

# Inferring Broiler Chicken Weight through Machine Learning

Lucas H. N. de Sousa<sup>1</sup>, Mauri Ferrandin<sup>1</sup>, Carlos Moratelli<sup>1</sup>

<sup>1</sup>Departamento de Engenharia de Controle, Automação e Computação  
Universidade Federal de Santa Catarina (UFSC)  
Rua João Pessoa, 2514 – Blumenau – SC – Brazil

lucasheilbuth@yahoo.com.br, {mauri.ferrandin, carlos.moratelli}@ufsc.br

**Abstract.** *This study presents a methodology for weighing broiler chickens using image analysis. Our method involves capturing images at a fixed height above the ground, with the processing occurring in three stages. First, a neural network is employed for classification, capable of determining the outline of the birds by applying ellipses, resulting in binary images. Second, a method is applied to extract geometric features from the generated binary images. Finally, another neural network is used to infer weight based on the geometric features. As a result, our technique allows for the inference of the bird's weight at their growing site. The average weight prediction error was 5.34%.*

**Resumo.** *Este trabalho apresenta uma metodologia para pesagem de aves de corte através de análise de imagens. Nosso método consiste na obtenção de imagens a uma altura fixa do solo, com o processamento ocorrendo em três etapas. Primeiro, utiliza-se uma rede neural para classificação, capaz de determinar o contorno das aves através da aplicação elipses, resultado em imagens binárias. Segundo, aplica-se um método para extração de características geométricas a partir das imagens binárias geradas. Por último, utiliza-se uma rede neural para inferência de peso a partir das características geométricas. Como resultado, nossa técnica permite a inferência do peso das aves in seu local de crescimento. O erro médio de predição de peso foi de 5.34%.*

## 1. Introdução

O agronegócio representou cerca de 25% do PIB brasileiro em 2022, conforme dados do CEPEA [Cepea 2022]. Ainda, o setor destaca-se como um dos maiores exportadores de alimentos do mundo. Mas nem sempre foi assim, a pouco mais de 50 anos, a agricultura e pecuária brasileiras eram rudimentares e pouco mecanizadas. Foi necessário muito estudo e tecnologia para elevarmos a produção aos patamares atuais. Apesar deste ritmo de crescimento, o agronegócio brasileiro ainda carece de tecnologia. Disparidade cambial e custos de importação inviabilizam a adoção de muitas tecnologias estrangeiras, principalmente na agricultura familiar [Souza et al. 2019]. Desta forma, o desenvolvimento tecnológico nacional, com custos compatíveis com nossa realidade, são de grande importância para os diversos setores da economia nacional, entre eles o próprio agronegócio.

Uma das áreas com a mais rápida evolução tecnológica nos últimos anos tem sido a Inteligência Artificial (IA). Somos surpreendidos por aplicações que não seriam possíveis a 15 ou 20 anos atrás, como carros autônomos, tradutores de texto, reconhecimento e geração de imagens, entre outros. Redes Neurais [Charu 2018] estão por

trás da maioria destes avanços. Trata-se de modelos inspirados na estrutura e funcionamento do cérebro humano, no qual, camadas de "neurônios" interconectados processam e transmitem informações. O aprendizado de máquina (ML, do inglês, Machine Learning) é uma área da IA focada em construir sistemas inteligentes capazes de aprender a partir de bases de dados, sem a necessidade de serem explicitamente programados [Müller and Guido 2016]. Usando ML podemos treinar redes neurais para realizar tarefas como reconhecimento de objetos em imagens e realizar inferências, desde que, um conjunto de informações esteja previamente disponível para treinamento. ML já tem sido aplicada em diversas áreas do agronegócio como: seleção de sementes, detecção de doenças e pestes, entre outros [Meshram et al. 2021].

Seguindo os demais setores do agronegócio, a avicultura cresceu em produção, exportação e consumo. Neste ramo, as aves são confinadas em estruturas, denominadas aviários, com capacidade para dezenas de milhares de animais. Uma tarefa cotidiana neste ambiente é determinar a média de engorda dos animais [AVILA et al. 2007]. Rotineiramente, os encarregados precisam selecionar aves, ao acaso, e pesá-las para aferir o peso médio. É necessário a pesagem de dezenas ou até centenas de aves para haver uma representação estatística válida do grupo, isso é uma tarefa que demanda tempo. Assim, as pesagens são realizadas em intervalos de tempo, por exemplo, uma vez por semana. Visando resolver este problema e agregar maior qualidade nos dados obtidos, via monitoração constante e uma maior amostragem de aves, apresentaremos um método de ML capaz de realizar inferência dos pesos das aves a partir de imagens. A nossa técnica consiste na gravação de vídeos, a uma altura fixa do solo, que são posteriormente processados e utilizados no treinamento de modelos de Rede Neural Convolutacional (CNN, do inglês, Convolutional Neural Network) pelo método de ML.

A metodologia deste trabalho consiste na extração de imagens dos vídeos obtidos e na posterior submissão destas imagens a um processo dividido em três etapas: 1) utilizar uma rede neural com arquitetura do tipo Mask RCNN [He et al. 2017] para detecção das aves. Neste etapa, o corpo das aves é delineado por uma elipse que exclui cabeça e cauda, obtendo-se uma imagem binária ao final do processo; 2) as imagens binárias são processadas para extração de certas características geométricas das elipses, como diâmetros e área. Observe que diversas aves podem ter sido detectadas na mesma imagem; 3) as características geométricas são submetidas a uma segunda rede neural do tipo *feedforward* [Fine et al. 1999] capaz de realizar a predição do peso. Como resultado, é possível realizar um acompanhamento contínuo do crescimento das aves, sem a necessidade de intervenção humana.

O objetivo deste trabalho é descrever a técnica utilizada desde (i) a captação dos vídeos, (ii) concepção dos modelos de rede neural e bibliotecas utilizadas, (iii) preparo da base de dados para treinamento e validação, (iv) detecção das aves e posterior inferência de peso e os (iv) resultados obtidos. Este trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalho relacionados, a Seção 3 apresenta a configuração física e de software utilizada, as Seções 4 e 6 demonstram os métodos utilizados para o treinamento das redes neurais e os principais resultados, a Seção 7 traz uma discussão geral sobre os resultados obtidos e possíveis pontos de melhorias e a Seção 8 apresenta a conclusão deste trabalho.

## 2. Trabalhos Relacionados

Neste trabalho, usamos como inspiração, a técnica de inferência de peso apresentada por [Somaye Amraei 2018]. Amraei, apresenta um método para segmentação de objetos em imagens, capaz de delinear as aves e, em seguida, aplica uma técnica para retirar a cabeça e a cauda da ave para obtenção de uma aproximação elíptica do objeto. Em seguida o autor extraiu as seguintes características geométricas do objeto elíptico: área, área convexa, perímetro, excentricidade, maior e menor raio. Para realizar a predição do peso do frango, criou-se um modelo de função de transferência que considera tais características geométricas como entrada, resultando no peso como saída. Em conclusão, os autores afirmam que é possível obter o peso da ave utilizando sua técnica com um erro médio aceitável.

Dohomen et al. [Dohmen et al. 2021] utilizou técnicas de ML para realizar a predição de peso de novilhas usando imagens. Ele testou a utilização de imagens de duas vistas distintas: superior e lateral. Primeiro, uma rede neural foi treinada e utilizada para realizar segmentação das imagens. Ao contrário de Amraei, Dohomen não extrai informações geométricas, mas utiliza o contorno exato do animal. Com um conjunto de imagens segmentadas e seus respectivos pesos, uma rede neural foi treinada para inferência do peso. Em suma, os autores afirmam que as imagens superiores têm maior precisão, ao evitar problemas na captura das patas e cabeça, e que sua técnica possui um erro médio aceitável.

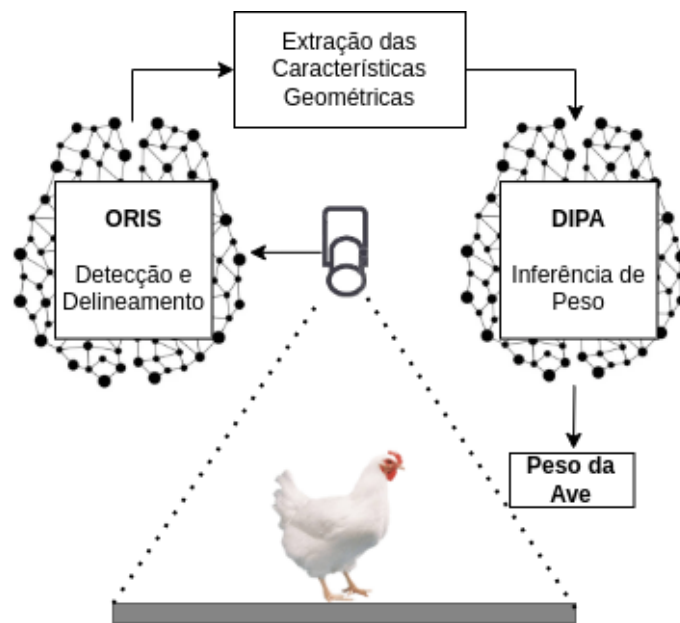
Tolentino et al. [Tolentino et al. 2020] realizou inferência de peso para tilápias do Nilo em ambiente aquático. Ao contrário de Amraei e Dohomen, cujos animais estão a uma distância fixa do dispositivo de gravação, os peixes podem estar a diferentes distâncias, conforme se deslocam no aquário, afastando-se ou aproximando-se da câmera. Desta forma, Tolentino precisou utilizar duas câmeras para criar um sistema de visão estéreo, assim, permitindo determinar a distância dos peixes. O peso é determinado pela correlação entre comprimento e largura do peixe. Estas informações são obtidas por técnicas clássicas de segmentação de imagens.

A metodologia utilizada neste trabalho aproveita a ideia de obtenção de características geométricas dos animais a partir de elipses proposta por Amraei, enquanto utiliza redes neurais para análise das imagens obtidas a partir da vista superior, para posterior inferência de peso, semelhante a Dohomen. Ainda, a câmera fica posicionada a uma distância fixa do solo, evitando a necessidade de visão estéreo.

## 3. Base de Dados e Ferramentas Utilizadas

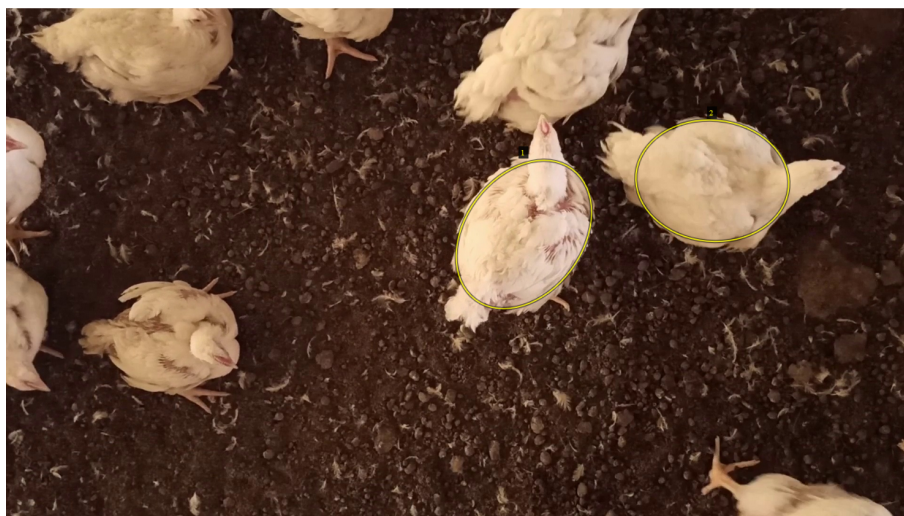
A Figura 1 ilustra o esquema proposto neste trabalho. Uma câmera suspensa acima das aves captura imagens que são processadas por duas redes neurais distintas. A primeira, chamada de *Object Recognition Instance Segmentation (ORIS)*, é utilizada para detecção e enquadramento das aves em elipses, conforme explicado na Seção 2. A partir disso, são geradas imagens binárias, das quais, extrai-se certas características geométricas das elipses. A segunda, chamada de *Deep Learning para Inferência de Peso de Ave (DIPA)*, utiliza como entrada as informações geométricas das elipses e realiza a inferência do peso da ave.

Foi necessário a formação de uma base de conhecimento para treinamento de ambas as redes. Para isso, foram capturados vídeos em um aviário abrigando cerca de 20.000



**Figura 1. Esquema proposto para inferência do peso.**

aves com 25 dias de vida. Para gravação, foi utilizado um Smartphone Xiaomi Redmi Note 11 com câmera de 50MP, sendo fixado a uma altura de 1,5 metro do solo. Primeiramente, foi obtido cerca de 8 minutos de gravação, onde os animais puderam circular livremente abaixo da câmera. Um quadro do vídeo resultante pode ser visto na Figura 2. Deste vídeo, foram extraídos quadros para o treinamento da ORIS para a detecção da ave e delineamento da elipse. Em um segundo momento, 46 aves foram selecionadas aleatoriamente e pesadas manualmente. Para cada uma delas, foi gravado um vídeo com alguns segundos de duração, mantendo-se a posição original da câmera. O peso da ave e o vídeo foram associados na base de conhecimento e, posteriormente, estas informações foram utilizadas no treinamento da DIPA.



**Figura 2. Exemplo de quadro capturado e anotação manual dos objetos.**

Para o treinamento da ORIS foi necessário gerar um conjunto de exemplos anotados. Para isso, foi utilizado um software de anotações, chamado VGG Image Annotator (VGGIA) [Dutta and Zisserman 2019]. Esta ferramenta foi utilizada por permitir anotações elípticas em objetos com qualquer ângulo de rotação. Na Figura 2, é observado um exemplo da anotação elíptica de uma ave, onde duas aves possuem um contorno em amarelo que exclui sua cabeça e cauda, tratando-se da região de interesse anotada. Nota-se que a rede neural foi treinada para reconhecimento da ave e exclusão da cauda e cabeça simultaneamente. Cauda e cabeça atrapalham na inferência do peso e a remoção imediata evita necessidade de pós-processamento. Esta anotação foi um processo manual que exigiu muito esforço, pois a precisão da detecção dos objetos depende da precisão das anotações. É necessário anotar o máximo possível de objetos por imagem para aumentar a quantidade de registros na base de conhecimento. Foi tomado o cuidado para selecionar apenas aves que tinham uma distância mínima com outras aves, assim, evitando registros sobrepostos, e também, garantindo que os animais estivessem totalmente presentes na imagem. Após realização da anotação manual de 1.645 imagens foi gerado um arquivo em formato JSON (*Java Script Object Notation*) contendo as informações geométricas de todos os objetos anotados.

A implementação da ORIS foi realizada em linguagem Python com a utilização das bibliotecas *detectron2*, *json*, *numpy*, *cv2* (*openCV*), dentre outras. A biblioteca *detectron2* foi utilizada para realizar segmentação dos objetos na imagem, a biblioteca *json* para leitura de arquivos e extração de informações das anotações. Já a biblioteca *numpy* foi usada para o processamento matemático, como, por exemplo, a partir da sequência de pontos da imagem extrair os valores mínimos e máximos para formar uma caixa delimitadora (*bounding box*) e *cv2* para a visualização dos resultados coletados pela rede neural.

Para treinamento da DIPA foram extraídas as seguintes características geométricas de cada ave reconhecida e delineada através da aplicação da elipse: área, área convexa, perímetro, raio menor, raio maior, área da caixa delimitadora e peso. Na implementação foram utilizadas principalmente as bibliotecas *numpy*, *pandas*, *seaborn*, *tensorflow* e *cv2*. Sendo que a biblioteca *pandas* foi usada para armazenamento das informações, *seaborn* para visualização dos dados e reconhecimento de padrões e *tensorflow* para o treinamento do modelo. Ambas as redes foram treinadas com auxílio do Google Colab.

#### **4. Treinamento da Rede ORIS e Avaliação dos Resultados**

Nesta seção será explicado o procedimento de treinamento das rede neural ORIS. Observe que, a ORIS realiza a detecção e delineamento das aves, excluindo cauda e cabeça. A partir disso, gera-se uma imagem binária, da qual, extraí-se as características geométricas mencionadas na Seção 3. Estas características são usadas como entrada para o treinamento da DIPA que (veja Seção 6), por sua vez, realiza a inferência do peso da ave.

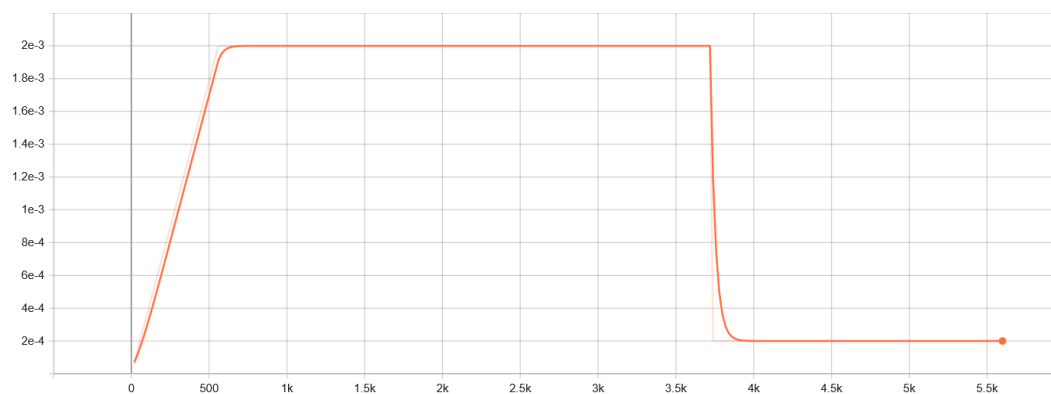
Para reconhecimento das aves e segmentação da imagem utilizou-se uma arquitetura de rede neural conhecida como Mask RCNN [He et al. 2017]. Neste modelo, alguns parâmetros de treinamento precisam ser definidos pelo projetista, são eles: LR (*Learning Rate* ou Taxa de Aprendizado), número de épocas e *gamma* (fator de decaimento de aprendizagem). Para LR, definiu-se um comportamento dinâmico para evitar alguns problemas como: *early overfitting* e não convergência do gradiente descendente. Para

solucionar o early overfitting é utilizado o método de incremento linear de LR, que consiste em aplicar uma rampa na função de LR, ou seja, inicia-se com um valor próximo a zero e incrementa-se linearmente o valor. A não convergência do gradiente descendente ocorre porque o ponto amostral pode diferir significativamente do resultado procurado, desta forma, ao aproximar-se do final do treinamento é necessário diminuir o LR. Desta forma, determina-se uma nova taxa multiplicando LR pelo fator gamma, a partir de uma determinada época do treinamento, resultando uma queda abrupta da taxa de aprendizado da rede. A Figura 3 ilustra uma curva de aprendizado típica para esta técnica. LR inicia próximo de 0 e tem aumento de 10% estabilizando-se no valor máximo de 0,002. Em 66% do treinamento, há um decréscimo de LR para a nova taxa gamma de 0,0002.

O dataset foi dividido através da técnica *hold-out* gerando-se três grupos de imagens: treino, validação e teste. Cada grupo ficou com um terço das imagens. Foram realizados alguns treinamentos para compreender a margem de valores ideais de LR, número de épocas e gamma. Em seguida define-se um conjunto de valores promissores para cada parâmetro, conforme visto abaixo:

- LR =  $[25 \cdot 10^{-3}; 125 \cdot 10^{-3}; 25 \cdot 10^{-2}]$
- épocas = [5400; 7200]
- gamma = [0,02; 0,1]

Assim, obteve-se a combinação entre todos os valores possíveis gerando 12 cenários, conforme demonstrado na Tabela 1.



**Figura 3. Curva LR durante o treinamento.**

Como métrica de validação dos resultados da rede ORIS, foi utilizado mAP (*Mean Average Precision*) [Beitzel et al. 2009], que é capaz de medir a similaridade entre a saída da rede com a saída esperada. Para realizar o cálculo desta métrica é fundamental a definição de IoU (Intersecção sobre união). IoU mede a performance da predição de um único objeto e é definida como a razão entre a área de intersecção dividida pela área de união. Se há uma sobreposição perfeita entre as duas áreas, a área de intersecção é igual à área de união e, portanto, IoU possui o valor de 1. Por outro lado, se há uma distinção completa, a área da intersecção IoU resulta em 0. Assim, esta métrica permite analisar o desempenho do modelo Mask RCNN na tarefa de segmentação. Ainda, utilizou-se as variações  $mAP_{50}$  e  $mAP_{75}$ , que é a mAP calculada em limiares IoU de 0,5 e 0,75, respectivamente. Por exemplo, para  $mAP_{50}$  uma intersecção é considerada verdadeira se

tiver uma sobreposição de 50% ou mais com a caixa verdadeira. Assim, estas são métricas mais rígidas.

**Tabela 1. Definição dos cenários de ORIS.**

Configuração	LR	Épocas	Gamma
<b>1</b>	$25 \cdot 10^{-3}$	5400	0,02
<b>2</b>	$25 \cdot 10^{-3}$	7200	0,02
<b>3</b>	$125 \cdot 10^{-3}$	5400	0,02
<b>4</b>	$125 \cdot 10^{-3}$	7200	0,02
<b>5</b>	$25 \cdot 10^{-2}$	5400	0,02
<b>6</b>	$25 \cdot 10^{-2}$	7200	0,02
<b>7</b>	$25 \cdot 10^{-3}$	5400	0,1
<b>8</b>	$25 \cdot 10^{-3}$	7200	0,1
<b>9</b>	$125 \cdot 10^{-3}$	5400	0,1
<b>10</b>	$125 \cdot 10^{-3}$	7200	0,1
<b>11</b>	$25 \cdot 10^{-2}$	5400	0,1
<b>12</b>	$25 \cdot 10^{-2}$	7200	0,1

#### 4.1. Avaliação dos resultados para ORIS

A partir configurações apresentadas, realizou-se o treinamento para cada cenário, coletando-se as métricas de desempenho. O resultado é mostrado na Tabela 2. Observa-se que o melhor resultado foi da configuração 9, pois obteve o maior mAP, considerada a principal métrica de avaliação, e valores próximo para as outras duas métricas.

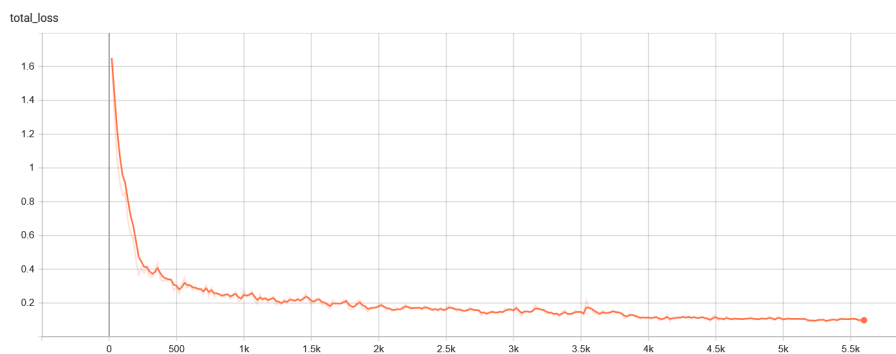
**Tabela 2. Performance da ORIS para cada cenário de treinamento.**

Configuração	mAP	mAP <sub>50</sub>	mAP <sub>75</sub>
<b>1</b>	79,032	88,62	88,603
<b>2</b>	79,414	88,406	88,406
<b>3</b>	79,165	87,317	87,317
<b>4</b>	79,639	87,794	87,794
<b>5</b>	79,527	87,179	87,179
<b>6</b>	78,997	86,231	86,231
<b>7</b>	79,037	88,088	88,088
<b>8</b>	79,465	88,043	88,043
<b>9</b>	<b>80,33</b>	<b>88,332</b>	<b>88,332</b>
<b>10</b>	79,372	87,277	87,277
<b>11</b>	79,852	87,86	87,86
<b>12</b>	78,81	86,351	86,351

A partir da configuração 9, novos ajustes foram obtidos através de testes arbitrários variando-se cada parâmetro individualmente, chegando aos valores finais utilizados na rede ORIS:

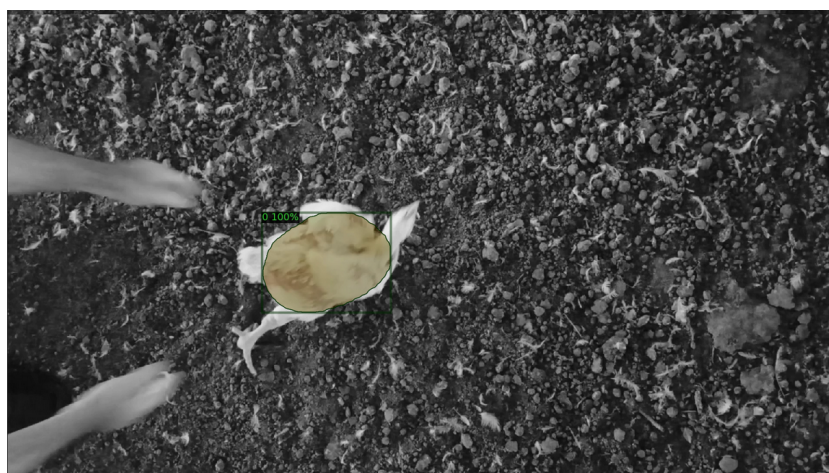
- LR =  $2 \cdot 10^{-3}$
- Época = 5600
- Gamma = 0,1

As métricas obtidas para estes parâmetros foram mAP de 80,668,  $mAP_{50}$  de 88,456 e  $mAP_{75}$  de 88,456. Estes valores correspondem ao melhor mAP entre todos os cenários e o segundo maior  $mAP_{50}$  e  $mAP_{75}$ . Na sequência, foi gerado o gráfico do erro total ao longo do treinamento, este é observado na Figura 4.



**Figura 4. Erro total.**

A Figura 5 nos dá um exemplo do resultado da segmentação, que demonstra um contorno aproximado por uma elipse do corpo da ave, excluindo a cabeça e a cauda. Neste imagem, fica claro que a rede neural foi capaz de eliminar até mesmo o pé que, dependendo da pose da ave, pode aparecer na imagem.



**Figura 5. Contorno inferido pela ORIS.**

## 5. Extração das Características Geométricas

A segmentação realizada pela rede ORIS é utilizada para a geração de imagens binárias e, conseqüente extração das características geométricas, conforme explicado na Seção 3. Iniciou-se com a pesagem de 46 aves, porém, devido a quantidade limitada de aves, utilizou-se a técnica conhecida como aumento de dados (do inglês, data augmentation) [Shorten and Khoshgoftaar 2019], onde foram retiradas 4 imagens aleatórias do vídeo de cada ave, resultando em um total de 184 imagens, cada qual, associada ao peso da respectiva ave. As 184 imagens foram submetidas a rede ORIS e, após a segmentação, gerou-se as imagens binárias das quais extraiu-se as características geométricas. O resultado foi



armazenado em uma tabela que associa as características geométricas com o peso, obtido manualmente. Então, gerou-se a matriz de correlação apresentada na Figura 6.



**Figura 6. Matriz de Correlação.**

As informações mais importantes estão na última linha da matriz que demonstra a correlação do peso com as características geométricas que serão as entradas da rede DIPA. Observa-se alta correlação entre as características geométricas e o peso, principalmente em relação a área, área convexa e perímetro. Estes resultados mostram que o treinamento de uma rede para inferência de peso baseada nestes parâmetros é promissora, conforme veremos na Seção 6.

## 6. Treinamento da Rede DIPA e Avaliação dos Resultados

A rede DIPA será usada para inferência do peso das aves a partir das características geométricas. Conforme mostrado na Seção 5, as características geométricas obtidas pela segmentação das aves tem correlação significativa com o peso. Assim, para proceder com o treinamento da rede neural para inferência de peso, aqui denominada DIPA, que utiliza uma arquitetura do tipo *feedforward*, iniciou-se pela aplicação do método de normalização *MinMaxScaler* [Raju et al. 2020] nos dados de entrada. Ainda, foram exploradas as funções de ativação *LReLU*, *PReLU*, *ELU*, *SELU*, sigmoide e tangente hiperbólica [Kaloiev and Krastev 2021]. Além destas, foram testadas a configuração de *LReLU* e *PReLU* com a última camada intermediária utilizando tangente hiperbólica. Cada função de ativação possui vantagens e desvantagens, contudo, neste trabalho objetiva-se maior acurácia. Desta forma, foi realizado o treinamento para cada função de ativação, coletando-se as métricas de desempenho.

Foi utilizado o método de *early stopping* para evitar *overfitting*, enquanto que, a métrica de monitoração do incremento do erro foi o Erro Quadrático Médio (MSE, do inglês, *Medium Square Error*). Além disso, foi utilizada a técnica de separação de dados, conhecida como Validação Cruzada [Kwok and Yeung 1995] (CV, do inglês, *Cross Validation*) com 5 folds entre dados de treino e validação. Ao final do treinamento, foi calculado a MSE dos folds.

Com o objetivo de definir a melhor função de ativação, foram realizados testes arbitrários variando-se os parâmetros da rede até obter-se um resultado satisfatório. Com os parâmetros definidos, foi realizado um treinamento para cada função de ativação. Em seguida, procurou-se pela rede com melhor performance com base na métrica MSE. As configuração dos parâmetros obtidas nesta etapa são:

- LR = 0,008
- Épocas = 900
- Número de neurônios = 512
- Profundidade = 2

Número de neurônios são a quantidade de neurônios em cada camada da rede neural, excetuando a camada de saída que apresenta apenas 1 neurônio. Profundidade é a quantidade de camadas da rede neural.

As métricas de validação utilizadas foram o Coeficiente de Determinação  $R^2$ , o Erro Absoluto Médio (MAE, do inglês, *Medium Absolute Error*) e o Erro Percentual Absoluto Médio (MAPE, do inglês, *Medium Absolute Percentage Error*) [Jierula et al. 2021]. Os resultados obtidos estão expostos na Tabela 3.

**Tabela 3. Métricas de avaliação do primeiro treinamento.**

Função De Ativação	Média $R^2$	Média MAE	Média MAPE
LReLU	0,48	0,16	0,07
PReLU	0,47	0,16	0,07
ELU	0,47	0,2	0,09
SELU	0,44	0,63	0,27
Sigmoide	0,46	0,31	0,14
Tangente Hiperbólica	0,5	0,16	0,07
<b>LReLU e Tangente Hiperbólica</b>	<b>0,54</b>	<b>0,14</b>	<b>0,06</b>
<b>PReLU e Tangente Hiperbólica</b>	0,52	0,22	0,09

Pelos resultados obtidos foi observado que a rede neural LReLU com Tangente Hiperbólica obteve a melhor performance em questões de acurácia. Em seguida foi gerada uma lista de valores dos possíveis parâmetros da rede para montar todos os cenários de treinamento, estes valores são descritos na lista abaixo.

- LR = [ $5 \cdot 10^{-4}$ ;  $1 \cdot 10^{-3}$ ;  $8 \cdot 10^{-3}$ ]
- Épocas = [900; 1500]
- Número de neurônios = [512; 1024]
- Profundidade = [2; 3]

Com os valores selecionados, foram gerados 24 cenários de aprendizado. A definição de cada um dos cenários é observado na Tabela 4.

Procedeu-se com o treinamento de todos os 24 cenários. Os resultados estão dispostos na Tabela 5.

Foi observada pouca diferença de performance na variação dos parâmetros, porém, o cenário 5 apresentou melhor resultado com o menor MAE, o terceiro maior  $R^2$  e o terceiro menor MAPE. Portanto, esta foi a configuração final do modelo DIPA.

**Tabela 4. Cenários de treinamento da rede para DIPA.**

Cenário	LR	Épocas	Número de Neurônios	Profundidade
1	$5 \cdot 10^{-4}$	900	512	2
2	$1 \cdot 10^{-3}$	900	512	2
3	$8 \cdot 10^{-3}$	900	512	2
4	$5 \cdot 10^{-4}$	1500	512	2
5	$1 \cdot 10^{-3}$	1500	512	2
6	$8 \cdot 10^{-3}$	1500	512	2
7	$5 \cdot 10^{-4}$	900	1024	2
8	$1 \cdot 10^{-3}$	900	1024	2
9	$8 \cdot 10^{-3}$	900	1024	2
10	$5 \cdot 10^{-4}$	1500	1024	2
11	$1 \cdot 10^{-3}$	1500	1024	2
12	$8 \cdot 10^{-3}$	1500	1024	2
13	$5 \cdot 10^{-4}$	900	512	3
14	$1 \cdot 10^{-3}$	900	512	3
15	$8 \cdot 10^{-3}$	900	512	3
16	$5 \cdot 10^{-4}$	1500	512	3
17	$1 \cdot 10^{-3}$	1500	512	3
18	$8 \cdot 10^{-3}$	1500	512	3
19	$5 \cdot 10^{-4}$	900	1024	3
20	$1 \cdot 10^{-3}$	900	1024	3
21	$8 \cdot 10^{-3}$	900	1024	3
22	$5 \cdot 10^{-4}$	1500	1024	3
23	$1 \cdot 10^{-3}$	1500	1024	3
24	$8 \cdot 10^{-3}$	1500	1024	3

Finalmente, foi realizado novo treinamento com a separação de dados *Hold Up*, que é uma separação direta dos dados de treino e teste. Com isto, foi retirada as métricas finais do modelo, que estão dispostas na lista abaixo.

- $R^2 = 0,617$
- MAE = 0,130
- MAPE = 0,058

### 6.1. Avaliação dos resultados para DIPA

Uma vez que a rede foi treinada e o modelo final salvo, foi gerado um gráfico do erro dos dados de validação ao longo das épocas para compreender a tendência do erro. A Figura 7 demonstra o MAE ao longo das épocas. Analisado a partir da época 0, observamos o comportamento do MAE é intensificar erros grandes por calcular o quadrado da diferença. Contudo, observa-se a tendência de queda do erro ao longo das épocas até a estabilização. Com isso conclui-se que não há necessidade de aumentar o número de épocas.

Em seguida, foi gerado um gráfico comparando a saída dos dados de teste e a saída predita pela rede neural. A Figura 8, consiste em um gráfico de pontos entre o peso de

**Tabela 5. Performance da DIPA para cada cenário de treinamento.**

<b>Cenário</b>	<b>Média R<sup>2</sup></b>	<b>Média MAE</b>	<b>Média MAPE</b>
<b>1</b>	0,471	0,159	0,070
<b>2</b>	0,491	0,154	0,069
<b>3</b>	0,508	0,165	0,073
<b>4</b>	0,535	0,143	0,063
<b>5</b>	<b>0,548</b>	<b>0,142</b>	<b>0,064</b>
<b>6</b>	0,515	0,152	0,068
<b>7</b>	0,458	0,155	0,069
<b>8</b>	0,499	0,147	0,066
<b>9</b>	0,554	0,200	0,087
<b>10</b>	0,532	0,154	0,069
<b>11</b>	0,536	0,152	0,068
<b>12</b>	0,554	0,155	0,068
<b>13</b>	0,489	0,158	0,069
<b>14</b>	0,528	0,143	0,064
<b>15</b>	0,470	0,202	0,088
<b>16</b>	0,532	0,142	0,063
<b>17</b>	0,499	0,147	0,066
<b>18</b>	0,523	0,158	0,070
<b>19</b>	0,476	0,156	0,070
<b>20</b>	0,489	0,157	0,070
<b>21</b>	0,520	0,283	0,122
<b>22</b>	0,466	0,155	0,069
<b>23</b>	0,546	0,150	0,066
<b>24</b>	0,536	0,148	0,065

teste e o peso inferido e foi obtida também a curva de tendência.

Na sequência foi, definido o erro da predição pela Equação 1 e gerado o histograma do erro pela Figura 9.

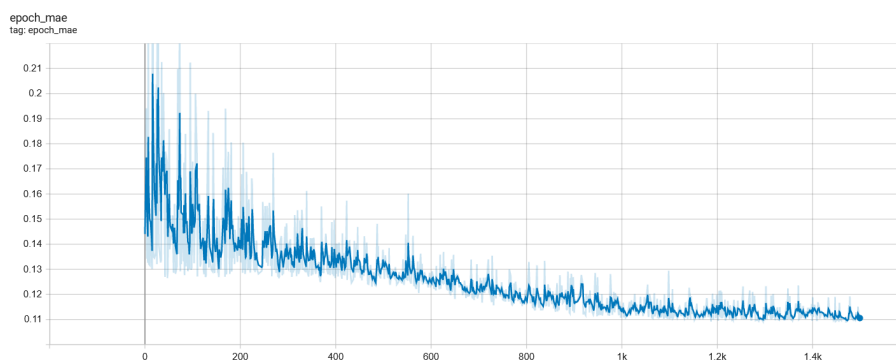
$$\text{Erro da predição} = y - \hat{y} \quad (1)$$

Como métrica final do trabalho, foi calculado a porcentagem de erro na predição do peso, esta métrica está definida na Equação 2.

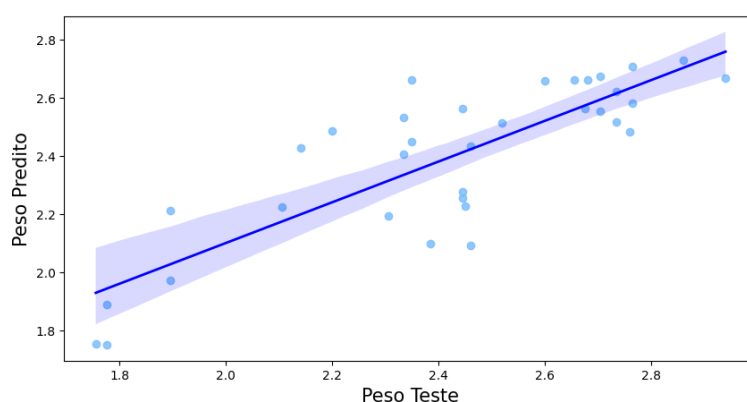
$$\text{Erro de medição} = \frac{MAE}{\text{Média do Peso}} \cdot 100 = \frac{0,130}{2,434} \cdot 100 = 5,34\% \quad (2)$$

## **7. Discussão dos Resultados**

A rede ORIS mostrou-se capaz de detectar diversas aves na mesma imagem, isso serve para permitir o aumento de amostragem de aves em menor período temporal. As elipses geradas e as imagens binárias resultantes foram satisfatórias, permitindo a extração das características geométricas conforme esperado. Consideramos que o volume de vídeo



**Figura 7. Acompanhamento do MAE ao longo das épocas.**



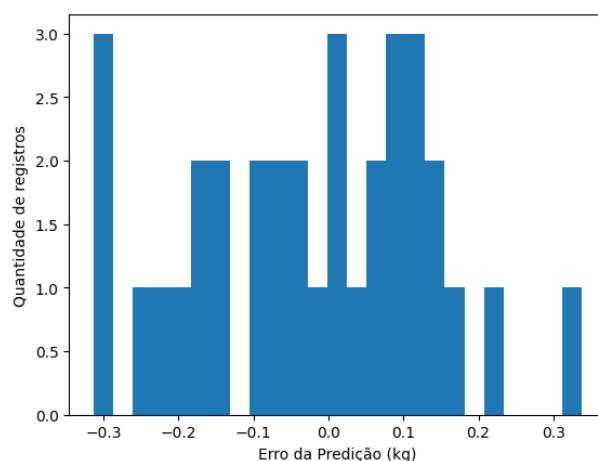
**Figura 8. Tendência de crescimento dos dados entre a saída real e a saída predita**

disponível e a quantidade de anotações realizadas foram suficientes para um treinamento eficiente da rede. Destaca-se também, a eficácia da delimitação da elipse com a exclusão da cauda e cabeça, dispensando a necessidade de pós-processamento.

A rede DIPA mostrou um resultado razoável com um erro médio de pesagem de 5,34%. Contudo, diversos pontos podem ser melhorados, a começar pela quantidade de aves amostradas. O trabalho de pesagem e gravação de vídeos para cada ave resultou num número reduzido de amostras. Graças a utilização da técnica de aumento de dados pudemos aumentar a quantidade de dados e chegar ao resultado apresentado. Estimamos que um total de 300 amostras poderia melhorar de maneira significativa os resultados.

Outro ponto que pode melhorar os resultados é a implementação de um mecanismo de rastreamento de aves. Esse mecanismo passaria a individualizar cada ave que entrasse no campo de visão da câmera, realizando o rastreamento dela pela imagem. Com isso, seria possível coletar um conjunto de imagens da mesma ave. Ao inferir o peso para cada imagem deste conjunto, seria possível eliminar as inferências que mais se distanciam da média geral, resultando em maior precisão. Contudo, apesar das limitações atuais, o resultado é promissor confirma que capacidade da técnica de inferência de peso por imagens.

A base de dados utilizada, as anotações e os algoritmos em Python estão publicamente disponíveis no Kaggle em <https://www.kaggle.com/datasets/lucasheilbuthh/inferring-broiler-chicken-weight>.



**Figura 9. Histograma do erro de predição**

## 8. Conclusão

Neste trabalho foram definidos dois modelos de redes neurais. O primeiro capaz de detectar e delinear aves de corte a partir de imagens. A segunda, capaz de inferir o peso das aves a partir de características geométricas extraídas das imagens. Para cada rede foram extraídas métricas de desempenho e selecionou-se os melhores parâmetros. Em seguida foi gerado uma série de gráficos para entender o comportamento do erro ao longo do treinamento e também foram extraídas as métricas de avaliação. Ao final, foi obtido um erro de predição do peso de 5,34%.

O erro obtido é considerado alto se comparado com a precisão de uma balança. Porém, o método proposto traz a vantagem de pesar a ave de forma automática e sem intervenção humana. Além disso, métodos tradicionais de pesagem utilizam poucas amostras para inferir a média da população. Neste método, é possível amostrar uma população muito maior de maneira contínua ao longo do manejo. Para redução do erro da predição do peso, planeja-se o aumento do banco de dados para o treinamento da DIPA. Neste trabalho amostrou-se apenas 46 aves e isso limitou a eficiência do modelo.

Todas as aves amostradas possuíam a idade de 25 dias. Como trabalho futuro, pretende-se gerar amostras com aves de diferentes idades, atribuindo o parâmetro idade à base de dados. Finalmente, pretende-se também aplicar o uso de métodos computacionais que automatizam a seleção dos parâmetros da rede neural, como a biblioteca optuna. O uso desta técnica permitirá explorar melhor os parâmetros da rede DIPA.

## Referências

- AVILA, V. S. D., FERREIRA, J. F. R., PIENIZ, L. C., LEDUR, M. C., ALBINO, L. F. T., and OLIVEIRA, P. A. V. D. (2007). Boas práticas de produção de frangos de corte. Circular interna, EMBRAPA.
- Beitzel, S. M., Jensen, E. C., and Frieder, O. (2009). *MAP*, pages 1691–1692. Springer US, Boston, MA.
- Cepea (2022). Pib do agronegócio. Indicadores, CEPEA.
- Charu, C. A. (2018). Neural networks and deep learning: a textbook.

- Dohmen, R., Catal, C., and Liu, Q. (2021). Image-based body mass prediction of heifers using deep neural networks. *Biosystems Engineering*, 204:283–293.
- Dutta, A. and Zisserman, A. (2019). The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, New York, NY, USA. ACM.
- Fine, T. L., Lauritzen, S. L., Jordan, M., Lawless, J., and Nair, V. (1999). *Feedforward Neural Network Methodology*. Springer-Verlag, Berlin, Heidelberg, 1st edition.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.
- Jierula, A., Wang, S., Oh, T.-M., and Wang, P. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. *Applied Sciences*, 11:2314.
- Kaloev, M. and Krastev, G. (2021). Comparative analysis of activation functions used in the hidden layers of deep neural networks. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5.
- Kwok, T.-Y. and Yeung, D.-Y. (1995). Efficient cross-validation for feedforward neural networks. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 5, pages 2789–2794 vol.5.
- Meshram, V., Patil, K., Meshram, V., Hanchate, D., and Ramkteke, S. (2021). Machine learning in agriculture domain: A state-of-art survey. *Artificial Intelligence in the Life Sciences*, 1:100010.
- Müller, A. and Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Incorporated.
- Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., and Padma, V. (2020). Study the influence of normalization/transformation process on the accuracy of supervised classification. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 729–735.
- Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *J. Big Data*, 6:60.
- Somaye Amraei, Saman A. Mehdizadeh, I. d. A. N. (2018). Development of a transfer function for weight prediction of live broiler chicken using machine vision. *Revista Engenharia Agrícola*, 38:776–782.
- Souza, P., Fornazier, A., Souza, H., and Ponciano, N. (2019). Diferenças regionais de tecnologia na agricultura familiar no brasil. *Revista de Economia e Sociologia Rural*, 57:594–617.
- Tolentino, L. K., Pedro, C., Icamina, J., Navarro, J., Salvacion, L., Sobrevilla, G., Villanueva, A., Amado, T., Padilla, M. V., and Madrigal, G. (2020). Weight prediction system for Nile tilapia using image processing and predictive analysis. *International Journal of Advanced Computer Science and Applications*, 11:399–406.