autor

alto custo computacional, devido a quantidade de parâmetros e multiplicações e adições necessárias, mesmo com imagens de baixa resolução. Além desse modelo ser bastante ineficiente, há um aumento considerável no número de parâmetros necessários para sua representação.

3.3 Processamento Visual Hierárquico

A partir do trabalho de Hubel e Wiesel (HUBEL; WIESEL, 1962), observando a resposta a determinados estímulos de células neuronais em um gato adulto foi possível entender melhor uma relação de estímulo hierárquica entre células mais simples e camadas mais complexas do córtex visual do animal.

Esse foi um resultado fundamental na literatura pois serviu de inspiração para importantes avanços na área visão computacional e processamento de imagens. A partir desses resultados, modelos baseados nessa estrutura hierárquica de processamento foram descobertos.

Dessa forma, os modelos inspirados nessa estrutura como o de Fukushima (FUKUSHIMA, 1980) passaram a incorporar a estruturação dos algoritmos de forma hierárquica. As células das primeiras camadas da rede neural passariam a se especializar em características mais a baixo nível como bordas e discontinuidades da imagem e as camadas posteriores passaram a se especializar em características de mais alto nível e abstratas como silhuetas de objetos.

3.4 Mapas de Características

Tomando como exemplo os operadores de Prewitt na detecção de bordas, são utilizadas ao menos duas matrizes de parâmetros diferentes: uma para detecção de bordas na horizontal e outra para bordas na vertical, conforme a Figura 10.

Após a aplicação de cada operador da Figura 10 obtém-se as duas imagens à

Figura 10 – Operadores Vertical e Horizontal de Prewitt

Kx =

-1	0	1
-1	0	1
-1	0	1

Ky =

1	1	1
0	0	0
-1	-1	-1

Fonte - (MUSA; WATADA, 2008)

esquerda na Figura 6. Cada uma dessas imagens representa um mapa de características que ao serem combinados resultam na imagem à direita da Figura 6 com as bordas detectadas na horizontal e vertical na imagem.

Todavia, os valores das matrizes da Figura 10 foram cuidadosamente escolhidas com o objetivo de realçar as bordas contidas na imagem. As redes neurais convolucionais incorporam a estrutura hierárquica de processamento aprendendo matrizes de parâmetros ou filtros mais simples nas primeiras camadas da rede neural e matrizes de parâmetros especializadas em detectar características mais complexas da imagem nas camadas posteriores da rede.

3.5 Compartilhamento de Parâmetros e Correlação Cruzada

Para resolver a questão da quantidade de parâmetros do modelo a técnica de compartilhamento de parâmetros foi aplicada em (LECUN et al., 1989), essa técnica foi inicialmente descrita por (RUMELHART; HINTON; WILLIAMS, 1986) como uma forma de reduzir o número de parâmetros de uma rede neural.

Além da redução do número total de parâmetros necessários a representação do modelo. Se uma característica relevante da imagem é detectada em uma determinada região da imagem é possível que seja importante também caso apareça em outra região da imagem, e.g. dois cachorros em uma única imagem. O compartilhamento de parâmetros permite que uma característica seja detectada em diferentes regiões da imagem.

O compartilhamento de parâmetros consiste em deslizar uma matriz pela imagem de maneira similar a filtragem linear descrita no Capítulo 2, no entanto, no caso da filtragem linear os parâmetros da matriz a ser aplicada na imagem são cuidadosamente desenhados para que seja implementável através da operação de convolução.

Apesar da similaridade com a filtragem linear, no caso do compartilhamento de parâmetros, os valores dos parâmetros das matrizes são aprendidos ao longo do processo através do *backpropagation*.

Outra distinção importante entre a filtragem linear e o compartilhamento de parâmetros é que a operação realizada não é uma convolução como descrita na enge-

nharia e matemática pura e sim uma correlação cruzada entre a matriz e a imagem (GOODFELLOW; BENGIO; COURVILLE, 2016b).

Portanto, apesar do nome do método ser descrito como Rede Neural Convolucional a operação realizada nesta arquitetura é a Correlação Cruzada. Além da redução de parâmetros, outro benefício importante é trazido pela técnica: a capacidade de detectar uma mesma característica mesmo que ela apareça em diferentes locais da imagem.

Os valores obtidos a partir dessa operação são denominados na literatura como mapas de características, os *feature maps*. A operação consiste então em encontrar automaticamente quais as matrizes de parâmetros serão capazes de extrair as características mais relevantes da imagem para a tarefa em questão através do *backpropagation*.

3.6 Padding e Stride

Um detalhe importante da implementação da operação de convolução é relativo as dimensões das imagens de entrada e as dimensões da matriz de parâmetros. Por exemplo, ao considerar uma imagem de dimensões $n * n$ pixels e aplicar a correlação cruzada com uma matriz da parâmetros $f * f$ e deslizar a matriz 1 pixel por vez obtém-se uma matriz de $n - f + 1 * n - f + 1$ como saída.

É possível deslizar a matriz de pixels mais de um pixel por vez na imagem. O número de pixels utilizado para deslocar a matriz de parâmetros ao longo da imagem é denominada *stride*. A depender das dimensões da matriz de parâmetros após aplicar sucessivas camadas de convolução a resolução da imagem original pode ser reduzida gradativamente.

Além disso, as informações contidas nas bordas das imagens podem ser perdidas quando consideradas redes neurais mais profundas com centenas de camadas, por exemplo. Quando considera-se um *stride* diferente de 1 pode-se perder ainda mais pixels resultando em *feature maps* substancialmente menores.

Para aliviar essa questão é possível utilizar uma técnica chamada *padding*. A técnica consiste em aumentar a resolução da imagem de entrada adicionando-se pixels as bordas. Geralmente são utilizados pixels com valor 0, de maneira que, a informação perdida após a aplicação da convolução corresponda a área adicional, mantendo assim toda a informação dos pixels localizados as bordas da imagem original projetadas no mapa de características obtido.

Por exemplo, um *padding* de 2 adiciona 2 pixels as bordas da imagem. A fórmula para calcular a dimensão de saída da imagem torna-se $n + 2p - f + 1 * n + 2p - f + 1$, p sendo o valor assumido na operação de *padding*.

3.6.1 Stride

Utilizar um valor de deslocamento (*stride*) maior do que 1 pode ser útil para realizar uma redução de resolução mais acentuada a fim de tornar viável, em termos de custo computacional, a aplicação do método para imagens de resolução mais alta. O custo dessa operação é a perda de detalhes mais finos da imagem, pois, a operação tende a comprimir a resolução original da imagem. Para o calcúlo dos *feature maps* com stride diferentes de 1 aplica-se a seguinte fórmula: $(n + 2p - f)/s + 1 * (n + 2p - f)/s + 1$, sendo f a dimensão da matriz de parâmetros, n a dimensão da imagem de entrada, p o valor do *padding* e s o valor do *stride* utilizado na operação.

3.6.2 Same Convolution e Valid Convolution

Na literatura, são mencionados frequementemente dois tipos de padding em convoluções: *Same* e *Valid*. A Convolução *Same* se refere a manter o mapa de características obtido com a mesma resolução da imagem original. Para isso, deve se considerar o valor do *padding* de acordo com a dimensão da matriz de parâmetros a ser aplicada.

Para calcular o padding necessário a partir da dimensão da matriz de parâmetros, sendo f a dimensão da matriz de parâmetros, n a dimensão da imagem de entrada e p o valor do *padding* aplicado, aplicando-se a seguinte fórmula $n+2p-f+1 = n$, simplificando, $p = (f - 1)/2$.

A Convolução *Valid* se refere a não aplicar nenhum *padding*, ou seja, assumir o valor $p = 0$ e obter um mapa de características de menor dimensão como saída.

3.7 Pooling

Apesar do compartilhamento de parâmetros reduzir o quantidade de parâmetros necessários a representação modelo, quando uma imagem com dimensões maiores é considerada, os mapas de características obtidos ainda terão dimensões similares a imagem original. Então, ao aplicar apenas esta técnica ainda tem se como resultado problemas relacionados ao custo computacional alto, conforme avança-se das camadas mais simples as mais profundas da rede neural.

Para aliviar esta questão e reduzir o número de parâmetros, é necessário aplicar uma espécie de compressão nos mapas de características obtidos através da correlação cruzada. Esta é uma forma de reduzir as dimensões espaciais do mapa de características, retendo apenas as informações mais importantes descartando informações não relevantes.

A operação é similar a operações de redimensionamento de imagens. Na operação de redimensionamento é calculada a média dos valores dos pixels em determinada vizinhança da imagem a fim de se obter uma imagem de menor resolução. O custo dessa operação

de redimensionamento de imagens é a perda de algumas informações de detalhes finos. A partir deste problema então foi proposto a aplicação da técnica de *pooling*.

A técnica de *pooling* ou *sub-sampling* consiste em aplicar uma operação para eliminar características irrelevantes de um sinal (BENGIO; LECUN, 1997). Existem diversas técnicas de *pooling*, sendo as mais populares o *Max-Pooling* e o *Average-Pooling*.

Outras variações de *pooling* foram descritas na literatura ao longo dos anos como *Sum-Pooling Mixed-Pooling*, *Stochastic-Pooling*, *Gaussian-Pooling*, *Median-Pooling*, *Minimum-Pooling* entre outros (PALA et al., 2018). Outro método relevante é o *Region of Interest Pooling (RoI Pooling)* (GIRSHICK, 2015)(LIU et al., 2018) bastante importante nas arquiteturas a serem discutidas no contexto de segmentação de imagens, um dos objetivos deste trabalho.

Para realizar a operação de redução da dimensão espacial de um mapa de características, primeiro é necessário definir o tamanho da matriz a ser aplicada. Ou seja, a área onde é aplicado o processo de pooling. Quanto maior a dimensão dessa matriz maior a capacidade de compressão da técnica.

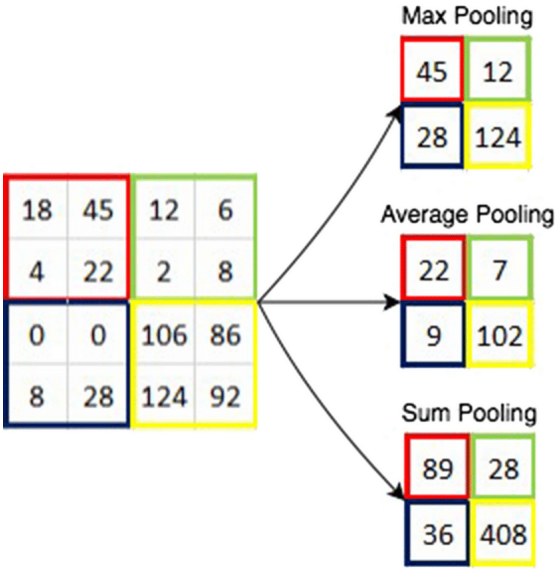
De maneira similar a camada convolucional da rede, a matriz de pooling é deslizada ao longo do mapa de características. Essa camada não possui parâmetros a serem aprendidos ao longo do processo de *backpropagation*. No entanto, é preciso encontrar a dimensão mais adequada para essa matriz, para não eliminar informações importantes do mapa de características. O tamanho dessa matriz é um hiperparâmetro pré-definido na arquiteura, logo, não é um parâmetro obtido durante o processo de aprendizagem.

3.7.1 Max-Pooling e Average-Pooling

As técnicas utilizadas mais comumente na literatura clássica são o *Max-Pooling* e *Average-Pooling*. A técnica de *Max-Pooling* refere-se a aplicar uma matriz quadrada de dimensão k e extrair o maior valor contido na área do mapa de características reduzindo dessa forma uma área de $k * k$ a um único valor.

A técnica de Average-Pooling consiste em aplicar uma matriz quadrada de tamanho k a área correspondente no mapa de características calculando-se o valor médio dos valores da área coberta pela matriz, de maneira similar ao Max-Pooling, reduzindo uma área $k * k$ do mapa de características a um único valor, nesse segundo caso a média dos valores contidos na área da matriz de pooling. Na Figura 11 o processo é diagramado visualmente.

Figura 11 – Operações de *Pooling*



Fonte - (SEVEN et al., 2022)

3.8 Considerações Finais

Neste capítulo foram discutidos os elementos básicos que caracterizam as Redes Neurais Convolucionais e a relação desses métodos com trabalhos anteriores.

Na Seção 3.1 foi revisada a abordagem dominante na literatura até o início da década de 90, que consistia no desenho manual de extratores de características utilizando os conceitos de filtragem revisitados no Capítulo 2.

Na Seção 3.2 foram abordados as propriedades desejáveis a modelos de processamento de imagens. Na sequência, na Seção 3.3 são revisitados trabalhos seminais que serviram de inspiração para a arquitetura das Redes Neurais Convolucionais. Nas seções 3.4, 3.5 e 3.7.1 foram discutidos os elementos básicos que caracterizam a arquitetura das Redes Neurais Convolucionais.

A partir dos conceitos documentados neste Capítulo, avança-se a discussão sobre como as arquiteturas de Redes Neurais Convolucionais evoluíram da década de 90 até o momento atual. No capítulo a seguir, são revisitados os principais trabalhos e quais problemas esses trabalhos buscaram endereçar.

4 Arquiteturas de Redes Neurais Convolucionais

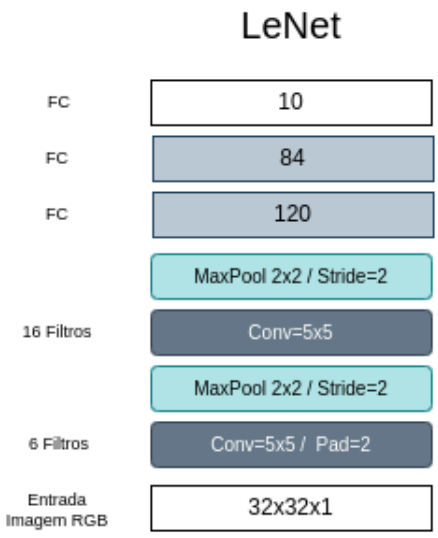
4.1 Introdução

Neste capítulo são revisitados em ordem cronológica a evolução das arquiteturas baseadas em Redes Neurais Convolucionais, as principais contribuições desses trabalhos, bem como, os problemas endereçados ao longo dessa evolução.

4.2 LeNet

A arquitetura LeNet (LECUN et al., 1998) foi um dos primeiros trabalhos baseados em Redes Neurais Convolucionais a obter desempenho similar as Máquinas de Vetores de Suporte (CORTES; VAPNIK, 1995). Essa era a abordagem dominante a época para este tipo de problema, devido a sua capacidade de lidar com problemas com alta dimensionalidade, fenômeno comum no contexto de imagens digitais. A Figura 12 ilustra a arquitetura da LeNet.

Figura 12 – Arquitetura da *LeNet*



Fonte - elaborado pelo autor

A LeNet foi uma evolução da abordagem desenvolvida em (LECUN et al., 1989). A qual, de maneira similar, realizava a identificação de códigos postais escritos a mão

da agência postal de Buffalo em Nova Iorque. O problema a ser resolvido neste segundo trabalho era o reconhecimento de dígitos escritos a mão em cheques bancários.

A arquitetura foi implementada comercialmente e operou durante bastante tempo, realizando a digitalização de informações. Sua estrutura foi construída com 2 camadas convolucionais seguidas por operações de pooling. Os mapas de características extraídos a partir desta operação eram alimentados a camadas densamente conectadas também denominada *Multi-Layer Perceptron (MLP).*

As imagens de entrada da arquitetura tinham dimensões de 32x32 pixels, como o maior dígito presente no conjunto de treinamento disponivel ocupava no máximo uma área de 20x20 pixels localizado aproximadamente ao centro da imagem, as imagens de 32x32 pixels foram reduzidas em 4 pixels.

Essa redução na imagem de entrada representava um ganho significativo de desempenho pois o hardware disponível na década de 90 era bastante restrito. Após a aplicação da primeira camada convolucional *same* com 6 filtros de 5x5 e *padding* = 2, 6 mapas de características de 28x28 pixels são obtidos, conforme a Seção 3.6.2 as dimensões obtidas no mapa de são as mesmas do padrão de entrada.

Após a camada de convolução, uma operação de *average-pooling* com uma matriz de dimensão 2x2 e *stride* = 2 reduz a resolução dos 6 mapas de características de 28x28 para 14x14 pixels. Seguindo a ideia de processamento hierárquico documentada na Seção 3.3, esses mapas de características são alimentados a uma segunda camada de convolução. Essa camada seria responsável por detectar características mais complexas da imagem de entrada.

A segunda camada era composta por convolução *valid* com 16 filtros 5x5 resultando em 16 mapas de características de 10x10 pixels. Após a segunda camada de convolução, uma operação de pooling 2x2 com *stride* = 2 reduz os 16 mapas de características para 5x5 pixels. Após essa camada, os mapas de características são alimentados a uma camada totalmente conectada *Fully Connected* (FC). No entanto, as camadas totalmente conectadas não podem ser aplicadas diretamente nos mapas de características, haja visto que, são valores tridimensionais: 16 matrizes de 5x5 pixels.

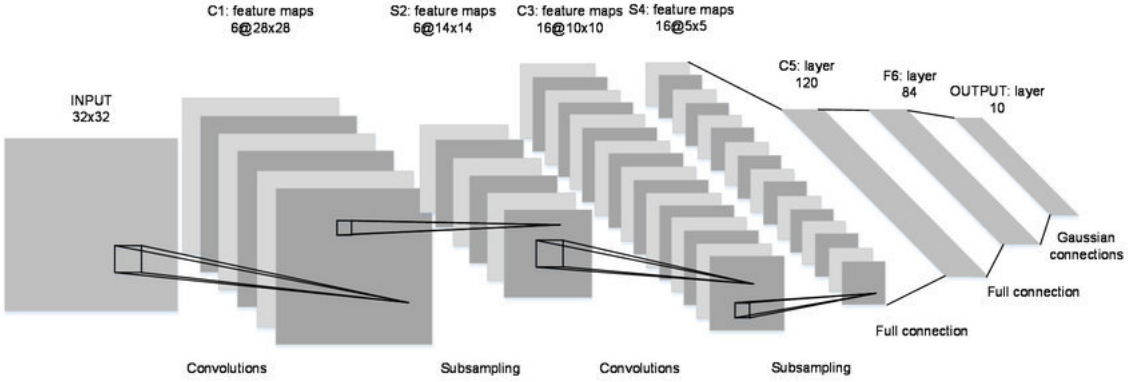
Para que esses valores possam ser alimentados a próxima camada, é necessário achatar essas matrizes de maneira que tenham apenas uma dimensão. Dessa forma, o achatamento das matrizes é realizado, obtendo-se assim um vetor de 16*5*5 = 400 posições.

Esse vetor é alimentado a uma camada totalmente conectada com 120 neurônios, seguida por uma segunda camada totalmente conectada com 84 neurônios e uma camada de saída de 10 neurônios. Um neurônio para cada uma das classes (dígitos de 1 a 10) modeladas no problema original.

A Figura 13 ilustra todas as camadas da rede com as dimensões dos mapas de

características obtidos em cada camada da arquitetura.

Figura 13 – Arquitetura da *LeNet*



Fonte - (LECUN et al., 1998)

4.3 AlexNet

Apesar do sucesso da LeNet no problema de classificação de dígitos escritos a mão, a viabilidade do uso das Redes Convolucionais ainda enfrentava um obstáculo quando são consideradas imagens com resoluções maiores, em parte a questão se dava em relação aos dispositivos de captura disponíveis a época.

Em 1998 dispositivos de captura de imagens com alta resolução ainda não estavam disponíveis. As imagens de entrada alimentadas a LeNet tinham apenas 28x28 pixels. Além disso, o poder computacional do hardware e capacidade de armazenamento das memórias primárias era muito limitado.

Nesse período, as abordagens dominantes para processamento de imagens continuou sendo a engenharia de extratores de características. Sendo manualmente alimentados a modelos clássicos como os métodos baseados em *kernel* ou modelos lineares (LOWE, 2004), (BAY; TUYTELAARS; GOOL, 2006).

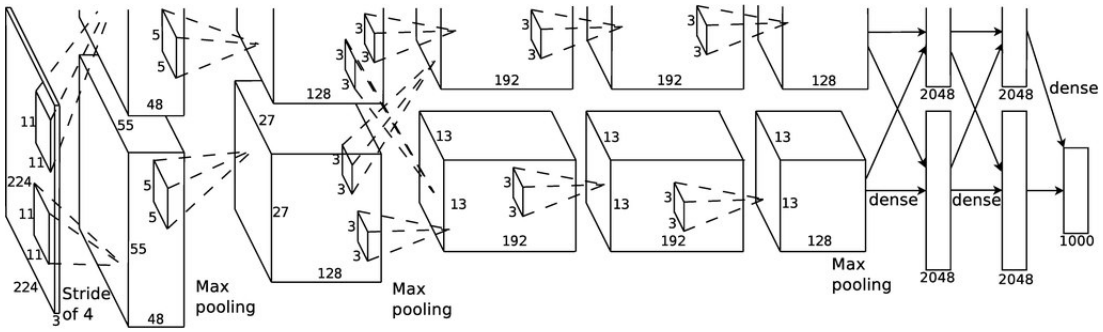
Apesar do custo computacional alto, as operações realizadas nas Redes Convolucionais são paralelizáveis. Todavia, as CPUs presentes na maioria dos computadores pessoais são otimizadas para operações sequenciais. Portanto, as CNNs não são implementáveis de maneira eficiente em CPUs devido ao elevado número de operações aritméticas necessárias. Quando executadas em CPUs de maneira sequencial podem tornar o tempo de processamento inviável.

No contexto de jogos digitais, unidades de processamento gráficos, as GPUs, se tornaram populares. Esses dispositivos eram especializados em operações aritméticas mais simples e altamente paralelas como a renderização de frames sequênciais em um jogo digital. No entanto, esses dispositivos eram otimizados para o processamento gráfico com linguagens específicas para este fim como o OpenGL.

Em 2007, a NVIDIA passa a disponibilizar o framework de computação *CUDA - Compute Unified Device Architecture* para realização de cálculos de propósito geral usando as unidades computação paralela, as GPGPUs - *General Pourpose Graphical Processing Units* de maneira que pudessem ser utilizadas para cálculos aritméticos em geral. Anteriormente as GPUs serviam o propósito de renderização de texturas em jogos primariamente.

Foi somente em 2012, que *Alex Krizhevsky* (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) em conjunto com outros pesquisadores desenvolveram uma versão paralela da operação de convolução para ser executada em GPUs comerciais. O hardware utilizado para treinar o modelo foi um par de placas NVIDIA GTX 580 com 3GB de RAM. A AlexNet precisou ser dividida em duas GPUs pois o número de parâmetros da arquitetura excedia a capacidade de computação e armazenamento primário de uma única unidade da época, conforme a Figura 14.

Figura 14 – Arquitetura da *AlexNet*



Fonte - (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)

A arquitetura da AlexNet é bastante similar a implementada por Yan LeCun em 1998 conforme a Figura 15, a diferença entre as duas estava na escala dos problemas que cada uma era capaz de resolver.

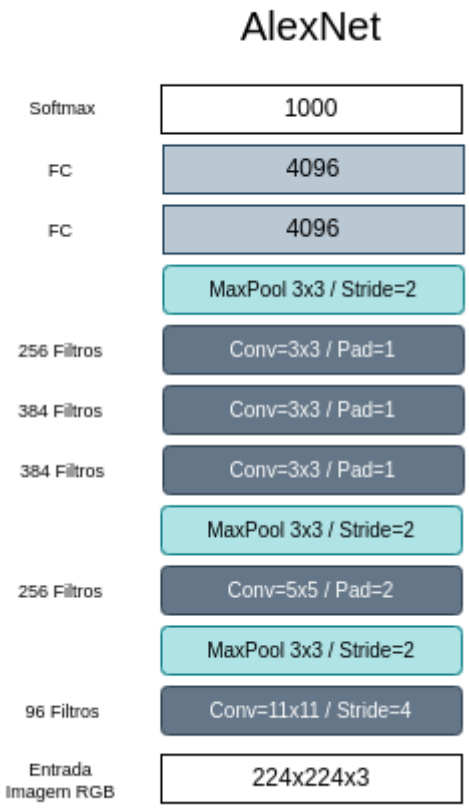
Além da escala dos problemas, outro detalhe importante foi a divisão do modelo em 2 partes, onde cada metade do modelo era computada em um dispositivo diferente devido a limitação de memória disponível aos hardware da época.

A AlexNet era composta por 10 camadas, 5 camadas convolucionais, 3 camadas de Max-Pooling e 2 camadas totalmente conectadas ao final. Além dessas camadas uma camada intermediária de normalização chamada de *Local Normalization Layer*.

Essa camada se fazia necessária devido ao uso da função de ativação ReLU, que diferentemente da Sigmóide empregada na LeNet não tem limite superior quando o valor de entrada é positivo.

Esse tipo de camada caiu em desuso ao longo do tempo em favor de outras abordagens como a inicialização dos parâmetros dos filtros usando heurísticas (HE et al.,

Figura 15 – Arquitetura da *AlexNet*



Fonte - elaborado pelo autor

2015b) de forma a resolver o problema de quebra de simetria ao mesmo tempo que acelera a convergência do modelo.

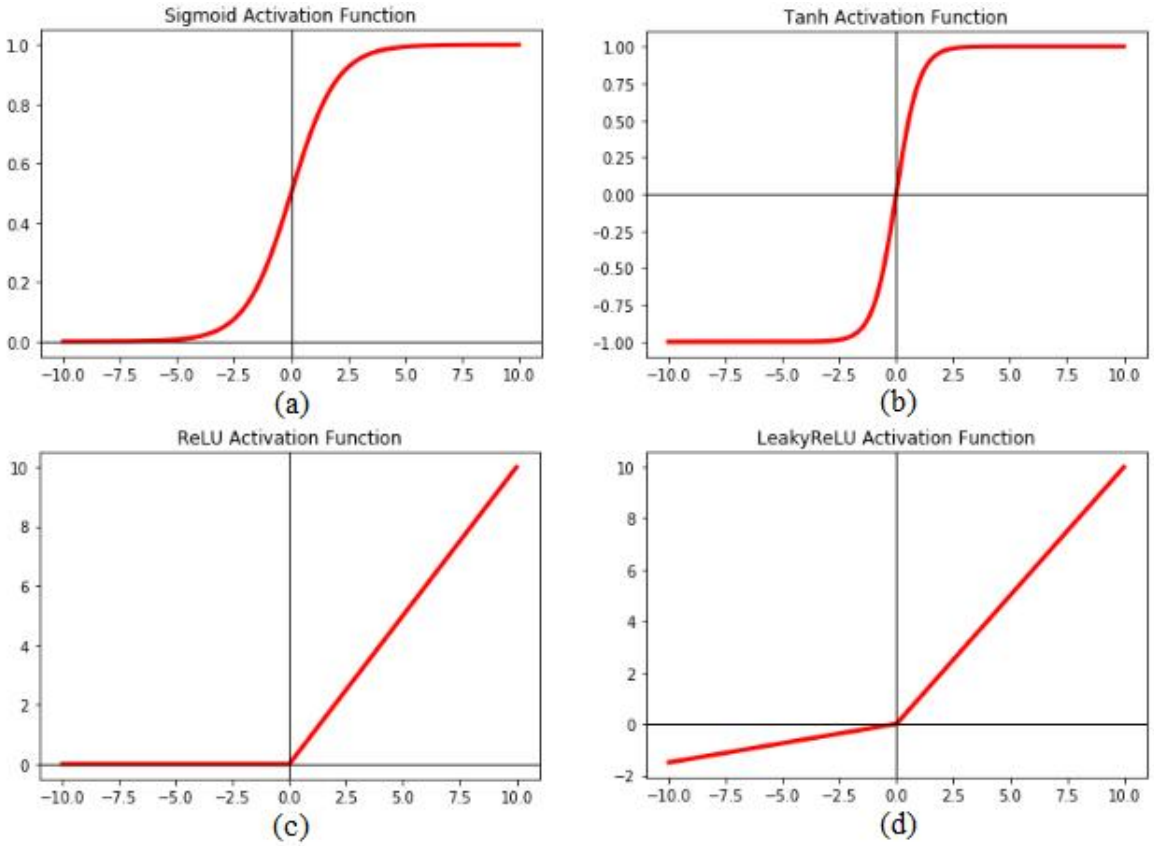
Outra diferença relevante em relação a LeNet é a utilização de um número alto de filtros bem como convoluções com dimensões maiores. A primeira camada de convolução da AlexNet aplicava um filtro de 11x11 para lidar com imagens de entrada com resolução mais alta.

Em relação a função da ativação utilizada, a AlexNet empregava a ReLU enquanto a LeNet utilizava a Sigmóide como função de ativação. A Figura 16 ilustra os intervalos onde as funções de ativação estão definidas.

A sigmóide, devido a operação de exponenciação utilizada, satura o facilmente quando o gradiente se aproxima de 0 ou de 1, já a ReLU, além de ser mais simples de ser calculada não sofre desse problema aliviando a necessidade de se normalizar a entrada da função para evitar a saturação.

Enquanto a LeNet tinha como entrada imagens de 28x28 pixels e 10 classes de saída, a AlexNet foi treinada para resolver um problema com imagens quase 10 vezes maiores de 224x224 pixels e 1000 classes de saída,o Imagenet (DENG et al., 2009).

Figura 16 – Funções de Ativação



Fonte - (KANDEL; CASTELLI, 2020)

As imagens do conjunto Imagenet são divididas em 1000 classes diferentes e serviram para avaliação e comparação de abordagens de Visão Computacional em uma competição que aconteceu anualmente de 2010 até 2017 o *ILSVRC - Imagenet Large Scale Visual Recognition Challenge*.

A AlexNet venceu a edição de 2012 da competição (RUSSAKOVSKY et al., 2013), o segundo colocado foi uma outra arquitetura clássica de Visão Computacional de um grupo de pesquisa de Oxford *Visual Geometry Group - VGG* que ainda empregava a abordagem clássica de desenhar extratores características manualmente e aplicar modelos de aprendizagem de máquina em uma segunda etapa.

Dois anos mais tarde, esse grupo de pesquisa venceria a competição com uma arquitetura baseada em Redes Neurais Convolucionais Profundas com duas arquiteuras proeminentes na literatura: a VGG16 e a VGG19 discutidas na Seção a seguir.

4.4 VGG

Após o sucesso da AlexNet, vários trabalhos subsequentes propuseram melhorias a arquitetura original da AlexNet como (ZEILER; FERGUS, 2013) que avalivou o efeito

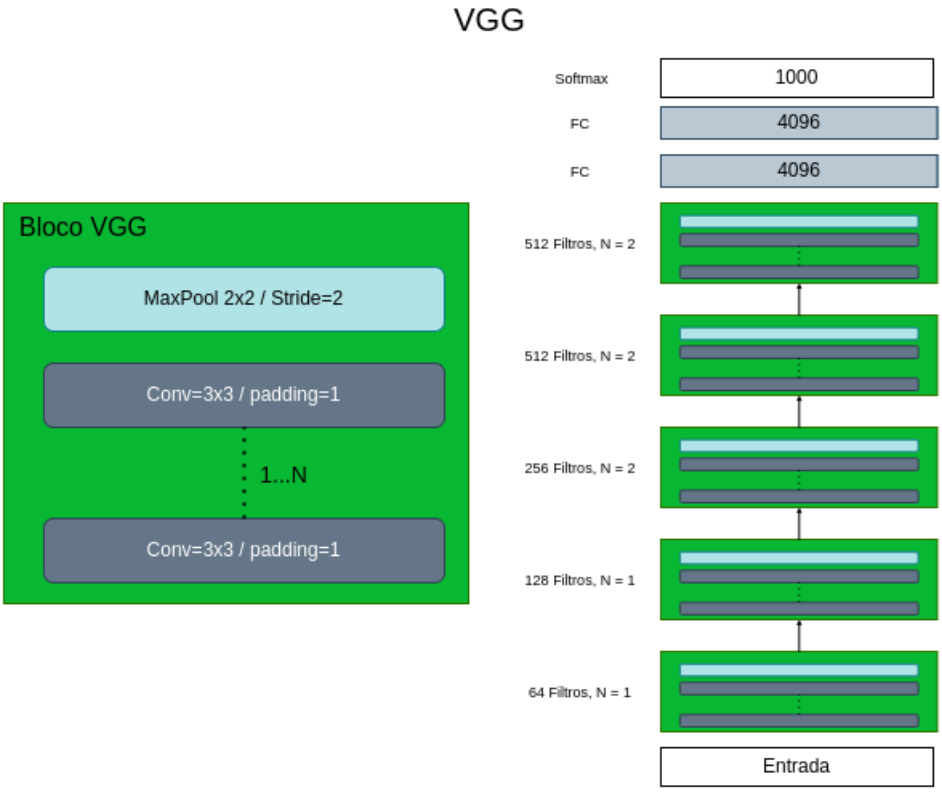
de alterações na estrutura de convolução utilizada na arquitetura da AlexNet empregando um filtro menor que a convolução de 11x11 na primeira camada da rede.

Essa abordagem venceu a edição de 2013 da competição *ILSVRC*. No entanto, foi somente na edição de 2014 que uma mudança mais relevante na estrutura das Redes Neurais Convolucionais ganharia destaque na literatura ao obter uma margem significativa na métrica de desempenho avaliada em algumas das categorias da competição.

A arquitetura proposta pelo grupo *VGG* de Oxford (SIMONYAN; ZISSERMAN, 2014) teria ao menos o dobro de camadas Convolucionais quando comparada a AlexNet. Além disso o grupo explorou um número maior de filtros com menor dimensão empregando primariamente filtros de 3x3 pixels.

Esses filtros seriam agrupados em blocos de sucessivas convoluções seguidas por camadas de ativação e operações de *pooling*. A Figura 17 ilustra a variação com 16 camadas da arquitetura VGG.

Figura 17 – VGG 16



Fonte - elaborado pelo autor

Apesar do sucesso da VGG, dois problemas relevantes no contexto de aprendizagem de máquina passam a se tornar mais evidentes com a arquitetura. Esses dois problemas guiaram o desenvolvimento das arquiteturas subsequentes baseadas em Redes Neurais Convolucionais conforme será abordado nas seções seguintes.

O primeiro problema se refere a presença de camadas totalmente conectadas empregadas após as convoluções. Se na AlexNet duas camadas totalmente conectadas de 120 e 84 neurônios recebendo um vetor de 400 posições como entrada representavam um alto custo computacional e de armazenamento, no caso da VGG o problema foi acentuado ainda mais pois possuía duas camadas totalmente conectadas de 4096 neurônios, recebendo um vetor de características de 7x7x512 = 25088 posições. Ou seja, só a primeira camada totalmente conectada possuía 25088 x 4096 = 102.760.448 paramêtros a serem aprendidos durante o processo de aprendizagem.

O segundo problema se refere a um fenômeno descrito na década de 90 (BENGIO; SIMARD; FRASCONI, 1994), (HOCHREITER, 1991). Esse fenômeno foi estudado no contexto de Redes Neurais Recorrentes que ao ultrapassarem um certo número de camadas na Rede Neural Recorrente o cálculo dos gradientes se aproximava de valores de saturação que aumentavam proibitivamente o tempo de treinamento do modelo.

Devido a natureza não linear do modelo, os gradientes podem facilmente explodir (extrapolar), sendo o oposto também possível caso um dos gradientes em um instante de tempo arbitrário t se aproximar muito de zero, causando um efeito em cascata anulando os gradientes de instantes de tempo ou camadas anteriores.

Esses dois fenômenos são descritos na literatura como a explosão do gradiente e desaparecimento do gradiente respectivamente. O problema é conhecido como problema fundamental do aprendizado profundo relatado inicialmente em (HOCHREITER, 1991). (BENGIO; SIMARD; FRASCONI, 1994) aponta que quando se utiliza a técnica de backpropagation, quanto maior for a quantidade de instantes de tempo no caso de Redes Neurais Recorrentes ou camadas no caso das Redes Neurais profundas, maior será a dificuldade de convergência do modelo.

É importante lembrar que estas condições não afetam somente as Redes Neurais recorrentes, afetam redes neurais de forma geral quando são consideradas arquiteturas com diversas camadas. Essas condições afetam as duas arquiteturas de maneiras similares pois ao representar cada instante de tempo t em uma Rede Neural Recorrente como uma camada escondida de uma Rede Neural Profunda as arquiteturas tornam-se muito similares. (BENGIO; SIMARD; FRASCONI, 1994) evidencia a dificuldade de se utilizar aprendizagem baseada em gradiente quando a duração das dependências nas informações se extendem por longos períodos de tempo.

Para ilustrar o problema basta tomarmos como base as operações realizadas durante o *backpropagation*. Se entendermos a Rede Neural como uma composição de funções $f(g(x))$ onde x é o padrão de entrada da Rede Neural, g é a função que representa a primeira camada e f é a segunda camada, o *backpropagation* é apenas a regra da cadeia aplicada a essa composição de funções.

Logo, segundo a regra da cadeia, a derivada da composição das funções será dada pela multiplicação das derivadas das funções f e g. Portanto, se a derivada da função g em relação a determinado exemplo é um número menor que 1, sucessivas multiplicações de números menores do que 1 vão tender a zero rapidamente, causando o desparecimento do gradiente caso a rede possua um número elevado de camadas. O inverso segue de maneira análoga, ao multiplicarmos sucessivamente números maiores do que 1 o gradiente rapidamente alcança valores muito maiores do que 1 causando assim a explosão ou extrapolação do gradiente.

Apesar dos problemas acentuados pela arquitetura, a VGG é um importante marco na literatura por explorar Arquiteturas Convolucionais mais profundas, obtendo resultados excelentes na competição além de servir como base para avanços na área de Visão Computacional e o sucesso das Redes Neurais Convolucionais.

4.5 Network in Network (NiN)

As arquiteturas discutidas até este ponto compartilhavam estruturas em comum: uma sucessão de operadores convolução seguidos de operações de pooling e por fim camadas totalmente conectadas ao final. As diferenças entre a LeNet, a AlexNet e a VGG consistiam basicamente nas dimensões e quantidades dos filtros das camadas de convolução e as diferentes operações de pooling aplicadas até então. Todavia, apesar da AlexNet ter introduzido o cálculo da operação de convolução de maneira paralela, usando hardware especializado, ainda assim as camadas totalmente conectadas ao final apresentavam um custo computacional e de armazenamento bastante elevado.

Para tentar resolver a questão, (LIN; CHEN; YAN, 2013) introduziu o *Network in Network (NiN)*. A ideia era substituir as estruturas totalmente conectadas após as camadas convolucionais por uma estrutura de pooling global. A estrutura das camadas convolucionais tradicionais era bastante similar a AlexNet, a diferença consistia na adição de duas convoluções de dimensão 1x1 com um número filtros igual ao número de classes do problema, após cada camada de convolução tradicional e antes da operação de pooling correspondente.

A princípio, a adição de uma operação de convolução com um filtro de dimensões 1x1 pode não parecer muito vantajoso, haja visto que as convoluções foram introduzidas para detectar características ao longo das dimensões espaciais da imagem (altura e largura da imagem). Portanto, uma convolução de 1x1 será aplicada em cada pixel da imagem de entrada. Sob essa ótica essa operação age como uma espécie de camada totalmente conectada com a vantagem da redução do número de parâmetros oferecida pelas operações de convolução.

Essa estrutura intermediária organizada em blocos se assemelha a uma pequena

rede neural. O bloco consiste em uma operação de convolução clássica seguida de duas operações de convolução 1x1, similar a uma pequena Rede Neural, seguida da operação de pooling, dai deriva-se o nome Network in Network. Ao final dessas operações, obtém-se uma quantidade de mapas de características equivalente ao número de classes do problema. A estes mapas de características aplica-se a operação de Global Average Pooling. A saída dessa camada é alimentada diretamente a camada de saída para obter a predição.

Apesar do sucesso obtido pela técnica em eliminar a camada totalmente conectada do modelo, a arquitetura não ganhou tanta relevância na época. Todavia, arquiteturas subsequentes inspiradas na estrutura da Network in Network e obtiveram sucesso e relevância na literatura, como é o caso da arquitetura *GoogLeNet - Inception* discutida da Seção a seguir.

4.6 GoogLeNet

Outra arquitetura que ganhou destaque na literatura foi a GoogLeNet (SZEGEDY et al., 2014). O modelo venceu em várias categorias a edição de 2014 do *ILSVRC* e incorporou estruturas descritas previamente na VGG (SIMONYAN; ZISSERMAN, 2014) como a estruturação em blocos repetidos e a aplicação de convoluções de 1x1 e eliminação das camadas totalmente conectadas ao final da arquitetura como descritas em (LIN; CHEN; YAN, 2013).

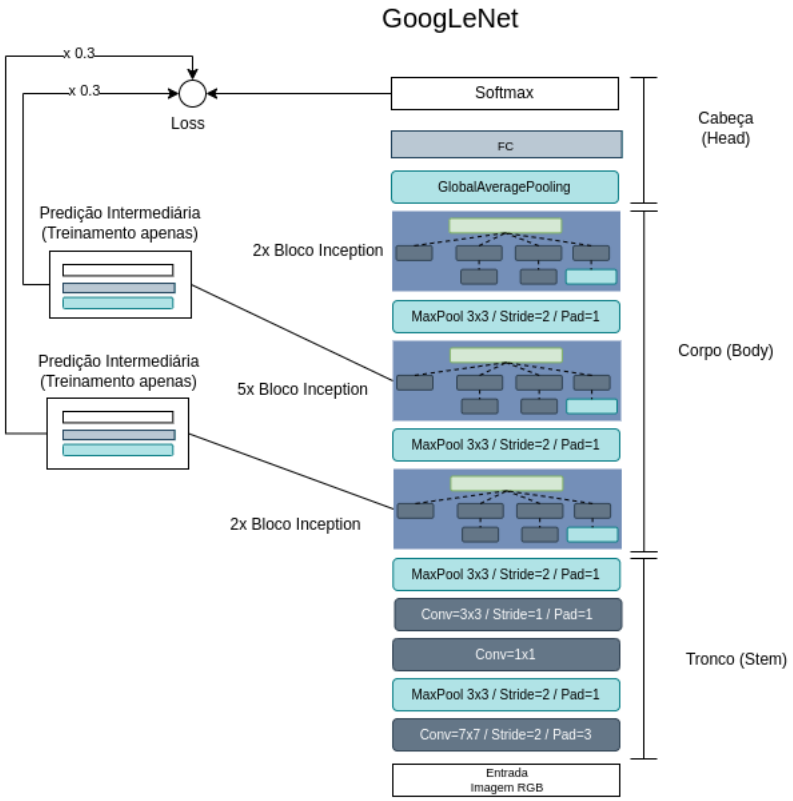
Trazendo uma estrutura ainda mais profunda que suas antecessoras. A GoogLeNet contava com 22 camadas e foi uma das primeiras a subdividir as camadas em 3 partes: tronco (stem), corpo (body) e cabeça (head), conforme a Figura 18. Essa taxonomia se perpetrou na literatura e aparece até nos modelos mais recentes de redes neurais. O tronco corresponde as convoluções iniciais que tinham por objetivo extrair características mais a baixo nível da imagem de entrada.

De maneira geral, convoluções com dimensões maiores reduzem drasticamente as dimensões espaciais da imagem. O corpo corresponde a convoluções intermediárias responsáveis por detectar características mais abstratas. E por fim, a cabeça responsável pela predição a partir das características extraídas em camadas anteriores.

Essa taxonomia é importante pois é possível utilizar o tronco e o corpo das arquiteturas descritas até aqui como extratores de características de propósito geral para tarefas de visão computacional além de classificação como detecção de objetos, segmentação de imagens e rastreio de objetos.

Além da taxonomia, uma característica importante da GoogLeNet foi o uso de múltiplas convoluções em paralelo aplicadas a mesma entrada e a concatenação dos filtros ao final de cada bloco. Essas convoluções são organizadas em blocos. Os blocos da GoogLeNet

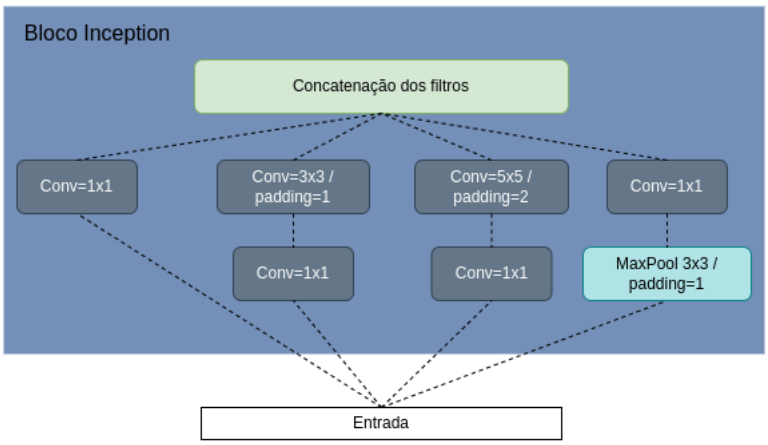
Figura 18 – Arquitetura GoogLeNet



Fonte - elaborado pelo autor

foram batizados de *Inception* e são organizados conforme a Figura 19, em referência ao filme A Origem (*Inception*) de 2010. O bloco aplica convoluções de diferentes dimensões

Figura 19 – Bloco Inception



Fonte - elaborado pelo autor

(1x1, 3x3, 5x5, 1x1). A ideia central das múltiplas convoluções sobre a mesma entrada é a detecção de múltiplas características em diferentes níveis de abstração na mesma camada. Além das convoluções com diferentes tamanhos de filtro, os blocos Inception introduzem

convoluções de 1x1 antes das convoluções de 3x3 e 5x5 e após a operação da Max-Pooling diretamente na entrada do bloco. As convoluções de 1x1 tem por objetivo reduzir o número de filtros adicionando não linearidades entre as camadas e, por fim, diminuindo o número total de operações necessárias ao modelo.

Por exemplo, ao alimentarmos um mapa de características de 28x28 com 256 filtros ao aplicarmos uma convolução de 1x1 com 64 filtros o resultado é um mapa de características de 28x28 e 64 filtros. Esses blocos também são chamados de gargalos de informação e tem por objetivo realizar uma espécie de compressão na informação de entrada projetando a para um espaço de menor dimensionalidade.

Assim como a arquitetura Network in Network descrita na Seção 4.5, a GoogleNet substituiu as camadas totalmente conectadas ao final da arquitetura pelo Global Average Pooling. Além do bloco Inception, a arquitetura introduz camadas auxiliares de predição ao longo da rede durante a fase de treinamento, o erro resultante dessas camadas auxiliares é ponderado por 0,3 e somado ao erro produzido na camada final.

A ideia dessas camadas preditivas intermediárias é forçar o gradiente a fluir das primeiras camadas de maneira que não desapareça ao longo da profundidade da rede. Esse problema é enfrentado em Redes Neurais Profundas em geral conforme descrito na Seção 4.4, dedicada a VGG. A ideia trouxe um conceito interessante para melhorar o fluxo do gradiente em Redes Neurais Profundas e foi explorado com sucesso na arquitetura discutida nas Seção a seguir.

4.7 Residual Networks

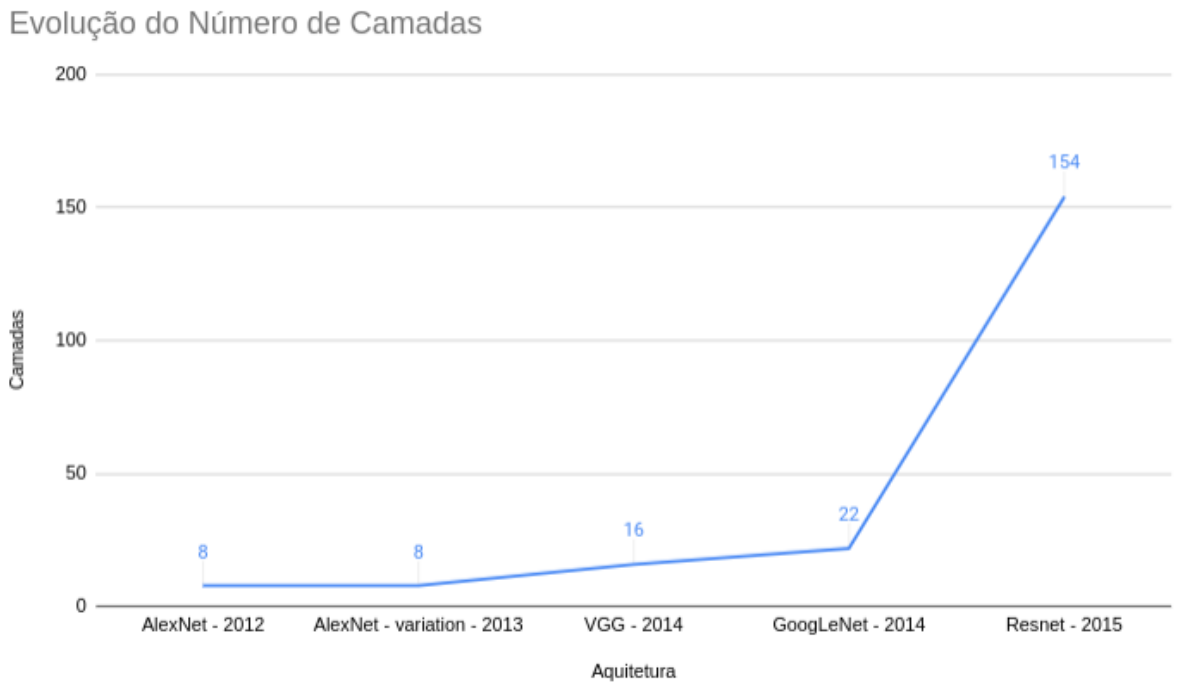
Até este ponto, a tendência geral nas arquiteturas de Redes Convolucionais que obtiveram sucesso nas edições da competição *ILSVRC* era de aumentar o número total de camadas convolucionais.

Todavia, aumentar a profundidade das arquiteturas as tornavam cada vez mais suscetíveis ao problema do desparecimento ou explosão do gradiente (BENGIO; SIMARD; FRASCONI, 1994).

Maneiras de melhorar o fluxo dos gradientes ao longo das camadas da Rede Neural foram exploradas na GoogLeNet (SZEGEDY et al., 2014) com as camadas de predição intermediária e também em outros trabalhos que não ganharam tanto destaque na literatura como as Highway Networks (SRIVASTAVA; GREFF; SCHMIDHUBER, 2015).

A arquitetura das Residual Networks (HE et al., 2015a), (HE et al., 2016) teve um profundo impacto na maneira como Redes Neurais profundas são construídas, ao possibilitar o treinamento de redes bastante profundas em relação a suas antecessoras conforme a Figura 20.

Figura 20 – Evolução da Profundidade das CNNs



Fonte - elaborado pelo autor

As Residual Networks venceram em várias categorias da edição de 2015 da *ILSVRC* com uma margem considerável em relação as suas antecessoras. Partindo da premissa que ao tornar uma rede mais profunda a capacidade de generalização de uma arquitetura de Rede Neural profunda aumenta, pois é possível representar mais informação através de um número maior de parâmetros.

Todavia, durante os experimentos os criadores da arquitetura perceberam que isso não é necessariamente verdade. Haja visto que ao comparar o desempenho de arquiteturas mais profundas com arquiteturas mais simples, as arquiteturas com menos camadas se saíam melhor.

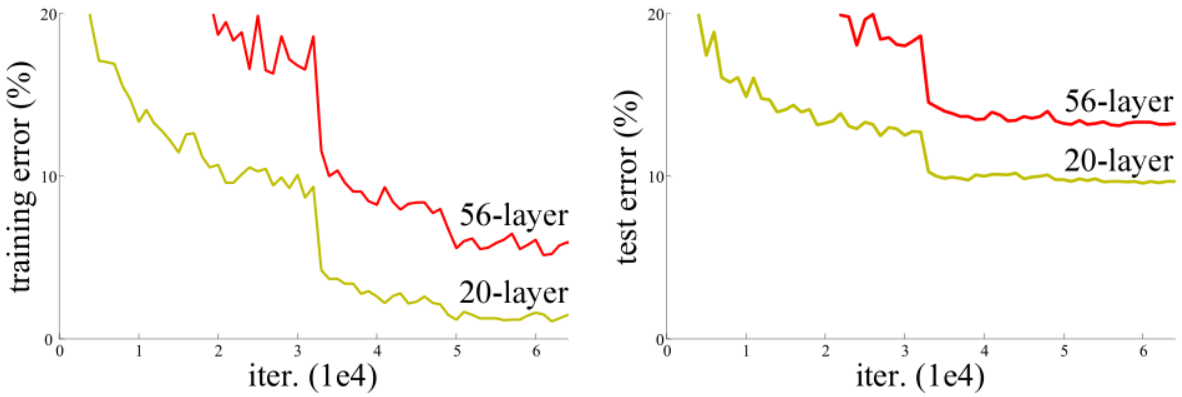
Ao aumentar o número de camadas da rede, geralmente se esta sujeito a um fenômeno chamado sobreajuste, onde, devido ao grande número de parâmetros a arquitetura simplesmente memoriza o conjunto de treinamento. Isso pode ser verificado através das curvas de desempenho dos algoritmos no conjunto de treinamento e no conjunto de testes.

O que os criadores da Resnet perceberam foi que arquiteturas mais profundas com mais camadas convolucionais tinham um desempenho pior tanto no conjunto de treinamento quanto no conjunto de testes, conforme ilustra a Figura 21.

Isso os levou a hípotese de que o problema não se tratava do fenômeno de sobreajuste e mas de um problema relacionado a otimização do modelo durante o treinamento.

No trabalho os autores da arquitetura avaliaram que possivelmente o problema não

Figura 21 – Erros no conjunto de treinamento e de teste



Fonte - (HE et al., 2015a)

estava relacionado ao desaparecimento do gradiente ao longo das camadas. Esse fenômeno que faz com que as camadas mais profundas da rede não sejam treinadas apropriadamente, causando o desempenho da arquitetura ser pior em ambos os conjuntos treinamento e de teste.

Eles então propuseram um mecanismo chamado *shortcut connections*. As *shortcut connections* consistem em pular camadas da rede e alimentar em uma operação aditiva diretamente a entrada de camadas posteriores. A Figura 22 ilustra uma *shortcut connection*. Dessa forma, as camadas que se encontram entre essas *shortcut connections* são forçadas a aprender somente o valor resídual entre uma camada e outra. Ou seja, a diferença entre

Figura 22 – Bloco Residual

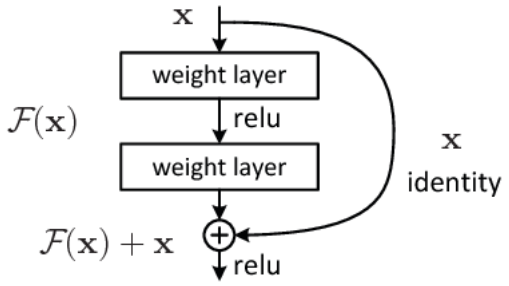


Figure 2. Residual learning: a building bloc

Fonte - (HE et al., 2015a)

a entrada da camada e o valor de saída. Os autores levantam como hípotese que é mais simples aprender o valor residual do que uma função de transformação original entre as camadas.

A fundamentação para o sucesso das Residual Networks consiste em tornar a função a ser aprendida em cada camada mais simples para os algoritmos de otimização. Se conjecturarmos que algumas das camadas de extração de características intermediárias podem ser irrelevantes, a aplicação de não linearidades entre as camadas pode tornar

complicada a otimização pois o algoritmo de otimização precisaria forçar essas camadas intermediárias a não aplicar nenhuma transformação na entrada.

Ao reformularmos as camadas com as *shortcut connections* a função modelada corresponde o valor residual entre a entrada e a saída da camada. Caso a diferença entre a entrada e a saída seja zero, efetivamente a função identidade da entrada ou uma aproximação dela é aprendida. Os autores argumentam que esse pode ser a principal razão para o sucesso que as Residual Networks obtiveram quando se trata do treinamento de arquiteturas mais profundas.

5 Segmentação de Imagens e Mask R-CNN

5.1 Introdução

Até este ponto, a discussão das Redes Neurais Convolucionais se concentrou nas tarefas de classificação de imagens, ou seja, dada uma imagem de entrada procura-se categorizar em uma ou mais classes a imagem de entrada. Entretanto, existem outras tarefas de visão computacional que usam como base as arquiteturas discutidos até aqui.

A partir da edição de 2011 do *ILSVRC*, a tarefa de localização foi incorporada. A tarefa de localização consiste em predizer uma caixa delimitadora (do inglês, *bounding box*) ao redor dos objetos de interesse na imagem. A predição da caixa delimitadora é dada por 4 coordenadas que formam um quadrilátero ao redor do objeto em questão.

Na edição de 2013 outra tarefa foi adicionada: a detecção de objetos. A detecção de objetos é uma tarefa similar a localização. Sua principal diferença é que o algoritmo em questão deve estimar as coordenadas da caixa delimitadora, a classe do objeto contido pela caixa delimitadora e um valor real representado uma pontuação de confiança em relação a predição da classe do objeto contido na caixa delimitadora.

Na edição de 2015 duas tarefas foram introduzidas: detecção de objetos a partir de vídeos e classificação de cenários. A tarefa de detecção de objetos a partir vídeos alterava apenas o formato de entrada para o algoritmo, ao passo que a classificação de cenários consistia em classificar diferentes cenários como montanhas, vales, florestas entre outros.

A partir da edição de 2016 uma nova tarefa passa a aparecer: a análise de cenários. A análise de cenários consiste em segmentar semânticamente todos os pixels de uma imagem de um cenário em categorias como céu, estrada, pessoa, árvore entre outros. A edição de 2017 e última a ser realizada não incluiu as tarefas de segmentação, mantendo apenas a tarefa de classificação, detecção de objetos e detecção de objetos em vídeos.

A tarefa de análise de cenários é de particular interesse a este trabalho pois foi adaptada posteriormente a outros contextos, sendo denominada na literatura como segmentação semântica e segmentação de instâncias. A segmentação de instâncias difere da segmentação semântica no sentido de identificar diferentes instâncias de um mesmo objeto na mesma imagem. A segmentação semântica o objetivo é apenas diferir entre as classes dos objetos contidos na imagem agrupando-os em uma mesma região, já na segmentação de instâncias o objetivo é isolar as diferentes instâncias de um mesmo objeto em diferentes regiões.

Outra competição relevante na área de Visão Computacional foi o PASCAL VOC

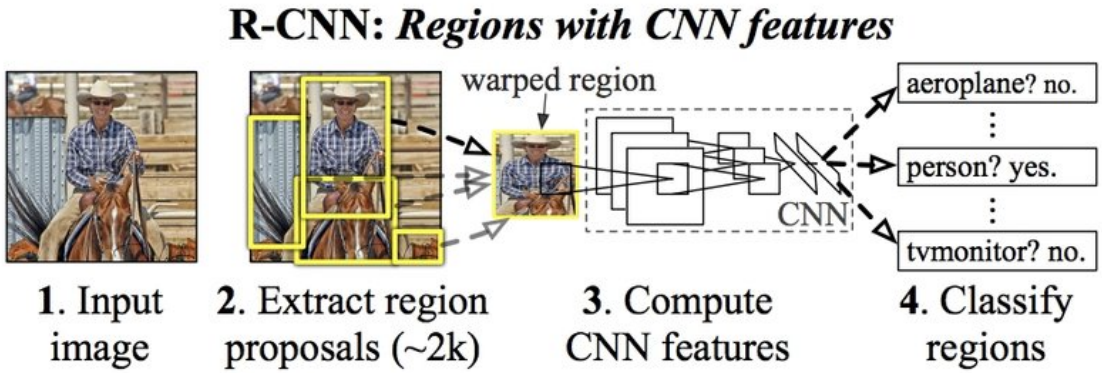
Visual Object Classes. A competição teve edições de 2005 a 2012 e de maneira similar ao *ILSVRC* era composta de tarefas como classificação e localização de objetos, todavia, com um número significativamente inferior de categorias em relação ao *ILSVRC*. Um ponto importante do Pascal VOC foi a incorporação da tarefa de segmentação a partir da edição de 2006 da competição.

De forma a basilar a discussão da evolução do algoritmo de segmentação de imagens utilizado neste estudo, serão discutidos em ordem cronológica as suas arquiteturas precursoras nas seções a seguir.

5.2 R-CNN

Até a re-introdução das Redes Neurais Convolucionais voltadas problema de detecção de objetos as abordagens dominantes consistiam em duas técnicas principais: HOG (DALAL; TRIGGS, 2005) e SIFT (LOWE, 2004). Essas duas técnicas eram aplicadas como extratores de características e posteriormente combinadas com classificadores como as SVMs (CORTES; VAPNIK, 1995). Após o sucesso das CNNs com a AlexNet em 2012, uma ideia promissora seria aplicar a capacidade de extração de características das CNNs ao problema de detecção de objetos. A R-CNN (GIRSHICK et al., 2013) foi um dos primeiros movimentos nessa direção. A Figura 23 exibe as etapas de predição envolvidas na arquitetura da R-CNN.

Figura 23 – R-CNN



Fonte - (GIRSHICK et al., 2013)

A R-CNN é um método de predição de múltiplas etapas: seleção, extração e classificação. O nome R-CNN se deve a divisão das imagens em regiões, do inglês *Region-based Convolutional Neural Networks*. A divisão da imagem ocorre através da técnica de busca seletiva, do ingês *Selective Search* (UIJLINGS et al., 2013). Na primeira etapa do algoritmo, cerca de 2000 regiões são extraídas da imagem.

Cada uma dessas regiões então é redimensionada para ser compatível com as dimensões de entrada de CNNs pré-treinadas no ImageNet. Especificamente, os autores realizaram dois experimentos um deles utilizando a AlexNet apresentado na Seção 4.3 e a VGG apresentada na Seção 4.4.

Após a etapa de extração de características através das CNNs os mapas de características obtidos eram alimentados a múltiplas SVMs lineares utilizando a estratégia um contra todos *One-vs-All*, ou seja, uma SVM linear para cada classe de interesse do problema.

Já a detecção da caixa delimitadora dos objetos é modelada como um problema de regressão a partir dos mapas de características extraídos a partir das regiões iniciais. A partir da descrição da arquitetura não é difícil perceber que tratam-se de operações com custo computacional extremamente elevado. Afinal é preciso aplicar a convolução ao menos 2 mil vezes, uma para cada região extraída pela CNN, isso sem considerar o custo da SVM também aplicada a cada uma das regiões.

Além disso, boa parte das regiões extraídas na primeira etapa tem sobreposição, o que leva a cálculos redundantes. Apesar disso a detecção de objetos observou um salto de desempenho com a abordagem. A questão do custo computacional seria endereçada em trabalhos subsequentes.

5.3 Fast R-CNN

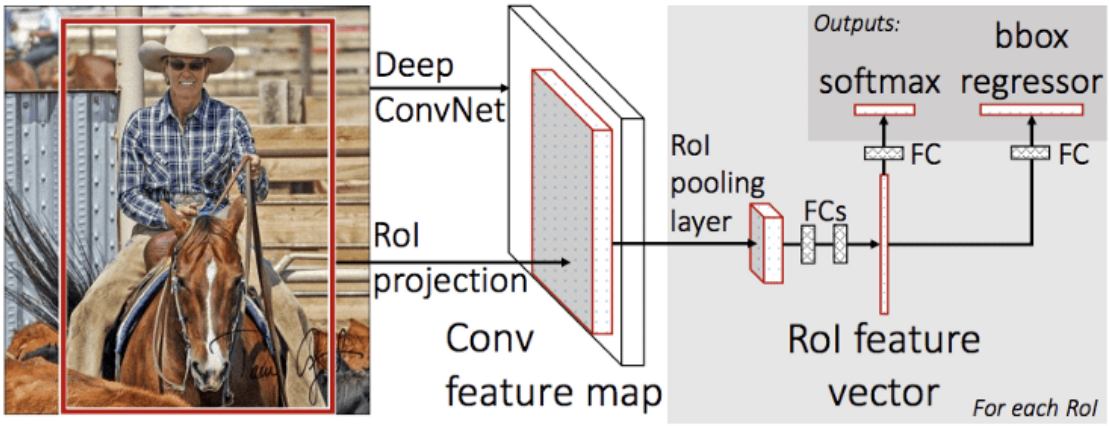
O principal gargalo da R-CNN era a quantidade de cálculos redundantes. Para evitar o problema e acelerar as R-CNNs a SPPNet (HE et al., 2014) realizava a aplicação da CNN uma única vez para a imagem inteira e posteriormente operações de seleção de regiões no mapa de característica resultando em sub-regiões de interesse.

Dessa forma, é possível evitar os cálculos redundantes. Todavia, a abordagem da SPPNet, assim como a R-CNN era um sistema de predição com múltiplas etapas diferentes: seleção, extração e classificação utilizando SVMs. Isso força o congelamento da CNN que atua apenas como extrator de características levando a um desempenho menor no caso da necessidade de ajustes finos para problemas que divergem muito dos dados originais do conjunto ImageNet.

A partir desses problemas, a nova iteração da R-CNN, a Fast R-CNN (GIRSHICK, 2015) propõe uma arquitetura com uma única etapa permitindo o treinamento do modelo fim a fim. A Fast R-CNN introduziu outro elemento, a camada de *RoI Pooling*.

Essa camada é responsável por extrair um vetor de comprimento fixo a partir de uma região do mapa de características obtidos, que é alimentado a camadas totalmente conectadas gerando um vetor de características que então é enviado a duas cabeças de

Figura 24 – Fast R-CNN



Fonte - (GIRSHICK, 2015)

predição, um regressor para as caixas delimitadoras e um classificador Softmax. A Figura 24 ilustra a arquitetura da uma Fast R-CNN.

Assim, o treinamento da arquitetura agora passa a ser passível de execução fim a fim, permitindo o ajuste fino mesmo das camadas iniciais do extrator de características. A função de custo dessa arquitetura é dado pela soma das funções de custo de cada função objetivo distinta para o regressor e o classificador, configurando assim uma função de custo multi-objetivo para a arquitetura.

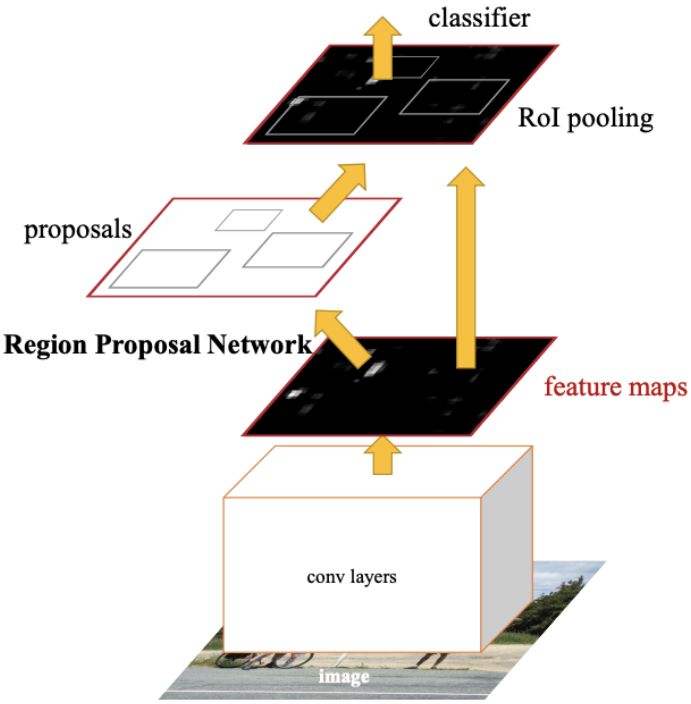
5.4 Faster R-CNN

Apesar dos avanços obtidos pelo Fast R-CNN, a operação de seleção através da técnica de Selective Search ainda era utilizada. As implementações do técnica geralmente utilizada na pesquisa são baseadas em CPU, logo, não se beneficiam do poder computacional paralelo oferecido pelas GPUs modernas. O tempo médio gasto no cálculo das regiões utilizando o *Selective Search* é de 2 segundos por imagem, configurando a maior parte do tempo gasto na aplicação do algoritmo.

Os autores dessa nova iteração das R-CNNs, a Faster R-CNN (REN et al., 2015), argumentam que seria possível implementar as operações de *Selective Search* em GPUs. Todavia, isso não aproveitaria de maneira adequada as melhorias de compartilhamento de cálculos realizados anteriormente. Ao invés disso, os autores propuseram uma nova maneira de realizar a seleção de regiões importantes na imagem através de uma RPN *Region Proposal Network*.

A RPN é uma Rede Neural Convolucional que toma como entrada uma imagem, calcula os mapas de características utilizando o tronco de uma CNN pré-treinada no conjunto ImageNet e obtém como saída regiões retangulares associadas a uma pontuação

Figura 25 – Region Proposal Network



Fonte - (REN et al., 2015)

de "objetividade"*(objectness)*. A Figura 25 ilustra a arquitetura de uma RPN. Essa pontuação representa qual a probabilidade da região retangular extraída conter um objeto de interesse ou conter apenas o plano de fundo de um objeto.

Dessa forma, é possível aproveitar os mapas de características que são extraídos uma única vez e reutilizá-los para múltiplos fins. As RPNs produzem múltiplas regiões de interesse, muitas vezes, regiões redundantes e sobrespostas. Para evitar esse problema a técnica de *Non-Max Supression, NMS* é utilizada para fundir subregiões em uma única região contígua quando há uma taxa de sobreposição significativa (>0.7) calculada a partir da métrica de Intersecção sobre a União *Intersection over Union (IoU)*.

O cálculo da função de custo para treinamento da arquitetura é a soma das 4 funções de custo para cada um dos objetivos, os regressores das regiões de interesse da RPN, o classificador de objetividade da região, o regressor das caixas delimitadoras da Fast R-CNN e o classificador do objeto contido na caixa delimitadora identificada.

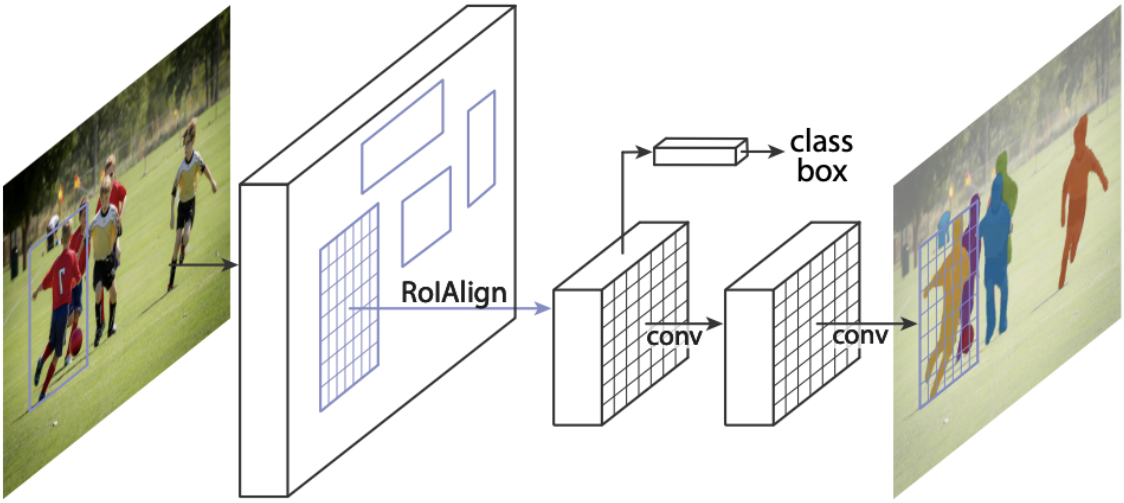
5.5 Mask R-CNN

Além da detecção e classificação de objetos, um outro problema relevante no contexto de processamento de imagens é a segmentação semântica. A segmentação semântica consiste na classificação dos pixels em regiões contíguas pertencentes a um objeto.

A Mask R-CNN (HE et al., 2017) extende a arquitetura da Faster R-CNN adicionando uma nova cabeça de predição responsável por gerar uma máscara sobre cada objeto detectado em paralelo as cabeças de predição da classe e das caixas delimitadoras.

A adição de uma cabeça de predição de máscara adiciona pouco custo computacional a predecessora Faster R-CNN, mantendo o ganho de desempenho obtido nas iterações anteriores da arquitetura. A Figura 26 ilustra a arquitetura da Mask R-CNN.

Figura 26 – Mask R-CNN



Fonte - (HE et al., 2017)

Além da cabeça de predição adicional, a Mask R-CNN também introduziu uma camada intermediária chamada de *ROI Align*. A ideia dessa camada intermediária é alinhar as regiões espaciais da imagem original as regiões correspondentes no mapa de caracterísitcas.

Conforme discutido anteriormente, a extração de mapas de características através de CNNs reduzem as dimensões espaciais como altura e largura da imagem e aumentam a profundidade da representação.

Para realizar o alinhamento da região da imagem original com um mapa de características com altura e largura potencialmente menores os autores utilizaram a interpolação bilinear. A interpolação bilinear é uma operação utilizada em visão computacional para redimensionamento de imagens.

A função objetivo otimizada para a cabeça adicional da arquitetura é um problema binário modelado através de uma sigmóide. Dessa forma, a saída dessa cabeça de predição é uma matriz com as dimensões espaciais da imagem original e a cada pixel obtido um valor de saída é obtido entre 0 e 1 após a aplicação da função sigmóide. Caso o valor seja maior do que 0.5 o pixel é considerado na máscara. Para cada instância de objeto detectado na imagem uma máscara binária é gerada no processo.

Além dos resultados obtidos na segmentação de instâncias a Mask R-CNN se mostrou relevante em outro problema: a estimação de pose. A estimação de pose consiste em classificar a posição das articulações de uma pessoa e identificar qual a posição de cada um de 14 pontos-chave do corpo humano. O sucesso da Mask R-CNN em duas novas tarefas consolidou a família das R-CNNs como métodos extremamente importantes na literatura de visão computacional.

6 Revisão Sistemática da Literatura (RSL)

Neste Capítulo é apresentada uma Revisão Sistemática da Literatura e os resultados obtidos para entender as técnicas utilizadas na fenotipagem de animais na Piscicultura. A RSL é uma modalidade de pesquisa que segue protocolos específicos e tem por objetivo coletar e entender um grande corpus documental sobre determinado tema a fim sistematizar a análise e facilitar o entendimento sobre um assunto em específico.

A revisão sistemática é considerada um estudo secundário, pois utiliza estudos primários como fonte de dados. Como a revisão sistemática obedece a um protocolo isso possibilita a reprodutibilidade do estudo por outros pesquisadores pois é necessário documentar as fontes de dados utilizadas no processo, bem como a metodologia utilizada.

Uma revisão sistemática configura um alicerce para novas atividades de pesquisa acerca de determinado tema. Este capítulo tem por objetivo realizar um levantamento bibliográfico sobre quais as técnicas tem sido utilizadas na fenotipagem de animais na Piscicultura.

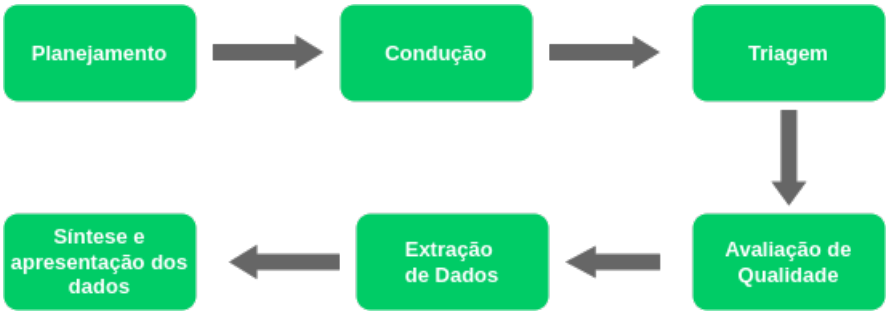
Para atingir este objetivo, foi realizada uma revisão sistemática, cujos resultados são apresentados nessa Seção. Existem diversos tipos de revisão sistemática da literatura (F., 2019) mas de maneira geral é possível delinear as etapas principais que compõem uma RSL.

- **Planejamento:** Nesta etapa, a finalidade é determinar qual o objetivo da pesquisa e quais são as questões a serem respondidas;
- **Condução da Revisão:** Esta etapa tem por objetivo construir strings de busca e realizar um levantamento em bases de dados científicas de quais são os trabalhos relevantes para determinado tema;
- **Triagem dos Estudos levantados:** Nesta etapa, os estudos selecionados são filtrados de acordo com critérios de inclusão e exclusão de maneira a refinar os resultados obtidos na etapa anterior;
- **Avaliação da Qualidade:** Após uma triagem dos estudos selecionados uma avaliação mais minuciosa visa filtrar trabalhos que não apresentam um grau de qualidade adequado a analise pretendida;
- **Extração dos Dados:** Nesta etapa, o foco é extrair informações-chave dos trabalhos de forma a possibilitar uma análise comparativa;

- **Síntese das evidências e apresentação dos dados:** Nesta etapa, as informações são sintetizadas e extraídas na etapa anterior e posteriormente são apresentadas essas informações a fim de facilitar a discussão dos resultados obtidos;

Na Figura 27 é possível observar as etapas que compõem o protocolo descrito.

Figura 27 – Etapas do protocolo do RSL



Fonte - Elaborado pelo Autor

6.1 Planejamento

Nesta Seção é apresentado o protocolo utilizado no desenvolvimento deste capítulo, com as questões que serão respondidas, bases de dados científicas que foram utilizadas, critérios de inclusão e exclusão (triagem), métodos de avaliação e dados que foram extraídos e sintetizados no decorrer da RSL.

O objetivo desta revisão é identificar na literatura a existência de estudos que proponham a aplicação de técnicas de fenotipagem de animais baseadas em imagens digitais no contexto da Piscicultura e Maricultura. A Maricultura se refere a criação de animais marinhos, a extensão dessa revisão sistemática a Maricultura se deve a escassez de trabalhos na área de computação especificamente para a fenotipagem na Piscicultura. O capítulo tem por objetivo secundário analisar também quais as abordagens presentes na literatura com relação ao desenvolvimento e à aplicação de fenotipagem automática na Piscicultura e Maricultura.

As bases de dados científicas indexadas utilizadas na elaboração deste trabalho são enumeradas abaixo.

- *IEEE Xplore*
- *ACM Digital Library*
- *Science Direct*
- *Scopus*

Com base nisso, foram determinadas as questões de pesquisa. Dessa forma, essas questões podem ser derivadas do modelo PICOC do inglês (Population, Intervention, Comparation, Outputs and Context), cujo objetivo é determinar termos bases para as questões de pesquisa. Assim, os parâmetros do modelo PICOC para o presente trabalho foram definidos da seguinte forma:

- **População (*Population*):** Artigos sobre a aplicação de Visão Computacional ou Processamento de Imagens aplicadas a medição ou estimativa de peso de animais na Piscicultura.
- **Intervenção (*Intervention*):** técnicas, conceitos e metodologias;
- **Controle (*Control*):** Artigos, teses e publicações científicas obtidas de bases indexadas, revisões sistemáticas anteriores;
- **Resultados (*Outputs*):** sistemas de medição de animais ; e
- **Aplicação (*Context*):** aplicações de medição automática na área de Piscicultura ou Maricultura.

A partir das definições apresentadas anteriormente, são definidas as questões centrais a serem respondidas.

- **Q1:** Quais as espécies estudadas nos estudos levantados?
- **Q2:** Quais são os problemas ou tarefas resolvidas nos estudos levantados?
- **Q3:** Os resultados obtidos nos estudos são reproduzíveis?
- **Q4:** Quais são os métodos e técnicas aplicadas a extração automática de medidas do trato físico de peixes através de informações imagéticas no contexto da Piscicultura?

Cada questão tem por objetivo mapear os métodos e técnicas aplicadas a extração automática de medidas na Piscicultura. A partir dessas questões, foram determinadas as palavras-chaves e a *string* utilizada para realizar a busca por estudos relacionados nas bases de dados citadas. Com isso, na próxima Seção, é apresentada a etapa de condução do presente trabalho.

6.2 Condução

A etapa de condução em uma revisão sistemática tem como objetivo delinear a seleção dos trabalhos a serem considerados. Essa seleção é construída inicialmente a partir de palavras-chaves, que servem como base para a elaboração da *string* de busca nas bases

indexadas. Dessa forma, as palavras utilizadas no presente trabalho são apresentadas a seguir:

(("aquaculture"OR "mariculture"OR "fish") AND ("computer vision"OR "deep learning"OR "image segmentation"OR "semantic segmentation") AND ("measurement"OR "phenotyping"OR "morphometric"))

Para auxiliar na pesquisa, foi utilizada a ferramenta StArt que foi desenvolvida para dar suporte às Revisões Sistemáticas nas áreas relacionadas a Engenharia de Software (GALVãO; RICARTE, 2010). Dessa forma, por meio dessa ferramenta, o protocolo e as iterações de refinamento de buscas foram registradas. Para refinar as buscas a partir das palavras-chave definidas inicialmente a ferramenta de busca de cada base foi utilizada.

As palavras-chave dentro da *string* foram escritas na língua inglesa. Com isso, na Tabela 1 são apresentadas as quantidades de documentos retornados em cada base, e a quantidade total obtida.

Tabela 1 – Quantidade de documentos selecionados nas bases científicas

Base científica	Documentos selecionados
IEEE Xplore	165
ACM Digital Library	496
Scopus	122
Science Direct	97
Total	880
Selecionados	52
ACEITOS	33

Fonte: Produzido pelo autor.

Do total obtido, foram selecionados 52 documentos. Além da leitura do título, do resumo e da conclusão, a aceitação desses documentos foi obtida por meio dos seguintes critérios de inclusão:

- Documentos em português ou inglês;
- Documentos entre 2017 e 2024; e
- Documentos relacionados ao tema proposto.
- Documentos que discorram sobre Visão Computacional ou Processamento de Imagens para estimar caracteristicas do trato físico de animais na Piscicultura ou Maricultura.
- Documentos que apliquem técnicas de deep learning na Piscicultura ou Mariculutra

Em contrapartida, para realizar a exclusão dos demais documentos, foram considerados os seguintes critérios:

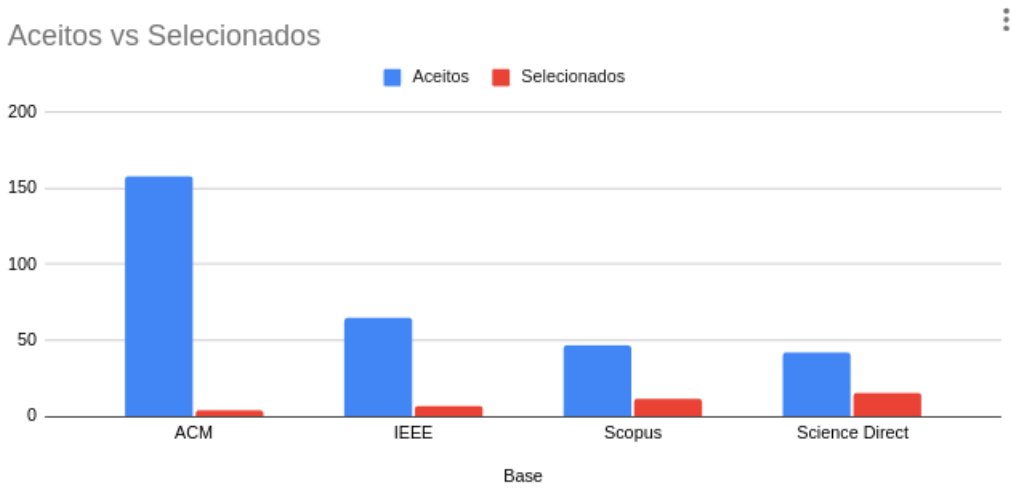
- Documentos anteriores a 2017;
- Documento não disponível para *download*;
- Documento é um tutorial, sumário de evento, pôster ou qualquer outro documento incompleto;
- Documento duplicado;
- Documento apresenta estudo irrelevante para a pesquisa;
- Artigo utiliza dados de sensores ou outra fonte de informação que não se limite a imagens digitais;
- Artigos que realizam a medição dos animais In-situ;
- Aplicação de Visão Computacional em áreas diferentes da Piscicultura ou Maricultura;
- Documento não é sobre Visão Computacional, Processamento de Imagens ou Aprendizagem de Máquina; e
- Documento não apresenta métodos ou técnicas.
- Documento não apresenta ao menos uma das tarefas de interesse do estudo como medição ou estimativa de peso.

A Figura 28 mostra a quantidade de artigos que foram aceitos em cada base científica. A base *Science Direct* foi a que apresentou a maior quantidade de estudos aprovados (13 estudos) de acordo com os critérios apresentados seguida pela *Scopus* (10 estudos) com *IEEE Xplore* e *ACM*, 7 e 3 documentos respectivamente. Os resultados obtidos dos documentos aceitos serão discutidos no Capítulo 7.

6.3 Resultados da RSL

A etapa de sumarização de uma RSL consiste em apresentar os resultados obtidos com base na pesquisa realizada nas etapas anteriores. Deste modo, este capítulo apresenta a análise conduzida com os 33 documentos selecionados por meio da RSL realizada. As análises foram conduzidas a partir das questões elaboradas na Seção 6.1.

Figura 28 – Artigos aceitos por base

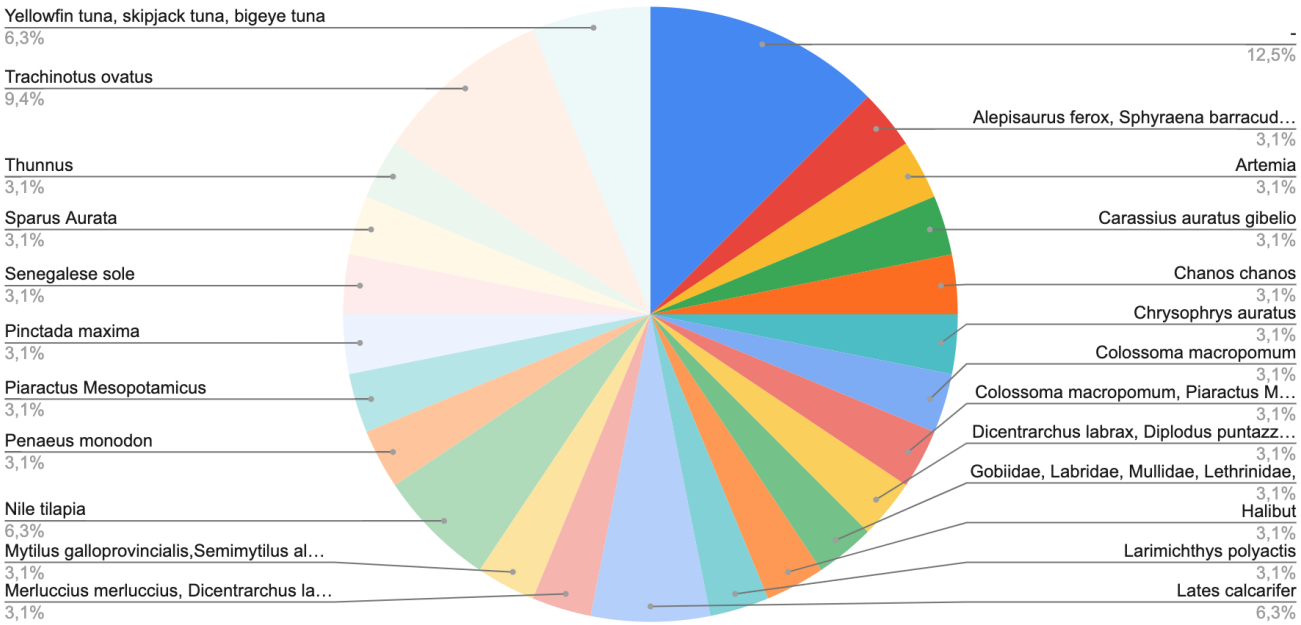


Fonte - Produzida pelo autor

6.3.1 Das espécies estudadas nos documentos - Q1

A primeira análise conduzida foi a avaliação da distribuição das espécies estudadas em cada documento apresentada na Figura 29. Durante esta análise foi constatado uma diversidade bastante grande de espécies nos documentos. A especie mais frequente observada foi a *Lates calcarifer* sendo o objeto de estudo de 4 documentos, seguida pela *Trachinotus ovatus* com 2 documentos; 2 documentos relatam resultados em 5 espécies distintas no mesmo estudo *Dicentrarchus labrax, Diplodus puntazzo, Merluccius merluccius, Sparus aurata.*

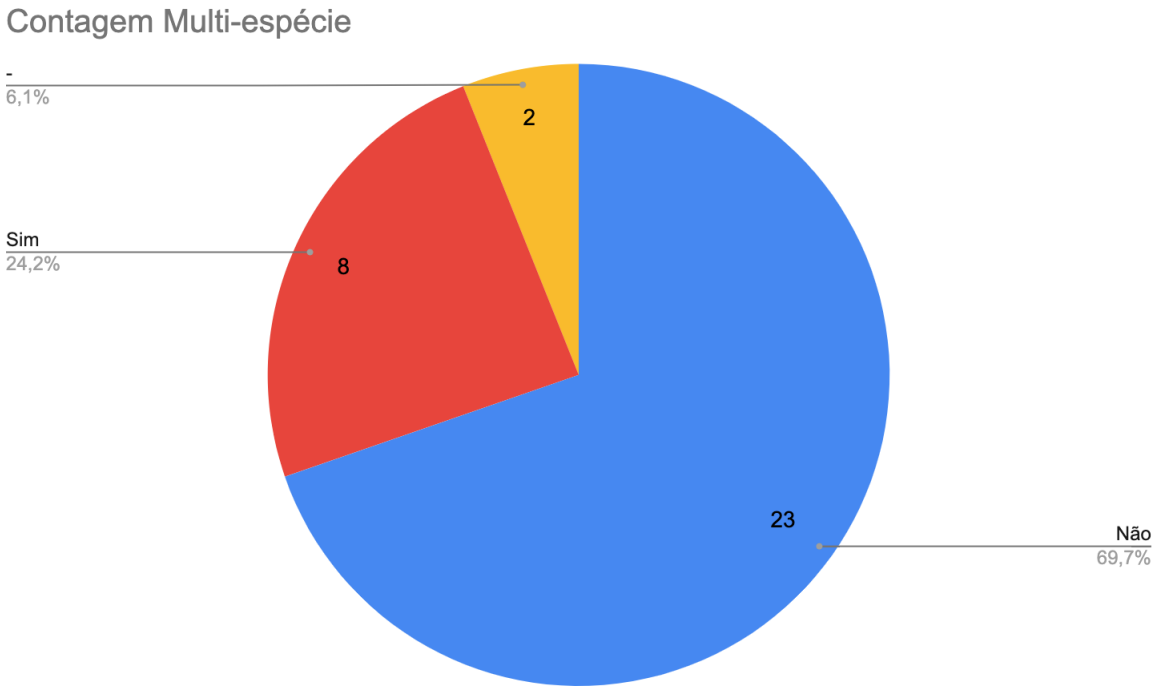
Figura 29 – Percentual de espécies estudadas



Fonte - Produzida pelo autor

Em relação a quantidade de espécies distintas avaliadas por estudo 69,7% (23 documentos) avaliam uma única espécie 24,2% (8 documentos) avaliando múltiplas espécies e 2 documentos não relatam qual espécie foi estudada conforme representado pela Figura 30. Na Tabela 2 são apresentados as espécies presentes em cada um dos estudos aceitos na RSL.

Figura 30 – Percentual de estudos que consideram múltiplas espécies



Fonte - Produzida pelo autor

Tabela 2 – Estudos por espécie

Estudo	Espécie
(TSENG; HSIEH; KUO, 2020)	Alepisaurus ferox
	Sphyraena barracuda
	Brama japonica
	Coryphaena hippurus
	Lampris guttatus
	Lepidocybium flavo-brunneum
	Ruvettus pretiosus
	Trachipterus ishikawae
	Acantho-cybium solandri

(COELHO-CARO et al., 2018)	Mytilus galloprovincialis, Semimytilus algosus Aulacomya ater, Mytilus chilensis Choromytilus Chorus
(PETRELLIS, 2021b)	Dicentrarchus labrax, Diplodus puntazzo Merluccius merluccius,Sparus aurata
(YANG et al., 2021)	Chrysophrys auratus
(WANG et al., 2018)	Halibut
(HAO; YIN; LI, 2022)	Carassius auratus gibelio
(PETRELLIS, 2021a)	Merluccius merluccius , Dicentrarchus labrax Sparus aurata , Diplodus puntazzo
(ANDRIALOVANIRINA et al., 2020)	Gobiidae, Labridae, Mullidae Lethrinidae, Pomacentridae, Scaridae Serranidae, Siganidae
(WANG et al., 2019b)	-
(ISLAMADINA et al., 2018)	-
(KONOVALOV et al., 2018)	Lates calcarifer
(LEKUNBERRI et al., 2022)	Yellowfin tuna, skipjack tuna bigeye tuna
(TANHIR; IQBAL, 2024)	Yellowfin tuna, skipjack tuna bigeye tuna
(LAPICO et al., 2019)	Pinctada maxima
(RIDHWAN et al., 2019)	Thunnus
(QASHLIM et al., 2019)	Chanos chanos
(JONGJARAUNSUK; TAPARHUDEE, 2021)	Lates calcarifer
(FERNANDES et al., 2020)	Nile tilapia
(KONOVALOV et al., 2017)	Lates calcarifer
(WANG; Van Stappen; De Baets, 2020)	Artemia
(YU et al., 2020)	Trachinotus ovatus
(YU et al., 2021)	Trachinotus ovatus
(MARCO et al., 2021)	Senegalese sole
(FENG et al., 2023)	Nile tilapia
(XUE et al., 2023)	Sparus Aurata
(FREITAS et al., 2023)	Piaractus Mesopotamicus
(YU et al., 2023)	Trachinotus ovatus
(YU; ZHANG; YUAN, 2023)	-

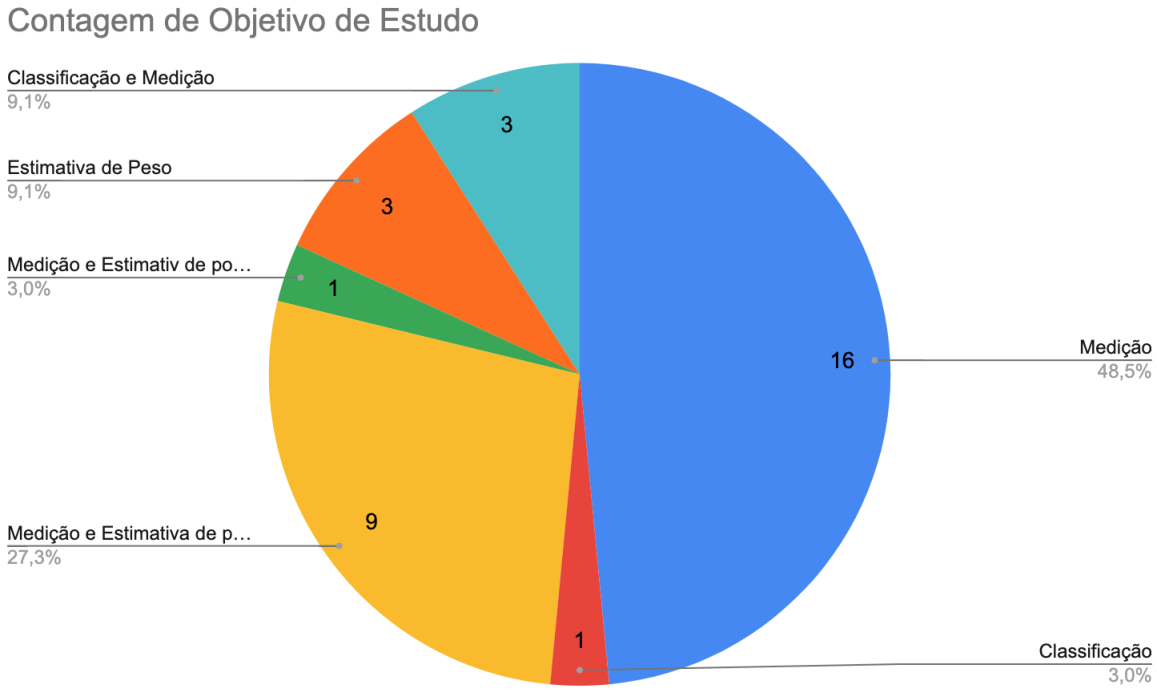
(ZENG et al., 2024)	Larimichthys polyactis
(SHIBATA et al., 2024)	-
(SALEH et al., 2023)	Penaeus monodon
(ARIEDE et al., 2023a)	Colossoma macropomum
(BATISTA; BREGA, 2024)	Colossoma macropomum Piaractus Mesopotamicus

6.3.2 Das tarefas de cada estudo - Q2

A análise conduzida na sequência foi qual a principal tarefa ou objetivo relatada no documento. Neste estudo, o objetivo é a medição ou estimativa de peso de animais. A partir da leitura dos documentos foi observado que alguns estudos se dedicam a mais de uma dessas tarefas, portanto, uma análise foi conduzida nesse sentido.

Das tarefas incluídas no protocolo desse estudo estão a medição e a estimativa de peso, no entanto, alguns documentos tratam sobre a classificação de espécie e estimativa de pontos-chave (Keypoint Estimation). Conforme apresentado na Figura 31, a tarefa mais frequente nos estudos é a medição presente em 48,5% (16 documentos) dos trabalhos analisados, seguida pela medição e estimativa de peso simultaneamente em 27,3% (9 documentos) dos casos. Os demais documentos apresentam combinações das duas principais tarefas consideradas ou a combinação com tarefas fora do escopo deste trabalho. A tabela 3 elenca as tarefas mapeadas em cada trabalho selecionado.

Figura 31 – Percentual de tarefas consideradas



Fonte - Produzida pelo autor

Tabela 3 – Estudos por tarefa abordada

Estudo	Objetivo do Estudo
(KONOVALOV et al., 2018)	Medição
(LAPICO et al., 2019)	Medição
(COELHO-CARO et al., 2018)	Medição

(WANG et al., 2019b)	Medição
(ISLAMADINA et al., 2018)	Medição e Est. de peso
(WANG et al., 2018)	Medição
(PETRELLIS, 2021a)	Medição e Landmark Est.
(YANG et al., 2021)	Medição
(ANDRIALOVANIRINA et al., 2020)	Medição
(HAO; YIN; LI, 2022)	Estimativa de Peso
(RIDHWAN et al., 2019)	Medição
(QASHLIM et al., 2019)	Estimativa de Peso
(JONGJARAUNSUK; TAPARHUDEE, 2021)	Estimativa de Peso
(PETRELLIS, 2021b)	Classificação e Medição
(FERNANDES et al., 2020)	Medição e Estimativa de peso
(TSENG; HSIEH; KUO, 2020)	Classificação e Medição
(KONOVALOV et al., 2017)	Medição
(WANG; Van Stappen; De Baets, 2020)	Medição
(YU et al., 2021)	Medição
(LEKUNBERRI et al., 2022)	Classificação e Medição
(MARCO et al., 2021)	Medição e Estimativa de peso
(FENG et al., 2023)	Medição e Estimativa de peso
(XUE et al., 2023)	Medição e Estimativa de peso
(FREITAS et al., 2023)	Medição e Estimativa de peso
(YU; ZHANG; YUAN, 2023)	Medição
(TANHIR; IQBAL, 2024)	Medição
(ZENG et al., 2024)	Medição
(SHIBATA et al., 2024)	Medição
(SALEH et al., 2023)	Medição e Estimativa de peso
(ARIEDE et al., 2023b)	Medição e Estimativa de peso
(BATISTA; BREGA, 2024)	Medição

6.3.3 Reproducibilidade de Resultados - Q3

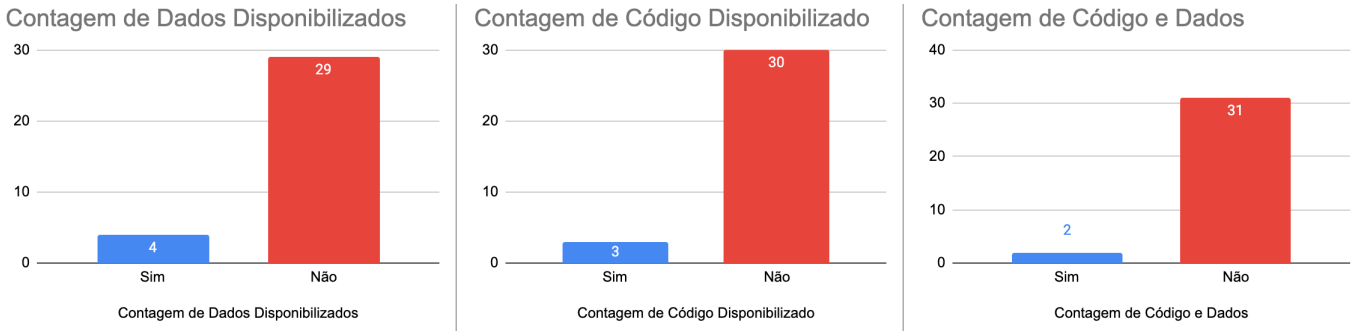
Uma outra questão relevante analisada foi a reproducibilidade dos resultados reportados nos estudos avaliados. A reproducibilidade consiste na capacidade de outros pesquisadores serem capazes de reproduzir os mesmos resultados reportados em um trabalho científico pelos autores (GUNDERSEN; KJENSMO, 2018), (IVIE; THAIN, 2018).

Algumas questões são relevantes nesse sentido para permitir a reproducibilidade de resultados em um estudo, entre elas estão a disponibilização dos conjuntos de dados utilizados nesses experimentos, a disponibilização do código ou ferramental utilizado na condução do experimento e um detalhamento adequado das etapas executadas durante o processo experimental.

Portanto, a análise a seguir observa sob essa ótica, duas dessas questões que são mais facilmente quantificáveis: a disponibilização do código e do conjunto de dados utilizado. Para analisar a terceira questão postulada, do nível de detalhamento das etapas do processo experimental seria necessária reprodução do experimento original, o que foge do escopo deste estudo.

Dos documentos avaliados 12,12% (4 estudos) disponibilizaram os conjuntos de dados utilizados durante os experimentos e 9,09% (3 estudos) disponibilizaram também o código utilizado na experimentação. Todavia, conforme ilustra a Figura 32, apenas 6,06% (2 estudos) disponibilizaram simultaneamente o código e os dados reportados no documento. Na Tabela 4 constam os resultados acerca da reproducibilidade dos estudos levantados.

Figura 32 – Reproducibilidade dos Resultados



Fonte - Produzida pelo autor

Tabela 4 – Estudos com código e conjunto de dados disponibilizados

Estudo	Dados Disp.	Código Disp.
(KONOVALOV et al., 2018)	✓	✗
(LAPICO et al., 2019)	✗	✓

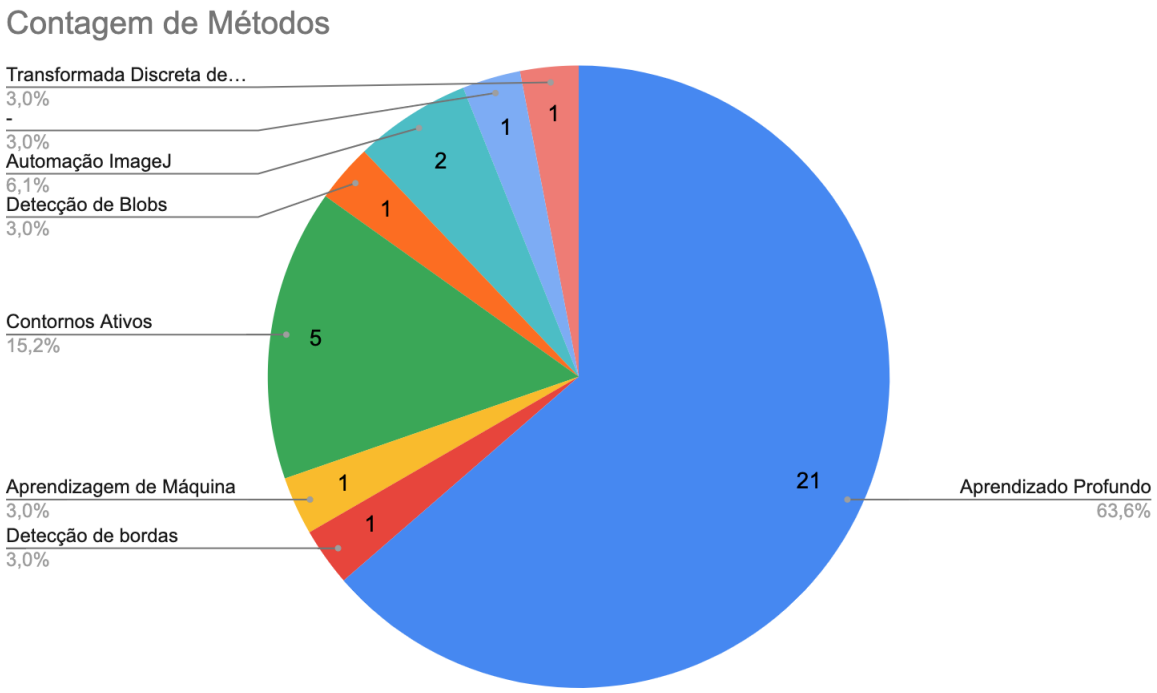
(COELHO-CARO et al., 2018)	✗	✗
(WANG et al., 2019b)	✗	✗
(ISLAMADINA et al., 2018)	✗	✗
(WANG et al., 2018)	✗	✗
(PETRELLIS, 2021a)	✗	✗
(YANG et al., 2021)	✗	✗
(ANDRIALOVANIRINA et al., 2020)	✗	✗
(HAO; YIN; LI, 2022)	✗	✗
(RIDHWAN et al., 2019)	✗	✗
(QASHLIM et al., 2019)	✗	✗
(JONGJARAUNSUK; TAPARHUDEE, 2021)	✗	✗
(PETRELLIS, 2021b)	✗	✗
(FERNANDES et al., 2020)	✗	✗
(TSENG; HSIEH; KUO, 2020)	✗	✗
(KONOVALOV et al., 2017)	✗	✓
(WANG; Van Stappen; De Baets, 2020)	✓	✗
(YU et al., 2020)	✓	✓
(YU et al., 2021)	✓	✗
(LEKUNBERRI et al., 2022)	✗	✗
(MARCO et al., 2021)	✗	✗
(FENG et al., 2023)	✗	✗
(XUE et al., 2023)	✗	✗
(FREITAS et al., 2023)	✗	✗
(YU et al., 2023)	✗	✗
(YU; ZHANG; YUAN, 2023)	✗	✗
(TANHIR; IQBAL, 2024)	✗	✗
(ZENG et al., 2024)	✗	✗
(SHIBATA et al., 2024)	✗	✗
(SALEH et al., 2023)	✗	✗
(ARIEDE et al., 2023a)	✗	✗
(BATISTA; BREGA, 2024)	✗	✗

6.3.4 Métodos e Técnicas utilizadas na resolução das tarefas - Q4

Por fim a última questão avaliada nos trabalhos foram as técnicas empregadas na resolução das tarefas elencadas anteriormente. O rol de técnicas presentes nos estudos avaliados são bastante diversas, por conta disso classifica-se as abordagens em técnicas de Aprendizagem de Máquina clássicas, técnicas baseadas em aprendizado profundo, técnicas baseadas em detecção de bordas, técnicas baseadas em *active contours*, detecção de *blobs* e macros (automações) de software especializados em medição.

A partir dessa análise é possível notar uma tendência no uso de técnicas baseadas em aprendizado profundo conforme representado pela Figura 33, o que condiz com a popularização de algoritmos baseados em Inteligência Artificial também em outras áreas das ciências. O barateamento do custo de hardwares especializados nesse tipo de computação como GPUs e NPUs tem contribuído significativamente para esta tendência.

Figura 33 – Métodos e técnicas empregados



Fonte - Produzida pelo autor

Dos estudos avaliados 63,6% (21 estudos) aplicam técnicas baseadas em Aprendizagem Profunda, seguida de *Active Contours* presente em 15,2% (5 estudos) dos documentos avaliados sendo os demais trabalhos divididos em outras técnicas como Aprendizagem de Máquina clássica 3% (1 estudo), automações 6,1% (2 estudos), detecção de bordas 3% (1 estudo), detecção de *blobs* 3% (1 estudo) e transformada de *Fourier* discreta 3% (1 estudo). A Tabela 5 sumariza as técnicas utilizadas nos estudos selecionados. A Tabela 6 mostra uma matriz de técnicas e soluções presentes nos trabalhos selecionados.

Tabela 5 – Técnicas utilizadas nos estudos

Artigo	**Técnica**
(KONOVALOV et al., 2018)	Deep Learning
(LAPICO et al., 2019)	Edge Detection
(COELHO-CARO et al., 2018)	Machine Learning
(WANG et al., 2019b)	Active Contours
(ISLAMADINA et al., 2018)	Blob Detection
(WANG et al., 2018)	Active Contours
(PETRELLIS, 2021a)	Deep Learning
(YANG et al., 2021)	Deep Learning
(ANDRIALOVANIRINA et al., 2020)	ImageJ Macro
(HAO; YIN; LI, 2022)	Active Contours
(RIDHWAN et al., 2019)	-
(QASHLIM et al., 2019)	Active Contours
(JONGJARAUNSUK; TAPARHUDEE, 2021)	ImageJ Macro
(PETRELLIS, 2021b)	Active Contours
(FERNANDES et al., 2020)	Deep Learning
(TSENG; HSIEH; KUO, 2020)	Deep Learning
(KONOVALOV et al., 2017)	Discrete Fourier Transform
(WANG; Van Stappen; De Baets, 2020)	Deep Learning
(YU et al., 2020)	Deep Learning
(YU et al., 2021)	Deep Learning
(LEKUNBERRI et al., 2022)	Deep Learning
(MARCO et al., 2021)	Deep Learning
(FENG et al., 2023)	Deep Learning
(XUE et al., 2023)	Deep Learning
(FREITAS et al., 2023)	Deep Learning
(YU et al., 2023)	Deep Learning
(YU; ZHANG; YUAN, 2023)	Deep Learning
(TANHIR; IQBAL, 2024)	Deep Learning
(ZENG et al., 2024)	Deep Learning
(SHIBATA et al., 2024)	Deep Learning
(SALEH et al., 2023)	Deep Learning
(ARIEDE et al., 2023a)	Deep Learning
(BATISTA; BREGA, 2024)	Deep Learning

Tabela 6 – Técnicas utilizadas nos estudos

Estudo	Est. de Peso	Medição	Keypoint	Classificação	Mask-RCNN	Edge/Blob
(KONOVALOV et al., 2018)	✗	✓	✗	✗	✗	✓
(LAPICO et al., 2019)	✗	✓	✗	✗	✗	✓
(COELHO-CARO et al., 2018)	✗	✗	✗	✓	✗	✗
(WANG et al., 2019b)	✗	✓	✗	✗	✗	✗
(ISLAMADINA et al., 2018)	✓	✓	✗	✗	✗	✓
(WANG et al., 2018)	✗	✓	✗	✗	✗	✗
(PETRELLIS, 2021a)	✗	✓	✓	✗	✓	✗
(YANG et al., 2021)	✓	✗	✗	✗	✗	✓
(ANDRIALOVANIRINA et al., 2020)	✗	✓	✗	✗	✗	✗
(HAO; YIN; LI, 2022)	✓	✗	✗	✗	✗	✗
(RIDHWAN et al., 2019)	✗	✓	✗	✗	✗	✗
(QASHLIM et al., 2019)	✓	✗	✗	✗	✗	✗
(JONGJARAUNSUK; TAPARHUDEE, 2021)	✓	✗	✗	✗	✗	✗
(PETRELLIS, 2021b)	✗	✓	✗	✓	✗	✗
(FERNANDES et al., 2020)	✓	✓	✗	✗	✗	✓
(TSENG; HSIEH; KUO, 2020)	✗	✓	✗	✓	✗	✗
(KONOVALOV et al., 2017)	✗	✓	✗	✗	✗	✓
(WANG; Van Stappen; De Baets, 2020)	✗	✓	✗	✗	✗	✗
(YU et al., 2020)	✗	✓	✗	✗	✓	✗
(YU et al., 2021)	✗	✓	✗	✗	✗	✗
(LEKUNBERRI et al., 2022)	✗	✓	✗	✓	✓	✗

(MARCO et al., 2021)	✓	✓	✗	✗	✗	✗
(FENG et al., 2023)	✓	✓	✗	✗	✗	✗
(XUE et al., 2023)	✓	✓	✗	✗	✗	✗
(FREITAS et al., 2023)	✓	✓	✗	✗	✓	✗
(YU et al., 2023)	✓	✓	✗	✗	✓	✗
(YU; ZHANG; YUAN, 2023)	✗	✓	✗	✗	✗	✗
(TANHIR; IQBAL, 2024)	✗	✓	✗	✗	✗	✗
(ZENG et al., 2024)	✗	✓	✗	✗	✗	✗
(SHIBATA et al., 2024)	✗	✓	✗	✗	✗	✗
(SALEH et al., 2023)	✓	✓	✗	✗	✗	✗
(ARIEDE et al., 2023a)	✓	✓	✗	✗	✓	✗
(BATISTA; BREGA, 2024)	✗	✓	✗	✗	✓	✗

6.3.5 Discussão da RSL

A maioria dos trabalhos publicados antes de 2020 empregam abordagens de processamento de imagens mais antigas. Apresentadas na Seção 2.5.4 e 2.5.5 as técnicas de detecção de bordas e blobs aparecem em (LAPICO et al., 2019), (WANG et al., 2019a) respectivamente. Outra técnica empregada em mais de 20% dos estudos levantados é o Active Contours, apresentado na seção 2.5.6 e presente em (QASHLIM et al., 2019),(WANG et al., 2018) e (WANG et al., 2019b).

No contexto desse trabalho, os estudos mais relevantes foram publicados após o ano de 2020. Esses estudos abordam majoritariamente o problema da fenotipagem através da aplicação de Redes Neurais Convolucionais.

Alguns dos estudos levantados aplicam arquiteturas convolucionais customizadas combinando redes perceptron multicamadas para modelar regressões das medidas obtidas diretamente do mapa de características da imagem como em (FERNANDES et al., 2020), (YANG et al., 2021) e (XUE et al., 2023).

Outros estudos realizam as medições em 2 etapas segmentando a imagem em uma primeira etapa e posteriormente pós-processam os resultados com algum esquema de calibração. Nessa categoria existe um grupo de estudos que se baseiam na arquitetura de segmentação de imagens U-Net (RONNEBERGER; FISCHER; BROX, 2015) como os trabalhos (WANG; Van Stappen; De Baets, 2020) e (YU et al., 2021). Todavia, a arquitetura de redes neurais convolucionais mais prevalente para modelagem desse tipo de problema é a Mask R-CNN que aparece em diversos dos trabalhos levantados neste estudo (YU et al., 2023), (BATISTA; BREGA, 2024), (PETRELLIS, 2021a),(YU et al., 2020), (FREITAS et al., 2023), (ARIEDE et al., 2023a),(LEKUNBERRI et al., 2022). A partir dos resultados obtidos pela RSL selecionamos a abordagem em 2 etapas para o presente trabalho. No Capítulo 7 a abordagem adotada é discutida e os resultados obtidos são apresentados.

7 Experimentos e Resultados

7.1 Introdução

A partir da Revisão Sistemática da Literatura conduzida foi possível observar uma predominância e tendência do uso de Aprendizagem Profunda para a solução do problema de fenotipagem automática. Conforme observado na Figura 33 no Capítulo 6.

Todavia, apenas 2 trabalhos levantados aplicam a solução de fenotipagem para extração de informações de sub-regiões de interesse dos animais como cabeça, nadadeiras e corpo. Além disso, nenhum dos trabalhos levantados abordam duas das espécies predominantes na Piscicultura Comercial brasileira o *Piaractus Mesopotamicus (Pacu)* e o *Colossoma macropomum (Tambaqui)*.

Portanto, esse trabalho tem por objetivo explorar esses dois pontos não cobertos pela literatura recente. Para viabilizar o projeto, uma parceria com o Laboratório de Genética em Piscicultura e Conservação (LaGeAC) da Unesp de Jaboticabal foi estabelecida.

Os especialistas em Piscicultura do laboratório anotaram um conjunto de imagens e construíram um conjunto de treinamento para automatizar o processo de fenotipagem através de aprendizado profundo. A seguir, são formalizados os principais conceitos e tarefas relacionadas ao trabalho desenvolvido.

7.2 Materiais e Métodos

O conjunto de dados utilizados nesse trabalho consistem em cerca de 1802 animais da espécie Pacu e 365 animais da espécie Tambaqui. Devido as diferenças morfológicas em diferentes estágios de crescimento, as imagens do conjunto de ambas as espécies foram capturadas em 2 momentos diferentes, com 15 meses e 28 meses de forma a tornar o modelo invariante ao estágio de desenvolvimento dos animais.

Para a extração das imagens os animais foram submergidos em uma solução de benzocaína (0.1 mg/l), identificados através de uma tag RFC, pesados e acomodados sobre uma base de isopor com uma régua para posterior verificação dos resultados do modelo. Após a coleta, para validação do sistema de medição, as imagens de um conjunto independente não utilizado durante o treinamento foram calibradas individualmente. A calibração consiste no mapeamento de uma distância de 1 cm na régua presente na imagem para um valor em pixels através do software ImageJ.

O processo de anotação foi realizado através do Software Labelbox (LABELBOX,

2024), por um único anotador e revisados por um segundo especialista. As imagens anotadas incorretamente foram invalidadas e corrigidas até que todas estivessem anotadas corretamente.

Foram anotados nas imagens 4 regiões de interesse, a cabeça, o corpo, as nadadeiras e a região da pelve conforme ilustra a Figura 34

Figura 34 – Máscara anotada com as regiões de interesse, à esq. *Colossoma macropomum* à dir. *Piaractus mesopotamicus*

Fonte - produzida pelo autor

As medidas são extraídas inicialmente em pixels e mapeadas posteriormente para uma medida em milímetros através do mapeamento de escala da medida em pixels correspondente a medida real em milímetros. Para facilitar o processo de validação da extração de medidas, uma régua foi posicionada sobre uma plataforma de isopor e a calibração em pixels foi realizada manualmente em cada imagem do conjunto utilizado para validação dos resultados.

Especificamente, a implementação da Mask R-CNN foi ajustada para aderir às restrições de campo relacionadas ao custo computacional. Para minimizar o estresse dos peixes durante o processo de medição, o modelo treinado precisa ser compacto e rápido o suficiente para rodar em um dispositivo de borda o mais próximo possível dos tanques dos peixes.

O trabalho original da Mask R-CNN emprega uma Rede Neural Convolucional de 50 camadas chamada Resnet-50 como extrator de características. Quanto maior o número de camadas convolucionais maior o custo computacional do modelo. Para reduzir este custo, foi utilizada um variante da arquitetura ResNet com apenas 18 camadas, que efetivamente reduziu o número de operações de ponto flutuante (FP), parâmetros necessários e tempo de treinamento para alcançar resultados similares.

Durante o treinamento, como técnica de *Data Augmentation*, as imagens foram espelhadas verticalmente e horizontalmente, rotacionadas em 90º ou 270º, redimensionadas em escalas de 0.10, 0.25, 0.5 e 0.75 todas com probabilidade de 0.5 de ocorrer, de maneira que o modelo se tornasse mais tolerante em relação a transformações lineares das imagens de entrada.

O conjunto de dados foi dividido aleatoriamente, de maneira estratificada, preservando as proporções das diferentes espécies em três subconjuntos: treinamento (80%), validação (20%) e 5% do conjunto de treinamento foi utilizado como conjunto de desenvolvimento.

O conjunto de desenvolvimento foi usado para ajustar hiperparâmetros durante a experimentação, evitando assim o uso direto do conjunto de validação e evitando assim um possível sobreajuste.

O experimento foi realizado utilizando uma GPU NVIDIA RTX 2060 com 8GB RAM, a CPU utilizada foi um i5 9600f com 6 núcleos físicos. O experimento foi executado por 350 épocas e levou aproximadamente 22h.

A métrica otimizada foi a Interseção sobre União (IoU). Ao final de cada época, a IoU é calculada para o conjunto de desenvolvimento. Durante os experimentos, o treinamento foi interrompido algumas vezes para para otimizar a taxa de aprendizado, que foi o hiperparâmetro com maior sensibilidade no modelo, usando o desempenho no conjunto de desenvolvimento como parâmetro decisório.

O IoU foi calculado usando a área obtida pelo modelo A e a área B esperada (anotada) com a seguinte fórmula $A \cap B / A \cup B$. Um valor de IoU de 1,0 indica que a área anotada sobrepõe exatamente a área obtida como saída pelo modelo.

Com base nas saídas obtidas pela Mask R-CNN, a caixa delimitadora foi utilizada para calcular as medidas morfométricas do peixe em pixels e as máscaras segmentadas foram utilizadas para calcular a área em pixels das regiões de interesse.

Para avaliação dos resultados de medição obtidos pelo modelo duas métricas foram avaliadas, a Correlação de Pearson (r) e o erro médio quadrático (MSE) entre a medida obtida pelo modelo e a medição manual realizada por um especialista. O erro médio quadrático foi escolhido devido a diferença entre o valor obtido pelo modelo e o valor anotado pelo especialista poder se estender para o domínio dos números negativos.

O coeficiente de Pearson é definido pela seguinte fórmula.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{7.1}$$

Já a fórmula que define o erro médio quadrático é dada por:

$$\mathrm{MSE}(y, \hat{y}) = \frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N} \tag{7.2}$$

7.3 Resultados

7.3.1 Treinamento do Modelo

O modelo foi treinado por 350 épocas, com a taxa de aprendizagem fixada em 10^{-5}. A Tabela 7 mostra os valores da IoU para *Precision* e *Recall* no conjunto de testes utilizado calculado para as caixas delimitadoras.Os indíces estão divididos em intervalos e em diferentes escalas de detecção. Os indíces de IoU com valor -1 na tabela indicam que não houve detecções de objetos na escala em questão.

Tabela 7 – Tabela do IoU para as Caixas Delimitadoras

Métrica	**IoU/Size**	**Valor**
Precisão Média (AP)	0.50:0.95 / all	0.941
Precisão Média (AP)	0.50 / all	0.995
Precisão Média (AP)	0.75 / all	0.995
Precisão Média (AP)	0.50:0.95 / small	-1.000
Precisão Média (AP)	0.50:0.95 / medium	0.869
Precisão Média (AP)	0.50:0.95 / large	0.953
Sens. Média (AR)	0.50:0.95 / all	0.960
Sens. Média (AR)	0.50:0.95 / small	-1.000
Sens. Média (AR)	0.50:0.95 / medium	0.906
Sens. Média (AR)	0.50:0.95 / large	0.964

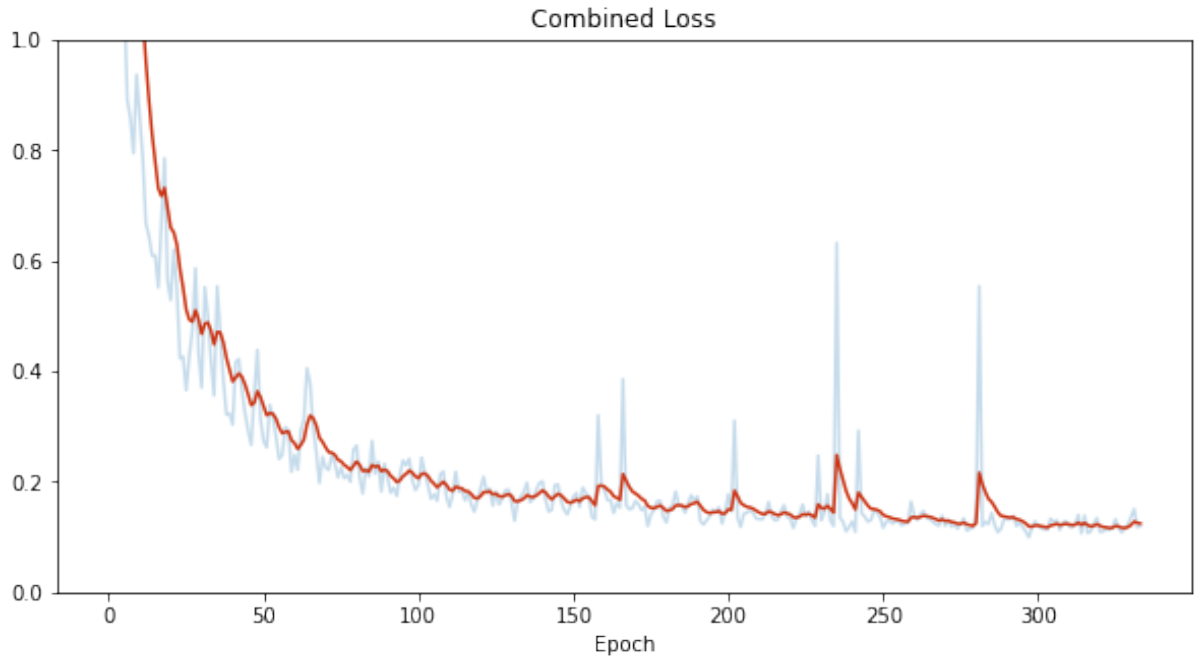
A Tabela 8 mostra os valores da IoU para *Precisão* e *Sensibilidade* no conjunto de testes utilizado calculado para a segmentação

Tabela 8 – Tabela do IoU para a Segmentação

Métrica	**Intervalo**	**Valor**
Precisão Média (AP)	0.50:0.95 / all	0.899
Precisão Média (AP)	0.50 / all	0.995
Precisão Média (AP)	0.75 / all	0.995
Precisão Média (AP)	0.50:0.95 / small	-1.000
Precisão Média (AP)	0.50:0.95 / medium	0.879
Precisão Média (AP)	0.50:0.95 / large	0.921
Sens. Média (AR)	0.50:0.95 / all	0.919
Sens. Média (AR)	0.50:0.95 / small	-1.000
Sens. Média (AR)	0.50:0.95 / medium	0.909
Sens. Média (AR)	0.50:0.95 / large	0.926

As curvas de perda *(loss)* do modelo são apresentadas na Figura 35. A perda total é dada pela soma de todas as funções de custo $L = L_{cls} + L_{box} + L_{mask} + L_{objectness}$ que corresponde a função multi-objetivo otimizada pela Mask R-CNN. A otimização obteve variações significativas até a época 150, após essa etapa o modelo foi otimizado por mais 200 épocas pois as métricas de IoU continuaram melhorando com o modelo sendo otimizado para detalhes mais finos das imagens.

Figura 35 – Função Perda Multi-objetivo da Mask R-CNN



Fonte - produzido pelo autor

7.3.2 Fenotipagem *Piaractus mesopotamicus*

Devido ao processo de coleta das imagens ter sido realizado por equipes e em estágios de desenvolvimento diferentes dos animais, para garantir a qualidade dos resultados reportados neste trabalho as imagens do conjunto utilizado na validação do modelo foram calibradas uma a uma.

Todavia, a calibração se mostrou com baixa variabilidade nesse conjunto, em que 1 cm variou de 46,68 a 51,67 pixels (média de 49,49 pixels). O conjunto de validação das medições para o Pacu continha 1031 animais, desses apenas 979 possuiam valores observados válidos.

Outro ponto importante que vale destacar é que as imagens foram capturadas por dois dispositivos com resoluções diferentes. Anomalias de segmentação foram identficadas em apenas 9 imagens do conjunto que foram descartadas das visualizações e métricas

apresentadas a seguir. Essas imagens apresentaram valores anômalos acima do limite superior de $\mu + 3 * \sigma$ e abaixo do limite inferior μ - $3 * \sigma$ das medições automáticas obtidas.

Para a medição, neste conjunto de testes em particular a medida da pelve foi desconsiderada pois os dados dessa métrica ainda não haviam sido padronizados. Portanto, as medidas da pelve pl *(Pelvis Length)* e ph *(Pelvis Height)* não foram considerados nessa análise em particular.

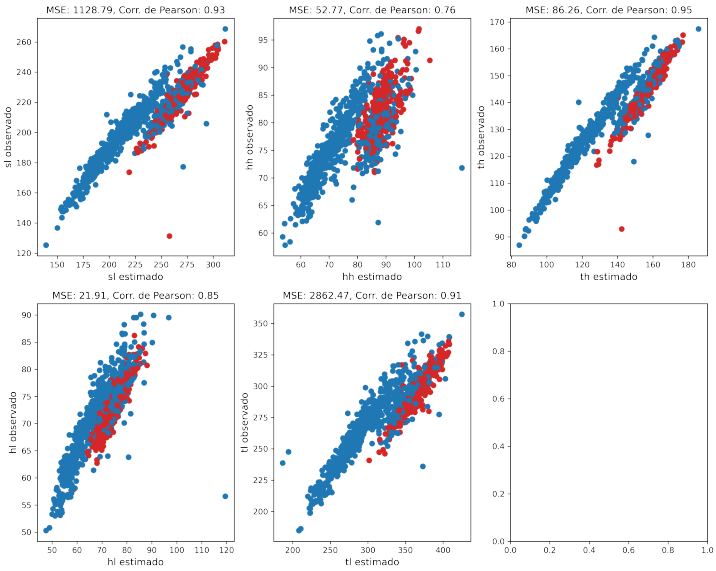
As demais medidas estão padronizadas conforme a Figura 1. A Tabela 9 apresenta as métricas obtidas a partir da comparação da medição automática com a observada por anotadores humanos.

Tabela 9 – Tabela MSE e Coef. de Pearson

Medida	MSE (mm)	Coef. Pearson
tl (Comprimento Total)	2862.47	0.91
th (Altura Total)	86.26	0.95
hh (Altura Cabeça)	52.77	0.76
hl (Comprimento Cabeça)	21.91	0.85
sl (Comprimento Padrão)	1128.79	0.93

A Figura 36 apresenta um gráfico de dispersão onde o eixo X representa o valor estimado pelo sistema e o eixo Y apresenta o valor anotado por um humano.

Figura 36 – Gráfico de Dispersão X = estimado, Y = observado



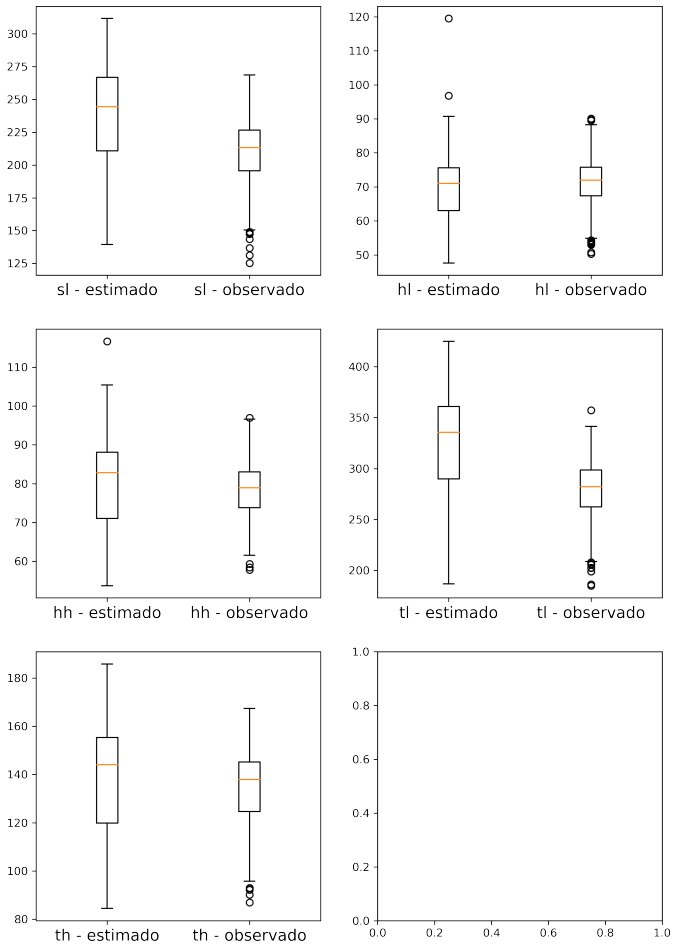
Fonte - produzido pelo autor

A diferenciação de cores se da pelo fato de as imagens terem sido capturadas em momentos diferentes por dispositivos com resoluções distintas.

Pelo fato da calibração ter sido realizada individualmente e o valores convertidos para a mesma escala é possível observá-los no mesmo gráfico. Todavia há um pequeno desvio em relação aos dois grupos, também observados no gráfico a seguir.

Apesar da alta correlação > 0.70 em todas as medições, o erro médio quadrado para as medidas de comprimento tl e sl chama a atenção. Para ilustrar um pouco melhor a questão a Figura 37 mostra *box-plots* comparando os valores obtidos via medição automática e a medição humana.

Figura 37 – Box-Whisker Plot - Observado x Estimado



Fonte - produzido pelo autor

Através desta visualização é possível observar que existe uma variabilidade maior bem como um número maior de valores extremos *(outliers)* nas duas medidas de comprimento, tl e sl, coletadas manualmente por anotadores humanos.

Foi atribuído a esse fenômeno diferenças na metodologia de medição manual. Portanto, os benefícios obtidos a partir da aplicação de um sistema de medição automática vão além da simples automação do processo.

O sistema de medição elimina as diferenças de metodologia entre diferentes anotadores humanos, trazendo, além da automação a padronização dos resultados e maior confiabilidade das informações produzidas neste tipo de operação.

7.3.3 Fenotipagem *Colossoma macropomum*

Na coleta dessa amostra a distância entre a camêra e a plataforma de isopor foi padronizada em 80 cm. O dispositivo de captura utilizado foi padronizado para uma câmera digital de 12 megapixels integrada ao smartphone Motorola One Action smartphone, configurado na resolução máxima e foco automático.

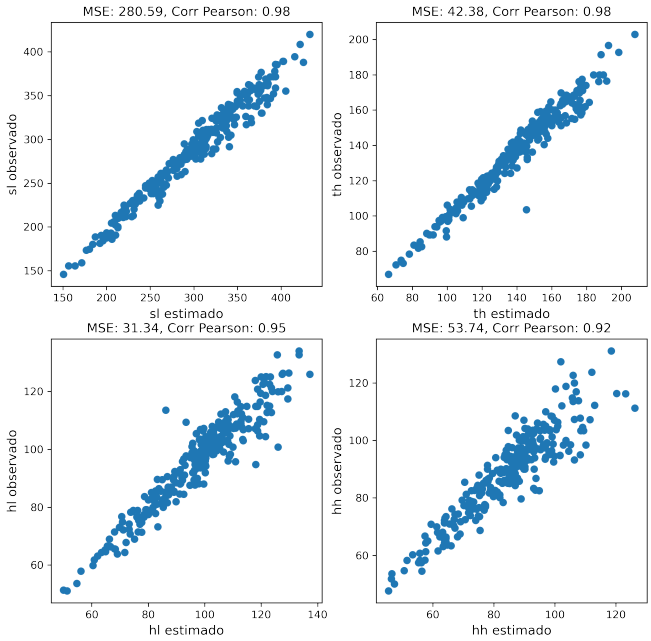
Conforme o experimento anterior, as demais medidas estão padronizadas conforme a Figura 1. A Tabela 10 apresenta as métricas obtidas a partir da comparação da medição automática com a observada por anotadores humanos.

Tabela 10 – Tabela MSE e Coef. de Pearson

Medida	MSE (mm)	Coef. Pearson
th (Altura Total)	42.38	0.98
sl (Comprimento Padrão)	280.59	0.98
hh (Altura Cabeça)	53.74	0.92
hl (Comprimento Cabeça)	31.34	0.95

A Figura 38 apresenta um gráfico de dispersão onde o eixo X representa o valor estimado pelo sistema e o eixo Y apresenta o valor anotado por um humano.

Figura 38 – Gráfico de Dispersão X = estimado, Y = observado



Fonte - produzido pelo autor

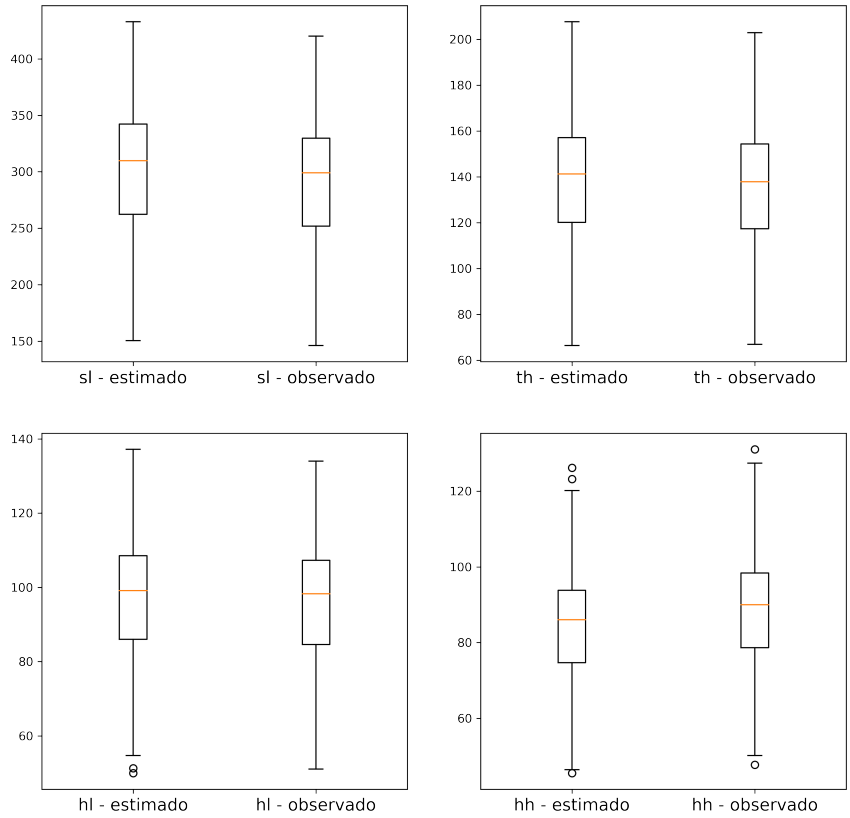
A fim de garantir a qualidade dos resultados reportados neste trabalho as imagens do conjunto utilizado na validação do modelo foram calibradas uma a uma.

Todavia, a calibração também se mostrou com baixa variabilidade nesse conjunto, em que 1 cm variou de 45,5 a 51,0 pixels (média de 48,1 pixels). O conjunto de validação das medições para o Tambaqui continha 303 animais, desses apenas 273 possuiam valores observados válidos.

Para a medição, neste conjunto de testes em particular a medida da pelve e o comprimento total foi desconsiderada pois os dados dessa métrica não foram coletados por observadores humanos. Portanto, as medidas da pelve pl *(Pelvis Length)*, ph *(Pelvis Height)* e tl *(Total Length)* não foram considerados nessa análise em particular.

Com a padronização da metodologia de coleta neste segundo experimento, não foi possível observar a existência de outliers conforme ilustra a Figura 39 comparando as medições realizadas por um humano com o estimado pelo algoritmo.

Figura 39 – Box-Whisker Plot - Observado x Estimado



Fonte - produzido pelo autor

Entretanto, é importante considerar que no experimento realizado com o *Piaractus mesopotamicus* o conjunto avaliado é quase 4 vezes maior, a escassez de outliers nesse conjunto pode ser atribuída também a este fato.

7.4 Discussão

A Visão Computacional tem se mostrado de grande potencial para automação de tarefas no contexto de Piscicultura de Precisão.

A aplicação da Mask R-CNN no contexto de fenotipagem de animais vivos, em condições naturais de iluminação, permitem uma carga menor de estresse nos animais em questão pois pode ser realizada mais próxima do habitat natural reduzindo a mortalidade durante o manejo desse tipo de operação.

Vale destacar alguns exemplos de sucesso da aplicação da arquitetura Mask R-CNN no problema de fenotipagem automática. Em (YANG et al., 2021) os autores relatam um coeficiente R^2 entre 0.96 e 0.95 para medições de cabeça e corpo na espécie Tilápia do Nilo utilizando uma abordagem combinada de arquitetura CNN customizada e cabeças de regressão no final da arquitetura.

Outro estudo com abordagem similar a apresentada neste trabalho dividindo o peixe em sub-regiões é (FENG et al., 2023) e tem como espécie alvo a Tilápia do Nilo. Esta trabalho reporta um IoU de 99.09%, 88.83%, 77.97%, 89.66%, e 90.06% para os segmentos cabeça, nadadeiras, corpo, e cauda respectivamente. Resultados em linha com os reportados na tabela 8.

Já (YU et al., 2020) reporta um erro médio relativo abaixo de 2,8% em todas as medidas obtidas pelo sistema por ele proposto para a espécie *Trachinotus ovatus*. Por fim, (YU et al., 2023) relata valores similares aos resultados obtidos neste estudo com IoU 0.50 de 0.93 para a espécie *Trachinotus ovatus*, também em linha com a tabela 8.

Apesar da popularidade da Mask R-CNN em problemas de fenotipagem automática não foram encontrados trabalhos direcionados a *Piaractus mesopotamicus* e *Colossoma macropomum*. Portanto, este trabalho representa um dos primeiros passos nessa direção voltadas a essas espécies extremamente relevantes no contexto alimentar brasileiro. Além do presente texto outras contribuições podem ser encontradas em (FREITAS et al., 2023), (ARIEDE et al., 2023a), (BATISTA; BREGA, 2024).

O sistema proposto, além de reduzir o tempo necessário, o algoritmo ainda é capaz de padronizar a operação evitando a variabilidade de metodologias de medição e permitindo a fenotipagem em larga escala para programas de melhoramento genético na Piscicultura, no aspecto comercial e de pesquisa.

Além disso, o sistema proposto pode ser utilizado para a medição de mais de uma espécie simultaneamente conforme demonstrado pelos resultados apresentados.

Para trabalhos futuros, pretende-se avaliar a possibilidade de utilizar as métricas de área em pixels obtidas para estimativa de outros valores de interesse através de regressões. Alguns valores podem ser obtidos apenas sacrificando os animais, portanto, é interessante

avaliar a qualidade das estimativas obtidas evitando sacrificar potenciais reprodutores que podem gerar indíviduos cada vez mais refinados geneticamente e com alto valor comercial agregado.

8 Conclusão

Os produtos da Piscicultura e Maricultura configuram uma importante fonte de proteína fazendo parte do cotidiano alimentar de vários países. O volume de produção mundial de pescados cresce anualmente, através dos avanços tecnológicos é possível otimizar a produção para tentar amenizar as questões de insegurança alimentar que o mundo vem passando nos últimos dois anos devido a Pandemia de Covid-19 bem como o cenário geopolítico na Europa (FAO, 2022).

Com isso, governos do mundo todo devem repensar em como alocar recursos de maneira a otimizar a produção de alimento bem como reduzir custos do processo produtivo. Avanços tecnológicos como o IoT e a Inteligência Artificial apresentam uma oportunidade de otimização desses processos produtivos.

A seleção genética é uma atividade de extrema importância para otimização de processos produtivos na agricultura. Na Piscicultura não é diferente, o uso de marcadores genéticos vem evoluindo bastante nos últimos anos, o que possibilita a seleção de indivíduos com características desejáveis a fim de aumentar a produção e reduzir custos. No entanto, para realizar a seleção genética é preciso realizar a fenotipagem dos indivíduos e selecionar aqueles que possuem as melhores características físicas como rendimento e tamanho.

A fenotipagem geralmente é realizada de maneira manual, o que impossibilita a realização desse tipo de processo em escala. Com o avanço das técnicas de processamento de imagem e Aprendizagem de Máquina é possível aplicar tais técnicas para resolução desse problema de maneira que não seja necessária a intervenção manual. Isso traz muitos benefícios ao processo de seleção reprodutiva haja visto que é possível trabalhar com populações de animais maiores.

Além disso, esse trabalho mapeou e avaliou esses conceitos desenvolvendo uma aplicação dos métodos e técnicas para obter resultados que possam apoiar o processo de seleção reprodutiva. Através dos resultados obtidos, foi possível criar um sistema capaz de reduzir custos e otimizar a produção nessa área de suma importância para reduzir a insegurança alimentar mundial nos próximos anos.

Referências

AGOSSOU, B. E.; TOSHIRO, T. Iot & ai based system for fish farming: Case study of benin. In: *Proceedings of the Conference on Information Technology for Social Good*. New York, NY, USA: Association for Computing Machinery, 2021. (GoodIT '21), p. 259–264. ISBN 9781450384780. Citado 3 vezes nas páginas 2, 3 e 11.

ANDRIALOVANIRINA, N. et al. A powerful method for measuring fish size of small-scale fishery catches using imagej. *Fisheries Research*, v. 223, p. 105425, 2020. ISSN 0165-7836. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0165783619302802>. Citado 7 vezes nas páginas 12, 14, 64, 67, 69, 71 e 72.

ARIEDE, R. B. et al. Computer vision system using deep learning to predict rib and loin yield in the fish colossoma macropomum. *Animal Genetics*, v. 54, n. 3, p. 375–388, 2023. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/age.13302>. Citado 7 vezes nas páginas 16, 65, 69, 71, 73, 74 e 86.

ARIEDE, R. B. et al. Computer vision system using deep learning to predict rib and loin yield in the fish colossoma macropomum. *Animal Genetics*, v. 54, n. 3, p. 375–388, 2023. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/age.13302>. Citado na página 67.

BATISTA, F. M.; BREGA, J. R. F. Image segmentation applied to multi-species phenotyping in fish farming. In: GERVASI, O. et al. (Ed.). *Computational Science and Its Applications – ICCSA 2024*. Cham: Springer Nature Switzerland, 2024. p. 96–111. ISBN 978-3-031-64605-8. Citado 7 vezes nas páginas 65, 67, 69, 71, 73, 74 e 86.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: LEONARDIS, A.; BISCHOF, H.; PINZ, A. (Ed.). *Computer Vision – ECCV 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 404–417. ISBN 978-3-540-33833-8. Citado na página 37.

BENGIO, Y.; LECUN, Y. Convolutional networks for images, speech, and time-series. 11 1997. Citado na página 33.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, v. 5, n. 2, p. 157–166, mar. 1994. Citado 2 vezes nas páginas 42 e 46.

BLAY, C. et al. Genetic parameters and genome-wide association studies of quality traits characterised using imaging technologies in rainbow trout, oncorhynchus mykiss. *Frontiers in Genetics*, v. 12, 2021. ISSN 1664-8021. Citado 2 vezes nas páginas 12 e 14.

BONGIOVANNI, R.; LOWENBERG-DEBOER, J. Precision agriculture and sustainability. *Precision Agriculture*, v. 5, p. 359–387, 08 2004. Citado na página 11.

CANNY, J. A computational approach to edge detection. In: FISCHLER, M. A.; FIRSCHEIN, O. (Ed.). *Readings in Computer Vision*. San Francisco (CA): Morgan Kaufmann, 1987. p. 184–203. ISBN 978-0-08-051581-6. Citado na página 21.

CAO, Y. et al. A computer vision program that identifies and classifies fish species. In: *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*. [S.l.: s.n.], 2021. p. 384–390. Citado na página 11.

COELHO-CARO, P. A. et al. Mussel classifier system based on morphological characteristics. *IEEE Access*, v. 6, p. 76935–76941, 2018. Citado 5 vezes nas páginas 64, 66, 69, 71 e 72.

CORDTS, M. et al. The cityscapes dataset for semantic urban scene understanding. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 3213–3223. Citado 2 vezes nas páginas 4 e 16.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado 3 vezes nas páginas 28, 35 e 51.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, MIT Press, n. 4, p. 303–314, 1989. Citado na página 23.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.: s.n.], 2005. v. 1, p. 886–893 vol. 1. Citado na página 51.

DANKER, A. J.; ROSENFELD, A. Blob detection by relaxation. *IEEE Transactions on Pattern Analysis Machine Intelligence*, IEEE Computer Society, Los Alamitos, CA, USA, v. 3, n. 01, p. 79–92, jan 1981. ISSN 1939-3539. Citado na página 21.

DARAPANENI, N. et al. Ai based farm fish disease detection system to help micro and small fish farmers. In: *2022 Interdisciplinary Research in Technology and Management (IRTM)*. [S.l.: s.n.], 2022. p. 1–5. Citado na página 11.

DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009. Citado na página 39.

DU, C.-J.; SUN, D.-W. Comparison of three methods for classification of pizza topping using different colour space transformations. *Journal of Food Engineering*, v. 68, n. 3, p. 277–287, 2005. ISSN 0260-8774. Citado na página 18.

F., Z. A. B. T. A. D. H. E. C. M. F. S. C. P. Revisão sistemática da literatura: conceituação, produção e publicação. *Logeion: Filosofia da Informação*, 2019. Citado na página 57.

FAO. *The State of World Fisheries and Aquaculture 2020. Sustainability in action Rome*. [S.l.]: FAO, 2020. Citado 3 vezes nas páginas 2, 3 e 11.

FAO. *State of Food Insecurity*. 2022. <https://www.fao.org/publications/sofi/en/>. Accessed: 2022-07-11. Citado na página 88.

FENG, G. et al. Accurate segmentation of tilapia fish body parts based on deeplabv3+ for advancing phenotyping applications. *Applied Sciences*, v. 13, n. 17, 2023. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/13/17/9635>. Citado 6 vezes nas páginas 64, 67, 69, 71, 73 e 86.

FERNANDES, A. F. et al. Deep learning image segmentation for extraction of fish body measurements and prediction of body weight and carcass traits in nile tilapia. *Computers and Electronics in Agriculture*, v. 170, p. 105274, 2020. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169919311561>. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

FREITAS, M. V. et al. High-throughput phenotyping by deep learning to include body shape in the breeding program of pacu (piaractus mesopotamicus). *Aquaculture*, v. 562, p. 738847, 2023. ISSN 0044-8486. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0044848622009644>. Citado 8 vezes nas páginas 16, 64, 67, 69, 71, 73, 74 e 86.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, v. 36, p. 193–202, 1980. Citado 2 vezes nas páginas 28 e 29.

GALVãO, M. C. B.; RICARTE, I. L. M. Start uma ferramenta computacional de apoio à revisão sistemática. *Brazilian Conference on Software: Theory and Practice - Tools session.*, 2010. Citado na página 60.

GIRSHICK, R. Fast r-cnn. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1440–1448. Citado 3 vezes nas páginas 33, 52 e 53.

GIRSHICK, R. B. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. Disponível em: <http://arxiv.org/abs/1311.2524>. Citado na página 51.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Citado na página 23.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 31.

GUNDERSEN, O. E.; KJENSMO, S. State of the art: Reproducibility in artificial intelligence. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. [S.l.]: AAAI Press, 2018. (AAAI'18/IAAI'18/EAAI'18). ISBN 978-1-57735-800-8. Citado na página 68.

HAO, Y.; YIN, H.; LI, D. A novel method of fish tail fin removal for mass estimation using computer vision. *Computers and Electronics in Agriculture*, v. 193, p. 106601, 2022. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169921006189>. Citado 5 vezes nas páginas 64, 67, 69, 71 e 72.

HAYKIN, S. S. *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education, 2009. Citado na página 25.

HE, K. et al. *Mask R-CNN*. 2017. Cite arxiv:1703.06870Comment: open source; appendix on more results. Disponível em: <http://arxiv.org/abs/1703.06870>. Citado 3 vezes nas páginas 12, 16 e 55.

HE, K. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014. Disponível em: <http://arxiv.org/abs/1406.4729>. Citado na página 52.

HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>. Citado 2 vezes nas páginas 46 e 48.

HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. Disponível em: <http://arxiv.org/abs/1502.01852>. Citado na página 39.

HE, K. et al. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. Disponível em: <http://arxiv.org/abs/1603.05027>. Citado na página 46.

HILDRETH;, D. M. E.; BRENNER, S. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, v. 207, p. 187–217, 1980. Citado 2 vezes nas páginas 20 e 21.

HOCHREITER, S. *Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München*. 1991. Citado na página 42.

HUBEL, D.; WIESEL, T. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, v. 160, p. 106–154, 1962. Citado na página 29.

IBGE. 2020. <https://cidades.ibge.gov.br/brasil/pesquisa/18/0>. Citado na página 11.

ISLAMADINA, R. et al. Estimating fish weight based on visual captured. In: *2018 International Conference on Information and Communications Technology (ICOIACT)*. [S.l.: s.n.], 2018. p. 366–372. Citado 5 vezes nas páginas 64, 67, 69, 71 e 72.

IVIE, P.; THAIN, D. Reproducibility in scientific computing. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 51, n. 3, jul 2018. ISSN 0360-0300. Citado na página 68.

JHA, S.; CHOPRA, S.; KINGSLY, A. Modeling of color values for nondestructive evaluation of maturity of mango. *Journal of Food Engineering*, v. 78, n. 1, p. 22–26, 2007. ISSN 0260-8774. Citado na página 18.

JONGJARAUNSUK, R.; TAPARHUDEE, W. Weight estimation of asian sea bass (lates calcarifer) comparing whole body with and without fins using computer vision technique. *Walailak Journal of Science and Technology (WJST)*, v. 18, n. 10, p. Article 9495 (11 pages), May 2021. Disponível em: <https://wjst.wu.ac.th/index.php/wjst/article/view/9495>. Citado 5 vezes nas páginas 64, 67, 69, 71 e 72.

KANDEL, I.; CASTELLI, M. Transfer learning with convolutional neural networks for diabetic retinopathy image classification. a review. *Applied Sciences*, v. 10, p. 2021, 03 2020. Citado na página 40.

KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, v. 1, n. 4, p. 321–331, 1988. Citado na página 22.

KONOVALOV, D. A. et al. Ruler detection for automatic scaling of fish images. In: *Proceedings of the International Conference on Advances in Image Processing*. New York, NY, USA: Association for Computing Machinery, 2017. (ICAIP '17), p. 90–95. ISBN 9781450352956. Disponível em: <https://doi.org/10.1145/3133264.3133271>. Citado 5 vezes nas páginas 64, 67, 69, 71 e 72.

KONOVALOV, D. A. et al. Automatic scaling of fish images. In: *Proceedings of the 2nd International Conference on Advances in Image Processing*. New York, NY, USA: Association for Computing Machinery, 2018. (ICAIP '18), p. 48–53. ISBN 9781450364607. Disponível em: <https://doi.org/10.1145/3239576.3239595>. Citado 5 vezes nas páginas 64, 66, 68, 71 e 72.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. Red Hook, NY, USA: Curran Associates Inc., 2012. (NIPS'12), p. 1097–1105. Citado na página 38.

LABELBOX. *Labelbox*. 2024. <https://labelbox.com>. [Online; accessed 07-May-2024]. Citado na página 76.

LAPICO, A. et al. Using image processing to automatically measure pearl oyster size for selective breeding. In: *2019 Digital Image Computing: Techniques and Applications (DICTA)*. [S.l.: s.n.], 2019. p. 1–8. Citado 6 vezes nas páginas 64, 66, 68, 71, 72 e 74.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, v. 1, n. 4, p. 541–551, 1989. Citado 3 vezes nas páginas 28, 30 e 35.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. Citado 2 vezes nas páginas 35 e 37.

LEEDS, T. D. et al. Response to five generations of selection for growth performance traits in rainbow trout (oncorhynchus mykiss). *Aquaculture*, v. 465, p. 341–351, 2016. ISSN 0044-8486. Citado 2 vezes nas páginas 12 e 14.

LEKUNBERRI, X. et al. Identification and measurement of tropical tuna species in purse seiner catches using computer vision and deep learning. *Ecological Informatics*, v. 67, p. 101495, 2022. ISSN 1574-9541. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574954121002867>. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

LHORENTE, J. P. et al. Advances in genetic improvement for salmon and trout aquaculture: the chilean situation and prospects. *Reviews in Aquaculture*, v. 11, n. 2, p. 340–353, 2019. Citado 2 vezes nas páginas 12 e 15.

LIN, M.; CHEN, Q.; YAN, S. *Network In Network*. 2013. Cite arxiv:1312.4400Comment: 10 pages, 4 figures, for iclr2014. Disponível em: <http://arxiv.org/abs/1312.4400>. Citado 2 vezes nas páginas 43 e 44.

LIU, W. et al. Real-time multilead convolutional neural network for myocardial infarction detection. *IEEE Journal of Biomedical and Health Informatics*, v. 22, n. 5, p. 1434–1444, 2018. Citado na página 33.

LOWE, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, v. 60, p. 91–, 11 2004. Citado 2 vezes nas páginas 37 e 51.

MALIK, S.; KUMAR, T.; SAHOO, A. K. Image processing techniques for identification of fish disease. In: *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*. [S.l.: s.n.], 2017. p. 55–59. Citado na página 11.

MARCO, V. S. et al. Low-cost automatic fish measuring estimation. In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. New York, NY, USA: Association for Computing Machinery, 2021. (SAC '21), p. 90–97. ISBN 9781450381048. Disponível em: <https://doi.org/10.1145/3412841.3441889>. Citado 5 vezes nas páginas 64, 67, 69, 71 e 73.

MENDOZA, F.; DEJMEK, P.; AGUILERA, J. M. Calibrated color measurements of agricultural foods using image analysis. *Postharvest Biology and Technology*, v. 41, n. 3, p. 285 – 295, 2006. Cited by: 260. Citado na página 18.

MUSA, Z.; WATADA, J. Dynamic tracking system using foot step direction. *BSCHS*, v. 13, p. 51–57, 01 2008. Citado na página 30.

NGUYEN, K.-T. et al. Two-phase instance segmentation for whiteleg shrimp larvae counting. In: *2020 IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.: s.n.], 2020. p. 1–3. Citado na página 11.

PALA, T. et al. Comparison of pooling methods for handwritten digit recognition problem. In: *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. [S.l.: s.n.], 2018. p. 1–5. Citado na página 33.

PETRELLIS, N. Fish morphological feature recognition based on deep learning techniques. In: *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. [S.l.: s.n.], 2021. p. 1–4. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

PETRELLIS, N. Measurement of fish morphological features through image processing and deep learning techniques. *Applied Sciences*, v. 11, p. 4416, 05 2021. Citado 5 vezes nas páginas 64, 67, 69, 71 e 72.

PIZER, S. M. et al. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, v. 39, n. 3, p. 355–368, 1987. ISSN 0734-189X. Citado na página 19.

PREWITT, J. Object enhancement and extraction. *Picture Processing and Psychophysics*, Elsevier Science, Amsterdam, p. 75–149, 1970. Citado na página 21.

QASHLIM, A. et al. Estimation of milkfish physical weighting as fishery industry support system using image processing technology. *Journal of Physics: Conference Series*, IOP Publishing, v. 1175, n. 1, p. 012029, mar 2019. Disponível em: <https://dx.doi.org/10.1088/1742-6596/1175/1/012029>. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. Disponível em: <http://arxiv.org/abs/1506.01497>. Citado 2 vezes nas páginas 53 e 54.

RIDHWAN, A. et al. Overview of machine vision on digital imaging approach for automatic tuna length measurement. *IOP Conference Series: Materials Science and Engineering*, v. 551, p. 012076, 08 2019. Citado 5 vezes nas páginas 64, 67, 69, 71 e 72.

ROBERTS, L. Machine perception of three-dimensional solids. *Optical and Electro-optical Information Processing, ed. by J.T. Tippet*, MIT Press, Cambridge, p. 159–197, 1965. Citado na página 21.

ROBERTSON, A. R. The cie 1976 color-difference formulae. *Color Research & Application*, v. 2, n. 1, p. 7–11, 1977. Citado na página 18.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. Disponível em: <http://arxiv.org/abs/1505.04597>. Citado na página 74.

ROSE, D. C.; CHILVERS, J. Agriculture 4.0: Broadening responsible innovation in an era of smart farming. *Frontiers in Sustainable Food Systems*, v. 2, 2018. Cited by: 166; All Open Access, Gold Open Access, Green Open Access. Citado na página 11.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In: RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. (Ed.). Cambridge, MA, USA: MIT Press, 1986. cap. Learning Internal Representations by Error Propagation, p. 318–362. ISBN 0-262-68053-X. Citado 3 vezes nas páginas 24, 25 e 30.

RUSSAKOVSKY, O. et al. Detecting avocados to zucchinis: What have we done, and where are we going? In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 2064–2071. Citado na página 40.

SALEH, A. et al. *Prawn Morphometrics and Weight Estimation from Images using Deep Learning for Landmark Localization*. 2023. Disponível em: <https://arxiv.org/abs/2307.07732>. Citado 5 vezes nas páginas 65, 67, 69, 71 e 73.

SEVEN, G. et al. Use of artificial intelligence in the prediction of malignant potential of gastric gastrointestinal stromal tumors. *Digestive Diseases and Sciences*, v. 67, p. 1–9, 01 2022. Citado na página 34.

SHIBATA, Y. et al. Length estimation of fish detected as non-occluded using a smartphone application and deep learning method. *Fisheries Research*, v. 273, p. 106970, 2024. ISSN 0165-7836. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0165783624000341>. Citado 5 vezes nas páginas 65, 67, 69, 71 e 73.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: . [S.l.: s.n.], 2014. Citado 2 vezes nas páginas 41 e 44.

SOBEL, I.; FELDMAN, G. A 3×3 isotropic gradient operator for image processing. *Pattern Classification and Scene Analysis*, p. 271–272, 01 1973. Citado na página 21.

SRIVASTAVA, R. K.; GREFF, K.; SCHMIDHUBER, J. Highway networks. *CoRR*, abs/1505.00387, 2015. Disponível em: <http://arxiv.org/abs/1505.00387>. Citado na página 46.

SZEGEDY, C. et al. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. Disponível em: <http://arxiv.org/abs/1409.4842>. Citado 2 vezes nas páginas 44 e 46.

SZELISKI, R. *Computer Vision - Algorithms and Applications, Second Edition*. [S.l.]: Springer, 2022. (Texts in Computer Science). Citado 3 vezes nas páginas 18, 19 e 20.

TAHERI-GARAVAND, A. et al. Meat quality evaluation based on computer vision technique: A review. *Meat Science*, v. 156, p. 183–195, 2019. ISSN 0309-1740. Citado na página 11.

TANHIR, M.; IQBAL, M. Implementation of yolov8-pose model for identification and estimation of the length of skipjack and tuna like species on the website. *BIO Web of Conferences*, v. 106, p. 01005, 05 2024. Citado 5 vezes nas páginas 64, 67, 69, 71 e 73.

TSENG, C.-H.; HSIEH, C.-L.; KUO, Y.-F. Automatic measurement of the body length of harvested fish using convolutional neural networks. *Biosystems Engineering*, v. 189, p. 36–47, 2020. ISSN 1537-5110. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511019308761>. Citado 5 vezes nas páginas 63, 67, 69, 71 e 72.

UIJLINGS, J. R. R. et al. Selective search for object recognition. *International Journal of Computer Vision*, v. 104, n. 2, p. 154–171, 2013. Disponível em: <https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013>. Citado 2 vezes nas páginas 16 e 51.

WALEED, A. et al. Automatic recognition of fish diseases in fish farms. In: *2019 14th International Conference on Computer Engineering and Systems (ICCES)*. [S.l.: s.n.], 2019. p. 201–206. Citado na página 11.

WANG, G. et al. Multi-scale fish segmentation refinement and missing shape recovery. *IEEE Access*, v. 7, p. 52836–52845, 2019. Citado 3 vezes nas páginas 11, 22 e 74.

WANG, G. et al. Multi-scale fish segmentation refinement and missing shape recovery. *IEEE Access*, v. 7, p. 52836–52845, 2019. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

WANG, G. et al. Coarse-to-fine segmentation refinement and missing shape recovery for halibut fish. In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. [S.l.: s.n.], 2018. p. 370–374. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

WANG, G.; Van Stappen, G.; De Baets, B. Automated artemia length measurement using u-shaped fully convolutional networks and second-order anisotropic gaussian kernels. *Computers and Electronics in Agriculture*, v. 168, p. 105102, 2020. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016816991931587X>. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

WERBOS, P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Tese (Doutorado) — Harvard University, 1974. Citado na página 24.

XUE, Y. et al. An analytical framework to predict slaughter traits from images in fish. *Aquaculture*, v. 566, p. 739175, 2023. ISSN 0044-8486. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0044848622012935>. Citado 6 vezes nas páginas 64, 67, 69, 71, 73 e 74.

YANG, Y. et al. Deep convolutional neural networks for fish weight prediction from images. In: *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)*. [S.l.: s.n.], 2021. p. 1–6. Citado 7 vezes nas páginas 64, 67, 69, 71, 72, 74 e 86.

YASRUDDIN, M. L. et al. Feasibility study of fish disease detection using computer vision and deep convolutional neural network (dcnn) algorithm. In: *2022 IEEE 18th International Colloquium on Signal Processing Applications (CSPA)*. [S.l.: s.n.], 2022. p. 272–276. Citado na página 11.

YU, C. et al. Segmentation and measurement scheme for fish morphological features based on mask r-cnn. *Information Processing in Agriculture*, v. 7, n. 4, p. 523–534, 2020. ISSN 2214-3173. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214317319301714>. Citado 6 vezes nas páginas 64, 69, 71, 72, 74 e 86.

YU, C. et al. Intelligent measurement of morphological characteristics of fish using improved u-net. *Electronics*, v. 10, n. 12, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/12/1426>. Citado 6 vezes nas páginas 64, 67, 69, 71, 72 e 74.

YU, C. et al. An intelligent measurement scheme for basic characters of fish in smart aquaculture. *Computers and Electronics in Agriculture*, v. 204, p. 107506, 2023. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169922008146>. Citado 6 vezes nas páginas 64, 69, 71, 73, 74 e 86.

YU, Y.; ZHANG, H.; YUAN, F. Key point detection method for fish size measurement based on deep learning. *IET Image Processing*, v. 17, n. 14, p. 4142–4158, 2023. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.12924>. Citado 5 vezes nas páginas 64, 67, 69, 71 e 73.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. Disponível em: <http://arxiv.org/abs/1311.2901>. Citado na página 40.

ZENG, J. et al. Deep learning to obtain high-throughput morphological phenotypes and its genetic correlation with swimming performance in juvenile large yellow croaker. *Aquaculture*, v. 578, p. 740051, 2024. ISSN 0044-8486. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0044848623008256>. Citado 5 vezes nas páginas 65, 67, 69, 71 e 73.