



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS SOBRAL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE
COMPUTAÇÃO (PPGEEC)

ANTONIO MÁRCIO ALBUQUERQUE ALMEIDA

DETECCÃO DE ANEMIA EM OVINOS ATRAVÉS DE APRENDIZAGEM
PROFUNDA EM IMAGENS DE MUCOSA OCULAR

SOBRAL, CEARÁ

2021

ANTONIO MÁRCIO ALBUQUERQUE ALMEIDA

DETECÇÃO DE ANEMIA EM OVINOS ATRAVÉS DE APRENDIZAGEM PROFUNDA EM
IMAGENS DE MUCOSA OCULAR

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação (PPGEEC) da Universidade Federal do Ceará, Campus Sobral, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica e de Computação. Área de Concentração: Algoritmos e Computação Distribuída

Orientador: Prof. Dr. Iális Cavalcante de Paula Júnior

SOBRAL, CEARÁ

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A298d Albuquerque Almeida, Antonio Márcio.

Detecção de anemia em ovinos através de aprendizagem profunda em imagens de mucosa ocular /
Antonio Márcio Albuquerque Almeida. – 2021.
111 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Campus de Sobral, Programa de Pós-Graduação
em Engenharia Elétrica e de Computação, Sobral, 2021.

Orientação: Prof. Dr. Iális Cavalcante de Paula Júnior.

1. FAMACHA. 2. Detecção de anemia e ovinos. 3. Segmentação. 4. Classificação. 5. Aprendizagem
Profunda. I. Título.

CDD 621.3

ANTONIO MÁRCIO ALBUQUERQUE ALMEIDA

DETECÇÃO DE ANEMIA EM OVINOS ATRAVÉS DE APRENDIZAGEM PROFUNDA EM
IMAGENS DE MUCOSA OCULAR

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação (PPGEEC) da Universidade Federal do Ceará, Campus Sobral, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica e de Computação. Área de Concentração: Algoritmos e Computação Distribuída

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Iális Cavalcante de Paula
Júnior (Orientador)
Universidade Federal do Ceará (UFC) - Campus de
Sobral

Prof^a. Dr. Deborah Maria Vieira Magalhães
Universidade Federal do Piauí (UFPI)

Dr. Marcel Teixeira
Empresa Brasileira de Pesquisa Agropecuária
(Embrapa) - Caprinos e Ovinos

Prof. Dr. Jarbas Joaci de Mesquita Sá Junior
Universidade Federal do Ceará (UFC) - Campus de
Sobral

Dedico este trabalho em memória ao meu amigo
Augusto Anderson de Lima Menezes.

AGRADECIMENTOS

Agradeço em especial a Deus, o autor do meu destino. Aos meus pais, Euclides e Liduina, por me orientar e cujo empenho em me educar sempre veio em primeiro lugar. A minha irmã Ana Beatriz, que sempre me ajuda nas dificuldades. A minha namorada Shely Oliveira, que é o pilar principal da minha vida.

Ao Prof. Dr. Iális Cavalcante de Paula Júnior, pela excelente orientação, pela amizade e conhecimento que pude adquirir ao longo desses anos.

Ao David Borges por me guiar na solução da segmentação.

Aos colegas Andrio Rodrigo Corrêa da Silva, Danilo Alves Oliveira, Luana Torres do Nascimento, Raimundo Farrapo Pinto Júnior, que ajudaram no processo de coleta de imagem do projeto.

Aos amigos, Acélio Luna, Brena Lima, Leila Rodrigues, Miguel Silva, Rafael Acurcio, Syllas Rangel, João Paulo, Joniel Bastos, Júnior Prado, do laboratório do Programa de Pós-Graduação em Engenharia Elétrica e de Computação (PPGEEC), que fizeram esses dois anos serem divertidos e alegres.

Aos grandes amigos Machado Neto, Marcos Pedrosa, Carlota Souza, Victória Tomé, pelo apoio durante os momentos do mestrado.

A Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) - Caprinos e Ovinos, localizada na região de Sobral - Ceará, por contribuir no processo de criação da base de dados para o projeto. E aos pesquisadores e demais colaboradores que auxiliaram no controle dos animais no processo de coleta de sangue. Agradecidamente ao idealizador do projeto Marcel Teixeira, por todo seu engajamento. Agradecimento aos colaboradores e Francisco Selmo Fernandes Alves, Eduardo Luiz de Oliveira, Luiz da Silva Vieira, Raymundo Rizaldo Pinheiro.

E à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), pelo financiamento de 20 meses da pesquisa de mestrado via bolsa de estudos.

“Quando você tem a capacidade de não falar, não ligar e não se importar, está aprendendo o que é ser forte.”

(Renato Russo)

RESUMO

O método FAMACHA é utilizado como instrumento para o diagnóstico parasitológico, podendo ser aplicado em ovelhas para constatação de diferentes graus de anemia. Este possui o objetivo de auxiliar na aplicação de vermífugo apenas para os animais doentes, porém ainda é um método que depende da interpretação humana. A tecnologia vem ajudando no processo de detecção automática e precisa de doenças. Neste trabalho realizamos a segmentação e classificação de imagens da mucosa do olho de ovinos para detecção de animais anêmicos baseada em modelos de aprendizado profundo. Este trabalho utilizou um total de 106 imagens distintas da mucosa ocular de ovinos, coletados no ambiente da EMBRAPA - Caprinos e Ovinos, localizada na região de Sobral - Ceará. Esses dados foram separados em conjunto de treino, validação e teste. Utilizando diversos modelos para a segmentação foi possível encontrar a região de interesse em todas essas imagens, sendo que o melhor modelo obteve uma precisão de 97,29 % no conjunto de teste, para o método de similaridade de Jaccard. Para o procedimento de classificação o melhor modelo obteve precisão de 95,23 % em detectar o estado de anemia do animal. Esse resultado foi obtido no conjunto de validação, com um total de 21 imagens. Com os modelos de segmentação e classificação já treinados foram realizados testes em uma aplicação *mobile*, em que o seu critério era a buscar o menor tempo de processamento, para segmentação foi de 0,375 (s) e para a classificação foi de 0,121 (s). Com as amostras coletadas também em outra base de dados disponibilizada por pesquisadores da Universidade Federal do Maranhão (UFMA), no qual o seu melhor resultado para segmentação foi de 78,76 %, no método de similaridade de Jaccard, já para classificação sua acurácia foi de 64.76%.

Palavras-chave: FAMACHA. Detecção de anemia e ovinos. Segmentação. Classificação. Aprendizagem Profunda.

ABSTRACT

The FAMACHA method is used as an instrument for parasitological diagnosis, and it can be applied to sheep to detect different degrees of anemia. This is intended to assist in the application of dewormer only for sick animals, but it is still a method that depends on human interpretation. The technology has helped in the process of automatic and accurate disease detection. In this work, we performed the segmentation and classification of images of the mucosa of the eye of sheep for the detection of anemic animals based on deep learning models. This work used a total of 106 distinct images of the ocular mucosa of sheep, collected in the environment of EMBRAPA - Goats and Sheep, located in the region of Sobral - Ceará. This dataset was separated into training set, validation set, and test set. Using several models for segmentation, it was possible to find the region of interest in all these images, and the best model obtained an accuracy of 97.29 % in the test set, for the Jaccard similarity method. For the classification procedure, the best model obtained an accuracy of 95.23 % in detecting the animal's anemia status. This result was obtained from the validation set, with a total of 21 images. With the segmentation and classification models already trained, tests were performed in a *mobile* application, in which the criterion was to seek the shortest processing time, for segmentation it was 0.375 (s) and for classification it was 0.121 (s). With the samples also collected in another database made available by researchers from UFMA, in which its best result for segmentation was 78.76 %, in the Jaccard similarity method, while for classification its accuracy was of 64.76%.

Keywords: FAMACHA. Detection of anemia and sheep. Segmentation. Classification. Deep Learning.

LISTA DE FIGURAS

Figura 1 – Exemplo do cartão FAMACHA.	17
Figura 2 – Etapas do hematócrito.	23
Figura 3 – Utilização do cartão FAMACHA para avaliação do grau de anemia em caprinos e ovinos.	24
Figura 4 – Local de coleta de imagens e sangue.	26
Figura 5 – Resultado da sequência de fotos.	27
Figura 6 – Aplicativo de coleta em <i>Android</i>	28
Figura 7 – Resultado de imagens do aplicativo.	28
Figura 8 – Comparando resultado do dispositivo <i>Android</i> com o aplicativo.	29
Figura 9 – Redimensionamento das imagens dos dispositivos.	30
Figura 10 – Fluxograma do processo de coleta da base de dados.	31
Figura 11 – Gráfico dos dados.	32
Figura 12 – Arquitetura típica de rede neural.	32
Figura 13 – Arquitetura da AlexNet.	33
Figura 14 – Arquitetura da U-Net.	34
Figura 15 – Aplicação da mudança de brilho.	36
Figura 16 – Modelo processo de aprendizagem.	38
Figura 17 – Fórmula do Jaccard <i>index</i>	40
Figura 18 – Aplicação da fórmula Jaccard <i>index</i>	41
Figura 19 – Fluxo da metodologia.	44
Figura 20 – Gráfico dos dados organizados para os experimentos.	45
Figura 21 – Gráficos das acurácias e perdas médias obtidas nos modelos <i>efficient residual factorized network</i> (ERFNet), sem normalização.	48
Figura 22 – Gráficos das acurácias e perdas médias obtidas nos modelos ERFNet, com normalização.	49
Figura 23 – Gráfico da acurácia e perda obtido pelo melhor modelo ERFNet, sem normalização.	50
Figura 24 – Gráfico da acurácia e perda obtido pelo melhor modelo ERFNet, com normalização.	51
Figura 25 – Gráficos das acurácias e perdas médias obtidas nos modelos U-Net, sem normalização.	53

Figura 26 – Gráficos das acurácias e perdas médias obtidas nos modelos U-Net, com normalização.	54
Figura 27 – Gráfico da acurácia e perda obtido pelo melhor modelo U-Net, sem normalização.	55
Figura 28 – Gráfico da acurácia e perda obtido pelo melhor modelo U-Net, com normalização.	56
Figura 29 – Gráficos das acurácias e perdas médias obtidas nos modelos <i>fully convolutional networks</i> (FCN), sem normalização.	58
Figura 30 – Gráficos das acurácias e perdas médias obtidas nos modelos FCN, com normalização.	59
Figura 31 – Gráfico da acurácia e perda obtido pelo melhor modelo FCN, sem normalização.	60
Figura 32 – Gráfico da acurácia e perda obtido pelo melhor modelo FCN, com normalização.	61
Figura 33 – Gráficos médio das acurácias e perdas obtidas nos modelos DeepLabV3, sem normalização.	63
Figura 34 – Gráficos médio das acurácias e perdas obtidas nos modelos DeepLabV3, com normalização.	64
Figura 35 – Gráfico da acurácia e perda obtido do melhor modelo DeepLabV3, sem normalização.	65
Figura 36 – Gráfico da acurácia e perda obtido do melhor modelo DeepLabV3, com normalização.	66
Figura 37 – Processo de segmentação de outra base de dados, sem normalização.	67
Figura 38 – Processo de segmentação de outra base de dados, com normalização.	68
Figura 39 – Aplicação <i>mobile</i> do modelo gerado pela FCN sem normalização.	70
Figura 40 – Processo de segmentação para o melhor modelo FCN, sem normalização.	73
Figura 41 – Gráficos das acurácias e perdas médias obtidas nos modelos AlexNet, sem normalização.	75
Figura 42 – Gráficos das acurácias e perdas médias obtidas nos modelos AlexNet, com normalização.	76
Figura 43 – Gráfico da acurácia e perda obtido pelo melhor modelo AlexNet, sem normalização.	77
Figura 44 – Gráfico da acurácia e perda obtido pelo melhor modelo AlexNet, com normalização.	78

Figura 45 – Gráficos das acurácias e perdas médias obtidas nos modelos MobileNetV2, sem normalização.	80
Figura 46 – Gráficos das acurácias e perdas médias obtidas nos modelos MobileNetV2, com normalização.	81
Figura 47 – Gráfico da acurácia e perda obtido pelo melhor modelo MobileNetV2, sem normalização.	82
Figura 48 – Gráfico da acurácia e perda obtido pelo melhor modelo MobileNetV2, com normalização.	83
Figura 49 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet18, sem normalização.	85
Figura 50 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet18, com normalização.	86
Figura 51 – Gráfico da acurácia e perda obtido pelo melhor modelo ResNet18, sem normalização.	87
Figura 52 – Gráfico da acurácia e perda obtido pelo melhor modelo ResNet18, com normalização.	88
Figura 53 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet152, sem normalização.	90
Figura 54 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet152, com normalização.	91
Figura 55 – Gráfico da acurácia e perda obtido do melhor modelo ResNet152, sem normalização.	92
Figura 56 – Gráfico da acurácia e perda obtido do melhor modelo ResNet152, com normalização.	93
Figura 57 – Gráficos das acurácias e perdas médias obtidas nos modelos <i>aggregated residual transformations for deep neural networks</i> (ResNeXt), sem normalização.	95
Figura 58 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNeXt, com normalização.	96
Figura 59 – Gráfico da acurácia e perda obtido do melhor modelo ResNeXt, sem normalização.	97
Figura 60 – Gráfico da acurácia e perda obtido do melhor modelo ResNeXt, com normalização.	98

Figura 61 – Imagens de teste da base da UFMA.	99
Figura 62 – Aplicação <i>mobile</i> do modelo gerado pela ResNeXt, sem normalização. . . .	101
Figura 63 – Imagens usadas como conjunto de teste para criação dos modelos da ResNeXt, sem normalização.	104

LISTA DE TABELAS

Tabela 1 – Relação do grau FAMACHA com a coloração da conjuntiva ocular e com o hematócrito.	25
Tabela 2 – Matriz de classificação.	41
Tabela 3 – Resultados médio da segmentação	68
Tabela 4 – Tempo de resposta no dispositivo móvel.	69
Tabela 5 – Resultados obtidos nos testes para os melhores modelos.	72
Tabela 6 – Resultados para a Figura 61a	99
Tabela 7 – Resultados para a Figura 61b	100
Tabela 8 – Estatísticas dos modelos.	100
Tabela 9 – Tempo de resposta celular android.	101
Tabela 10 – Resultados obtidos nos testes para os melhores modelos.	103

LISTA DE ABREVIATURAS E SIGLAS

CCE	perda de entropia cruzada (do inglês: <i>cross-entropy loss</i>)
CNN	rede neural convolucional (do inglês: <i>convolutional neural network</i>)
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
ERFNet	<i>efficient residual factorized network</i>
FCN	<i>fully convolutional networks</i>
GPU	Unidade Central de Gráfico (do inglês: <i>Graphics Processing Unit</i>)
Ht	hematócrito
IoU	Intersecção sobre União (do inglês: <i>intersection over Union</i>)
ResNeXt	<i>aggregated residual transformations for deep neural networks</i>
RGB	vermelho, verde, azul (do inglês: <i>red, green, blue</i>)
ROI	regiões de interesse (do inglês: <i>region of interest</i>)
SGB	descida gradiente estocástica (do inglês: <i>gradient descent method</i>)
UFC	Universidade Federal do Ceará
UFMA	Universidade Federal do Maranhão

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Trabalhos Relacionados	19
<i>1.1.1</i>	<i>Anemia Detection using Image Processing</i>	19
<i>1.1.2</i>	<i>Detection of anemia from image of the anterior conjunctiva of the eye by image processing and thresholding</i>	19
<i>1.1.3</i>	<i>Anemimetro: app móvel para implementação do método Famacha</i>	20
1.2	Objetivos	20
<i>1.2.1</i>	<i>Objetivos Específicos</i>	20
1.3	Organização	21
2	MATERIAIS E MÉTODOS	22
2.1	Hematócrito	22
2.2	FAMACHA	23
2.3	Processo de coleta da base de dados	26
2.4	Aprendizagem profunda	32
<i>2.4.1</i>	<i>Divisão dos dados</i>	35
<i>2.4.2</i>	<i>Aumento de dados para conjunto de treino</i>	35
<i>2.4.3</i>	<i>Transformação dos dados</i>	37
<i>2.4.4</i>	<i>Modelos</i>	37
<i>2.4.4.1</i>	<i>Função de perda</i>	39
<i>2.4.4.2</i>	<i>Otimização</i>	39
<i>2.4.5</i>	<i>Métricas</i>	39
<i>2.4.5.1</i>	<i>Segmentação</i>	40
<i>2.4.5.1.1</i>	<i>Jaccard index</i>	40
<i>2.4.5.2</i>	<i>Classificação</i>	41
3	METODOLOGIA	43
4	APLICAÇÃO E RESULTADOS EXPERIMENTAIS	45
4.1	Segmentação	46
<i>4.1.1</i>	<i>Modelos</i>	47
<i>4.1.1.1</i>	<i>Efficient Residual Factorized Network (ERFNet)</i>	47
<i>4.1.1.2</i>	<i>U-Net</i>	52

4.1.1.3	<i>Fully Convolutional Networks (FCN)</i>	57
4.1.1.4	<i>DeepLabV3</i>	62
4.1.2	<i>Aplicação em outra base de dados</i>	67
4.1.3	<i>Aplicação no celular</i>	69
4.1.4	<i>Análise dos resultados</i>	70
4.2	Classificação	73
4.2.1	Modelos	73
4.2.1.1	<i>AlexNet</i>	73
4.2.1.2	<i>MobileNetV2</i>	79
4.2.1.3	<i>Deep Residual Learning (ResNet)</i>	84
4.2.1.3.1	<i>ResNet18</i>	84
4.2.1.3.2	<i>ResNet152</i>	89
4.2.1.4	<i>Aggregated Residual Transformations for Deep Neural Networks (ResNeXt)</i>	94
4.2.2	<i>Aplicação em outra base de dados</i>	99
4.2.3	<i>Aplicação no celular</i>	101
4.2.4	<i>Análise dos resultados</i>	102
5	CONCLUSÃO	105
5.1	Conclusões	105
5.1.1	<i>Verificação dos objetivos propostos</i>	106
5.2	Perspectivas futuras	107
	REFERÊNCIAS	108

1 INTRODUÇÃO

A criação de ovinos e caprinos é uma cultura representativa no sertão do Ceará. Segundo o IBGE, no ano de 2016, o Ceará possuía o quarto maior rebanho de caprinos e ovinos do país (CARLÔTO, 2018). Dessa forma, o cuidado e trato com esses animais se torna de suma importância e traz impacto direto na economia em nível local e estadual. Na ovinocultura, os animais podem sofrer com doenças que levam ao estado anêmico, podendo-se citar doenças parasitárias, como *Haemonchus contortus*. Por isso, para prevenir ou tratar a doença, o produtor faz uso de medicações aplicadas em todo o rebanho periodicamente, acarretando um maior custo para a produção.

A detecção de anemia pode ser feita com uso do método FAMACHA, que utiliza um cartão guia ilustrativo, conforme ilustrado na Figura 1. Esse método foi desenvolvido na região da África do Sul e, após um longo período de estudo, estabeleceu-se uma correlação entre a coloração da mucosa do olho de pequenos ruminantes (ovelhas e cabras) com cinco intervalos de anemia (VAN WYK; MALAN; BATH, 1997; KAPLAN *et al.*, 2004). Desse modo, é possível constatar a doença no animal mais rapidamente e de forma satisfatório. Além disso, seria menos custoso para a produção pois não é necessário aplicar o vermífugo em todos os animais.

Figura 1 – Exemplo do cartão FAMACHA.



Fonte: (MEDEIROS, 2020).

A utilização do cartão FAMACHA é um método manual, em que a mucosa do olho de ovinos é comparada a cinco graus de coloração. As cores em tons de vermelho na região inferior do cartão simbolizam o aspecto de saúde do animal. Da esquerda para a direita, há uma variação dos graus 1 a 5. Os graus 1 e 2 são de animais com coloração vermelha mais intensa, ou seja, praticamente sem traços de anemia. No grau 3, já é sugerida a vermifugação. Nos graus 4 e 5, a vermifugação é imprescindível, pois a mucosa apresenta palidez intensa (CHAGAS *et*

al., 2007). Esse método é aplicado especialmente quando os animais são predominantemente infectados por *Haemonchus contortus*. Dessa forma, a técnica FAMACHA parece ser um método adequado para detectar infecções por parasitas (MOORS; GAULY, 2009).

Mesmo com a utilização desse método, é possível que haja de falhas, que podem ser atribuídas a diversos fatores, como luminosidade, estresse do animal no momento da análise da mucosa, além de erro humano ao comparar a mucosa do animal ao cartão (MOORS; GAULY, 2009; DEMOLINER; ALVES, 2017; VAN WYK; BATH, 2002). O método mais preciso é o teste em laboratório chamado hematócrito, em que é avaliada a porcentagem de volume ocupado pelos glóbulos vermelhos ou hemácias em relação ao volume total de sangue. Entretanto, esse teste demanda tempo e custo para o produtor (CHAGAS *et al.*, 2007).

A análise de anemia através do cartão FAMACHA se torna mais interessante para o produtor, porém ainda é um processo que depende da análise humana. Os sistemas computacionais atuais facilitam a análise de doenças através de sistema de imagens e reconhecimento de padrões. Neste contexto, relevante a construção de uma metodologia que utilize técnicas de processamento de imagens e inteligência computacional para classificação da coloração da mucosa do animal, baseado no método do cartão FAMACHA e teste laboratorial hematócrito.

Para o desenvolvimento desse sistema computacional pode-se explorar duas áreas: Processamento de imagem e Reconhecimento de padrões. Para o processamento de imagem é interessante usar métodos de segmentação a fim de destacar a região de interesse do restante da imagem. Nesse caso, a região da mucosa do animal é a região mais relevante para o processo de reconhecimento da anemia do animal. Porém, uma das dificuldades associada à segmentação é a localização da região de interesse, pois a imagem pode conter diversas informações não relevantes para o processo de reconhecimento de padrões (GONZALEZ; WOODS, 2010; SZELISKI, 2010; RUSSELL *et al.*, 2010).

Com a utilização de aprendizado profundo e as redes neurais que utilizam dessa técnica, chamadas redes neurais profundas, é possível extrair padrões e características das imagens com bastante eficiência. Suas estruturas baseadas em agrupamentos de camadas associadas a algoritmos de treinamento como a retropropagação (*backpropagation*) auxiliam na variação dos pesos dos neurônios da rede e na função de erro. Esse método pode ser aplicado no processo de segmentação para destacar a região da mucosa do animal.

Portanto, o presente trabalho aborda a segmentação e classificação da região de interesse com aprendizagem profunda nas imagens da mucosa do olho de ovinos, com uma

base de dados própria desenvolvida em colaboração com a EMBRAPA - Caprinos e Ovinos, localizada na região de Sobral - Ceará, que foi coletadas através de diversos celulares. O projeto propõe a união da ideia do método FAMACHA com a precisão dos dados laboratoriais para o desenvolvimento de um sistema computacional que utilize o conceito de aprendizagem profunda, para o destaque da região da mucosa nas imagens e o reconhecimento da saúde do animal.

1.1 Trabalhos Relacionados

Nesta seção são descritos alguns trabalhos relacionados com a proposta do projeto. Em particular, consideramos os trabalhos Irum *et al.* (2016), Tamir *et al.* (2017) e Demoliner e Alves (2017).

1.1.1 Anemia Detection using Image Processing

Os autores Irum *et al.* (2016) propõem uma detecção automática não invasiva de anemia em pessoas utilizando técnicas de processamento de imagem e reconhecimento de padrões. Esses recursos são aplicados a imagens digitais do rosto expondo a conjuntiva, pois a anemia pode ser identificada pela palidez da mesma. A segmentação da região conjuntiva é realizada usando o agrupamento K-means, mas antes desse processo aplica-se um aprimoramento nas imagens. Nessa etapa, aplica-se um histograma de equalização no canal V (*value*), da estrutura de imagem HSV (*hue*: define a cor, *saturation*: pureza da imagem, quanto maior o valor menos tons de cinza são encontrados e *value*: que define o brilho da imagem). Após a extração da conjuntiva, o conjunto de recursos foi calculado para cada imagem, para ser utilizado na etapa de treinamento. O classificador Máquina de Vetores de Suporte é aplicado para classificar entre imagens anêmicas e não anêmicas. O algoritmo proposto alcançou 85% de precisão, utilizando uma base de imagens com anêmicos, 56 imagens, e não anêmicos, 21 imagens.

1.1.2 Detection of anemia from image of the anterior conjunctiva of the eye by image processing and thresholding

Os autores Tamir *et al.* (2017) trabalharam na coleta de fotografias digitais do olho captadas com uma câmera de *smartphone* de resolução apropriada sob condições de iluminação adequadas, com a ajuda de uma aplicação android que foi desenvolvida por eles. Essas imagens são então processadas para obter os espectros dos componentes vermelho e verde da cor, isso

através do modelo de cor RG, da conjuntiva e comparadas com um limiar para determinar se o paciente está anêmico ou não. Esse limiar adequado é estimado e, se a diferença entre os dois espectros vermelho e verde for maior que o valor limite, o algoritmo rotula esse paciente como Não Anêmico. Foi empregado o método em 19 pessoas com níveis de hemoglobina conhecidos. Os resultados obtidos no processo concordaram com o relatório de sangue do paciente em 15 dos 19 casos, o que se traduz em uma precisão de 78,9%.

1.1.3 Anemímetro: app móvel para implementação do método Famacha

Os autores Demoliner e Alves (2017) apresentam uma ideia bem similar à problemática proposta usando a metodologia que consiste no uso do aplicativo de celular. Ao todo foram 414 imagens. Foi realizada a coleta das imagens através do aplicativo e as imagens tiveram que ser recortadas manualmente, uma a uma, para que fosse possível manter na imagem a ser processada somente a conjuntiva do animal. O processo a identificação da região de interesse foi feita de forma manual. O aplicativo utiliza o padrão de cores HSV. Eles usaram classificador Naive Bayes com o histograma de cada imagem para captar a frequência e as semelhanças de pixels e cores. Foram utilizadas 66% das imagens para o treinamento e 34% para a fase de testes. Para os testes de classificação usando os cinco graus FAMACHA, foi alcançada uma taxa de acerto de 50,6%. Ao mudar a classificação para dois grupos, rotulados como Tratar (Graus 3 a 5) ou Não tratar (Graus 1 e 2), a taxa de acerto aumentou em 16%, chegando a 66,3%.

1.2 Objetivos

Desenvolver uma metodologia que utilize o banco de dados de imagem construído neste projeto para reconhecer a região da mucosa de ovinos, e a partir desta determinar o estado anêmico, por meio aprendizagem profunda.

1.2.1 Objetivos Específicos

1. Criar e organizar uma base de dados de imagens da mucosa do olho de ovinos para divulgação e pesquisa científica;
2. Analisar diferentes modelos de aprendizagem profunda com a função de segmentação da região da mucosa do olho de ovinos;
3. Analisar diferentes modelos de aprendizagem profunda com a função de classificar se a

região da mucosa do olho de ovinos apresenta sinais de anemia ou não.

1.3 Organização

Os capítulos desta dissertação estão divididos como apresentado abaixo:

- **Capítulo 2:** apresentação de uma resenha das técnicas utilizadas para criação da base de dados, recolhimento de dados laboratoriais, técnicas de aprendizagem profunda, tratamento dos dados;
- **Capítulo 3:** descrição da metodologia proposta, os passos adotados para obtenção dos resultados da segmentação e classificação utilizando aprendizagem profunda;
- **Capítulo 4:** apresentação dos experimentos realizados durante a aplicação da metodologia proposta e os resultados obtidos pelas arquiteturas de rede neural convolucional (do inglês: *convolutional neural network*) (CNN), das segmentações e classificações;
- **Capítulo 5:** apresenta uma discussão sobre os objetivos alcançados, suas contribuições e as perspectivas dos trabalhos futuros.

2 MATERIAIS E MÉTODOS

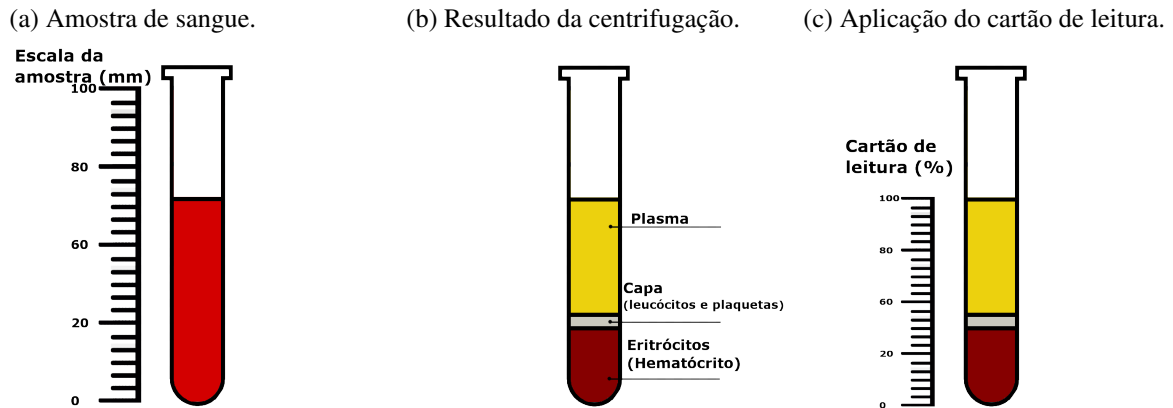
2.1 Hematócrito

O hematócrito (Ht), também conhecido com a abreviatura Hct, é uma medida laboratorial que resulta na porcentagem de volume ocupada pelos eritrócitos (conhecidos também como células vermelhas, glóbulos vermelhos, hemácias) no volume total de sangue da amostra coletada. Sua utilização é importante para identificar a saúde do animal, pois em algumas situações de doenças parasitárias levam o animal ao estado anêmico, como a verminose *Haemonchus contortus* (CLIMENI *et al.*, 2008). Esse estado de anemia causa uma baixa quantidade de volume ocupada pelos eritrócitos no sangue do animal.

As técnicas laboratoriais para a medição do Ht podem ser apresentadas da seguinte forma (DAYYAL, 2017; LACVET, 2018; KASVI, 2019; AZEVEDO, 2019):

1. O tubo capilar tem que estar preenchido com 3/4 de sangue da sua altura, apresentado na Figura 2a:
2. O capilar em uma centrífuga apropriada, o tempo de centrifugação varia dependendo do animal:
 - a) Após o tempo necessário a amostra será separada em três partes, conforme ilustrado na Figura 2b. São elas: plasma, capa, que são os leucócitos e as plaquetas, e por fim os eritrócitos.
3. Para análise do resultado é aplicado um cartão de leitura em que se limitam as marcas de 0 a 100, observando na escala o limite de separação dos eritrócitos com o plasma. Em uma das extremidades do eritrócito fica o índice 0 e em uma das extremidades do plasma fica o índice 100 do cartão de leitura. O resultado é expresso em porcentagem de eritrócitos em relação ao sangue total. Por exemplo, na Figura 2c, os eritrócitos chegam até o índice 32 do cartão de leitura. Isso significa que o resultado do seu Ht foi de 32 % para o volume ocupado na amostra preenchida.

Figura 2 – Etapas do hematócrito.



Fonte: adaptação de LACVET (2018).

Algumas falhas podem ocorrer no processo laboratorial, tais como: falha na obtenção do sangue, falha na interpretação do cartão de leitura, erro no tempo da centrifugação, irregularidade no diâmetro do capilar, dentre outras. Ainda assim, esse processo laboratorial é considerado o mais preciso, e é frequentemente utilizado em pesquisas em diferentes animais a fim de medir a anemia. Com esse método é possível compreender os estados de saúde do animal em certos períodos de tempo e analisar se a alimentação afeta os seus níveis do Ht, ou se o ambiente faz com que o animal desenvolva anemia (WEISS; WARDROP, 2011). Serão utilizado neste projeto, os valores do Ht para separar as classe dos animais em duas categorias: anêmicos e saudáveis.

2.2 FAMACHA

Com a precisão do método Ht, apresentado na Seção 2.1, ele aponta dificuldades na aplicabilidade para o produtor, tais como:

- **Tempo elevado de processamento:** O produtor irá coletar o sangue do rebanho, em seguida, mandar para um laboratório e aguardar o resultado. E, por fim, separar o mesmo rebanho novamente com aqueles animais que apresentaram anemia no resultado do Ht, para então neste subgrupo do rebanho aplicar a vermifugação. Esse processo tem que ser repetido periodicamente;
- **Custo operacional:** Para o pequeno produtor, a utilização do Ht se torna impraticável, pois o custo de operação seria muito oneroso. O pequeno produtor pode optar por fazer a vermifugação em todo o seu rebanho, com isso ele faz a aplicação do vermífugo nos animais anêmicos e nos saudáveis, provocando um efeito futuro de maior resistência do

parasita ao vermífugo aplicado.

Outro método que pode solucionar as dificuldades mencionadas foi desenvolvido pelos autores Bath, Malan e Wyk (1996), que o nomearam de FAMACHA. Esse estudo foi desenvolvido na região da África do Sul. Após um longo período foi estabelecida uma correlação entre a coloração da mucosa do olho de pequenos ruminantes, como ovelhas e cabras, com cinco intervalos de anemia indicados pelos seus níveis do Ht, apresentado na Seção 2.1 (VAN WYK; MALAN; BATH, 1997; KAPLAN *et al.*, 2004). O método FAMACHA é um tratamento seletivo, cujo objetivo é vermifugar somente os animais do rebanho que apresentam anemia, isso apenas com a comparação do cartão FAMACHA e mucosa do olho (DIAS, 2017). A vantagem da aplicação do método é a redução do número de tratamentos aplicados, ocasionando uma economia na produção do rebanho e agilidade no processo de vermifugação (DIAS, 2017).

O método manual da utilização do cartão FAMACHA consiste em uma técnica aplicada para mucosa do olho de ovinos, que se configura em cinco graus de coloração, ilustrados em um cartão, como mostrado na Figura 3, direcionando a vermifugação dos animais. Os graus 1 e 2 são de animais com coloração bem vermelha, ou seja, praticamente sem traços de anemia. No grau 3, já é indicada a vermifugação. Nos graus 4 e 5, a vermifugação é imprescindível, pois a mucosa apresenta palidez intensa (CHAGAS *et al.*, 2007). Tal recurso é aplicado especialmente quando os animais são predominantemente infectados por *Haemonchus contortus* (MOORS; GAULY, 2009).

Figura 3 – Utilização do cartão FAMACHA para avaliação do grau de anemia em caprinos e ovinos.



Fonte: (DIAS, 2017).

Esse método apresenta falhas, que estão atribuídas a diversos fatores, como luminosidade, estresse do animal no momento da análise da mucosa, além de erro humano ao comparar a mucosa do animal ao cartão (VAN WYK; BATH, 2002; PETERSSON; KASS, 2016; DEMOLINER; ALVES, 2017). Pesquisadores vêm aplicando em várias áreas testando sua eficiência, sendo possível citar: Aplicação como um recurso para o controle da verminose em ovinos, um estudo para levantar a questão se esse método é adequado para todas as raças, sua utilização para diferentes regiões do planeta (KAPLAN *et al.*, 2004; CHAGAS *et al.*, 2007; MOORS; GAULY, 2009). A literatura aponta testes de eficiência em várias áreas, podendo citar: Macedo *et al.* (2014) que avaliaram a precisão com os valores de Ht, e a classificação pelo método FAMACHA, chegando a um resultado de 79,91 % de correlação dos dados, que é considerado para o estudo.

Para interpretar o valor do resultado do Ht, foi utilizada a Tabela 1, fundamentada no artigo dos autores Chagas *et al.* (2007). Como exemplo, um resultado de exame igual a 22 %. Analisando o valor na Tabela 1, é possível concluir que nesse caso o animal estava com grau 3 do FAMACHA e em um estado anêmico.

Tabela 1 – Relação do grau FAMACHA com a coloração da conjuntiva ocular e com o hematócrito.

Grau FAMACHA	Hematócrito (%)	Laudo clínico
1	>27	Não anêmico
2	23 a 27	Não anêmico
3	18 a 22	Anêmico
4	13 a 17	Anêmico
5	<13	Anêmico

Fonte: adaptação do artigo dos autores Chagas *et al.* (2007).

2.3 Processo de coleta da base de dados

Por se tratar de um trabalho com uma aplicabilidade para a pecuária, não foi possível encontrar uma base de dados significativa, que contivesse imagens da mucosa do olho de ovinos, e que estivessem catalogadas através do Ht. A etapa inicial do projeto consistiu em coletas das imagens da mucosa do olho do animal.

O processo de coleta da base de dados foi realizado na EMBRAPA - Caprinos e Ovinos, localizada na região de Sobral - Ceará, e foi selecionado um ambiente com uma iluminação controlada e também que fosse mais fácil o manuseio do animal, como na Figura 4. Também foi estabelecido pela equipe apenas a coleta de imagens de uma espécie, nesse caso os ovinos, com foco nas raças Santa Inês, Morada Nova, Somalis. Porém, havia ainda animais sem raça definida. Os colaboradores da EMBRAPA - Caprinos e Ovinos eram responsáveis pelo manuseio do animal, a coleta do sangue e a realização do Ht, enquanto os estudantes da Universidade Federal do Ceará (UFC) - Campus de Sobral responsáveis pela coleta das imagens.

Figura 4 – Local de coleta de imagens e sangue.



Fonte: o autor.

Com a definição dos dispositivos de aquisição, visando uma melhor qualidade nas imagens, foi inicialmente usado uma câmera semi-profissional. Porém, houveram dificuldades no registro das fotos porque algumas imagens saíam desfocadas em decorrência das movimentações e agitações de corpo realizadas pelos animais. Nas coletas seguintes foram utilizados apenas *smartphones*. Para ter imagens mais diversificadas foi admitido o uso de vários dispositivos *mobiles*, com sistema operacional *Android* e *iOS*.

Na etapa de coletar as imagens, através desses dispositivos, adotou-se a seguinte sequência de organização de imagens: Brinco do animal (Figura 5a); Pelagem (Figura 5b); Mucosa do olho (Figura 5c). Com as fotos da numeração do brinco do animal, foi possível associar os valores do resultado do Ht, e com as fotos da pelagem era realizada a verificação da raça do animal. Foram capturadas no mínimo duas imagens da mucosa por animal. Essa sequência foi apresentada na Figura 5. No entanto, houveram dificuldades nesse processo, tais como: não ser possível visualizar as imagens no ato da coleta; Demora para coletar a imagem da mucosa; Muitas imagens desnecessárias para futuras aplicações (brinco do animal e pelagem), o momento da organização das imagens tornava o processo mais demorado; Algumas inconsistência entre as imagens e as informações resultantes do Ht, medido em laboratório.

Figura 5 – Resultado da sequência de fotos.

(a) Brinco do animal.



(b) Pelagem.



(c) Mucosa do olho.



Fonte: o autor.

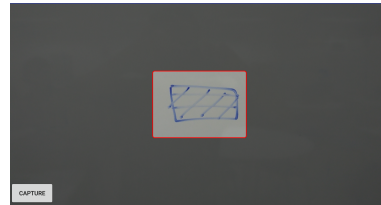
Para solucionar os problemas dessa forma de coleta foi desenvolvido um aplicativo em *Android*. Essa aplicação é composta de 3 telas, como apresentado na Figura 6. A primeira tela tem a função de coletar informações, a respeito da identificação do animal e sua raça, como apresentado na Figura 6a. A segunda tela é a estrutura da câmera, composta de um quadrado para representar a área que será coletada na imagem, como apresentado na Figura 6b. A terceira tela é a de visualização, que permite perceber se a imagem foi coletada corretamente, apresentado na Figura 6c. O resultado do uso do aplicativo na coleta foi apresentado na Figura 7. Cada imagem

final tem um tamanho de 800x800 *pixels*.

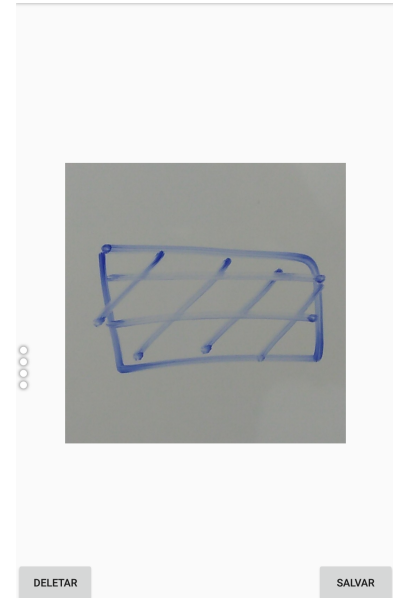
Figura 6 – Aplicativo de coleta em *Android*.

(a) Entrada de informação.

(b) Câmera.



(c) Deletar ou Salvar.



Fonte: o autor.

Figura 7 – Resultado de imagens do aplicativo.

(a) Grau 1 - Santa Inês.



(b) Grau 2 - Somalis.



(c) Grau 3 - Morada Nova.



Fonte: o autor.

O aplicativo proporcionou diferentes facilidades, tais como:

- Agilidade na coleta de informações do animal, pois as informações já eram relacionadas às imagens coletadas, e assim descartando a necessidade de coletar as imagens do brinco e da pelagem;

- A possibilidade de visualização da imagem, podendo selecionar as imagens com melhor definição para salvar ou deletar uma imagem desfocada ou que não tinha a região da mucosa, isso no próprio ato da coleta;
- Proporcionou uma facilidade na organização das imagens, através de um algoritmo que realizava uma associação de forma automática das informações do Ht, enviadas pela EMBRAPA, com as imagens coletadas, pelo aplicativo.

Contudo, ainda com o algoritmo de organização alguns erros foram encontrados na associação das imagens ou dados fornecidos pela EMBRAPA. Entende-se que esses erros envolviam faltar com os dados de laboratório frente às imagens coletadas (aqui as imagens eram excluídas da base de dados); ou ainda, havia os dados de laboratório mas os mesmos não correspondiam a nenhuma das imagens coletadas. Erros simples, como uma interpretação errada do número do brinco do animal ou na raça do animal. A análise e organização das imagens, incluindo as fotografias coletadas de outros dispositivos, evitava exclusão de arquivos. Quando não era possível chegar a um consenso, então as imagens eram apagadas da base de dados final.

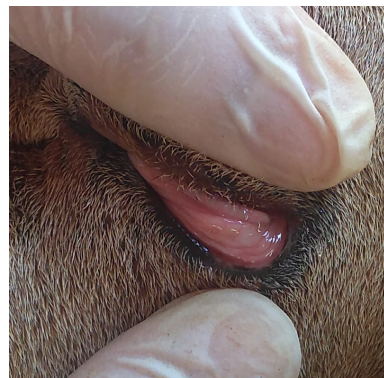
A Figura 8 apresenta a comparação de duas imagens coletadas no mesmo dia e do mesmo animal, uma através da câmera do dispositivo e outra através do processamento do aplicativo. Com essa comparação é possível observar que a imagem do aplicativo contém menos informações desnecessárias do que a imagem coletada pelo dispositivo. Ainda assim, com a vantagem do aplicativo, optou-se por continuar a utilização das câmeras dos dispositivos *mobiles*, *Android* e *iOS*. Isso foi admitido para termos uma base de dados com imagens diversificadas no sentido de tamanho e forma de coleta.

Figura 8 – Comparando resultado do dispositivo *Android* com o aplicativo.

(a) Dispositivo *Android*.



(b) Aplicativo.



Fonte: o autor.

Depois que o resultado do Ht era provido pela EMBRAPA, teve início o processo de organização das imagens, feito pelo próprio pesquisador. Três processos de organização dos dados foram utilizados e são destacados a seguir:

- **Seleção de imagens:** Era analisada manualmente imagem por imagem, e excluído da base de dados final as imagens avaliadas como possivelmente problemáticas para uma futura aplicação, como fotografias com a região da mucosa do olho do animal muito desfocada, com efeito de movimentação, muito brilho ou com uma grande interferência dos dedos do tratador na região da mucosa do olho do animal;
- **Renomeação das imagens:** Eram catalogadas as imagens com o nome do arquivo, contendo os seus dados de número do brinco, sua raça e valor do resultado do Ht;
- **Redimensionamento das imagens:** No processo final as imagens dos dispositivos foram redimensionadas para o mesmo tamanho do aplicativo, nesse caso um tamanho de 800x800 *pixels*. Utilizou-se um algoritmo em que todas as imagens eram abertas e de forma manual era selecionado um ponto “central” na região da mucosa do olho do animal. O objetivo desse processo era criar uma base de dados com a localização da região da mucosa, conforme mostrado na Figura 9.

Figura 9 – Redimensionamento das imagens dos dispositivos.

(a) Amostra coletada por dispositivo *Android*.



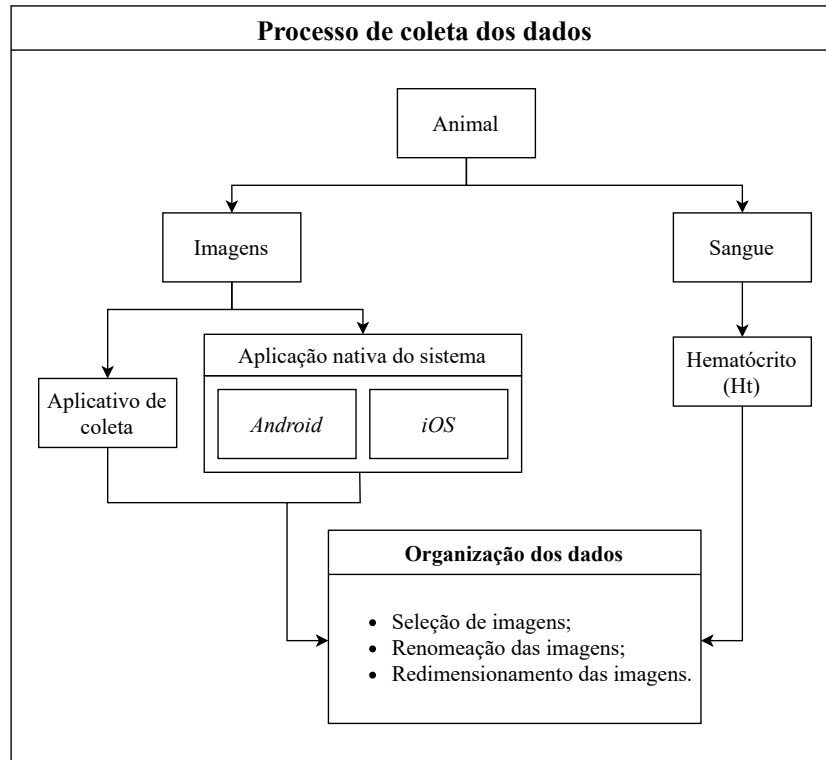
(b) Mesma amostra após redimensionamento a partir da região da mucosa.



Fonte: o autor.

Todo o processo realizado para criação da base de dados pode ser visualizado no fluxograma da Figura 10.

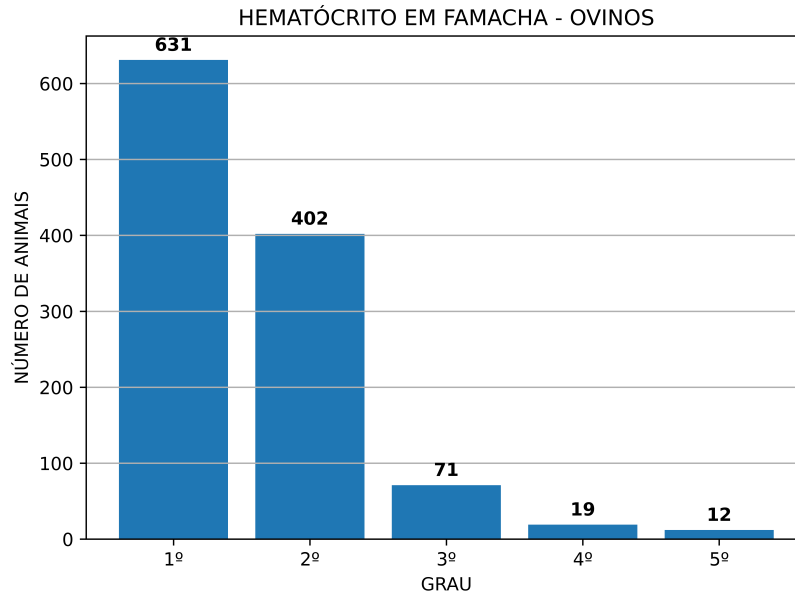
Figura 10 – Fluxograma do processo de coleta da base de dados.



Fonte: o autor.

Até o momento foram realizadas 27 coletas, entre os dias 27/08/2018 até 14/01/2020, totalizando um número de 7232 imagens organizadas. Cada coleta foi realizada em dias diferentes, assim poderíamos coletar o mesmo animal de uma coleta realizada anteriormente, pois ao passar de dias o Ht pode apresentar resultados diferentes. Para entendimento dos dados, foi feito um levantamento com os animais reunidos em todas as coletas sobre os seus níveis FAMACHA, conforme a Figura 11. As conclusões inferidas através do gráfico apontam que há uma grande desproporcionalidade nos dados, havendo mais animais saudáveis, no nível de grau 1° e 2° do FAMACHA, do que animais doentes, no nível de grau 3°, 4° e 5°. Considerando o nível de grau 3° como doente apto a ser vermifugado, foi admitido que as coletas continuariam a ser realizadas até que mais dados de animais doentes fossem obtidos, especialmente do 4°. O 5° representa dados mais escassos pois trata de animais com a saúde bem fragilizada e com a alta probabilidade de morte e são mais difíceis de obter.

Figura 11 – Gráfico dos dados.

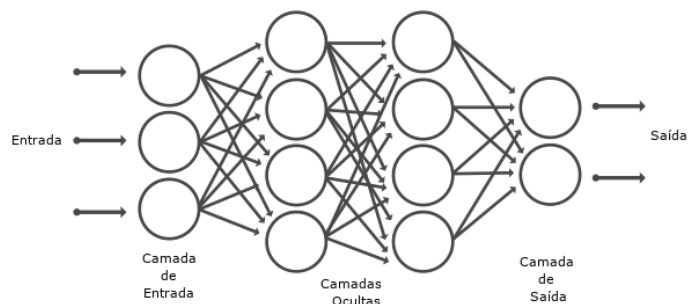


Fonte: o autor.

2.4 Aprendizagem profunda

As arquiteturas de aprendizagem profunda, tais como, *deep belief networks (DBN)*, *CNN*, *deep stacking networks (DSNs)*, são baseadas em redes neurais, então são estruturados por conexões e camadas que formam uma sequência temporal, com isso podendo aprender com os dados, assim reconhecendo os padrões deles (JONES, 2017). A Figura 12 ilustra essas estruturas de conexões e camadas, em que a Entrada: representa os dados brutos a serem analisados; Camada de entrada: são responsáveis pelo recebimento dos dados, também com a associação com os pesos de entrada; Camadas ocultas: extraem as informações associadas aos dados, sendo também responsáveis pela maior parte do processamento deles; Camada de saída: agrega os dados das camadas anteriores e ativa uma resposta adequada; Saída: a resposta de rede neural (KHAN *et al.*, 2020).

Figura 12 – Arquitetura típica de rede neural.

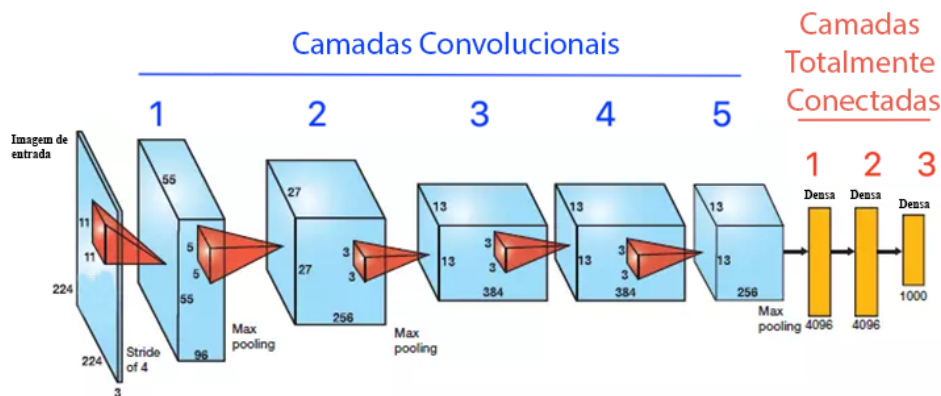


Fonte: adaptação do autor (MATHWORKS, 2021)

A rede neural profunda mais popular na aplicação no aprendizado de imagens é a CNN ou **ConvNet** (MATHWORKS, 2021). A CNN utiliza esses recursos de imagens para aprender sobre a base de dados. Ela aprende a detectar diferentes características de uma imagem usando dezenas ou centenas de camadas ocultas. Quanto mais camadas ocultas maior é a complexidade da rede. Cada camada oculta pode apreender sobre uma determinada característica da imagem. Por exemplo, a primeira camada oculta pode aprender como detectar bordas, e a última aprender como detectar formas mais complexas especificamente relacionadas à forma do objeto que estamos tentando reconhecer (MATHWORKS, 2021; SULTANA; SUFIAN; DUTTA, 2020).

Cada CNN pode ter arquiteturas distintas e ter mudanças simples, como um filtro diferente e com isso aumentar mais a precisão da rede. É possível citar uma rede difundida como a AlexNet, desenvolvida por Krizhevsky (2014), que uma é arquitetura simples para os modelos definidos atualmente, conforme a Figura 13.

Figura 13 – Arquitetura da AlexNet.



Fonte: adaptado de SACHAN (2017)

A AlexNet é composto por 5 camadas convolucionais seguidas de 3 camadas totalmente conectadas. A imagem de entrada terá as dimensões 224x224x3 (vermelho, verde, azul (do inglês: *red, green, blue*) (RGB)) e a mesma será processado pelas camadas que seguem:

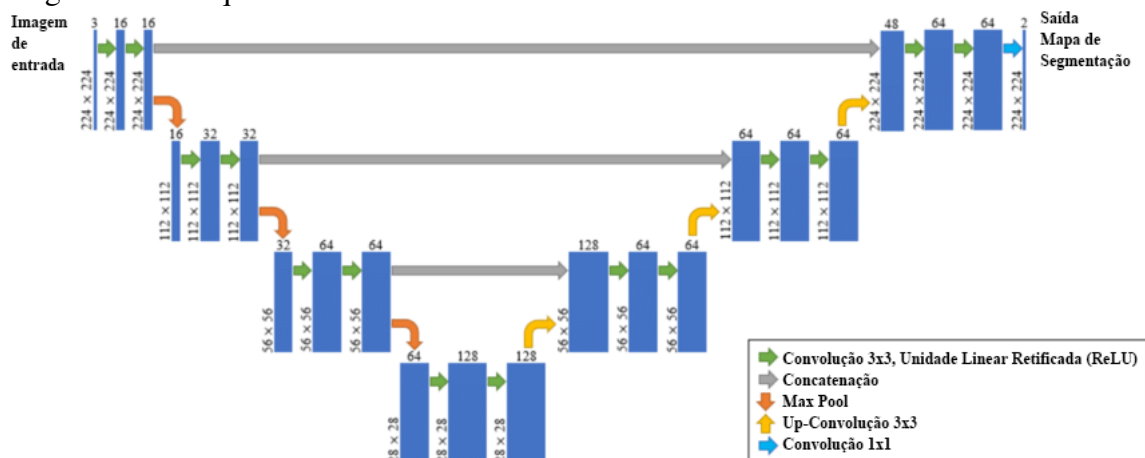
1. Convolutacional 1: 96 filtros com 11x11x3, stride 4;
 2. Convolutacional 2: 256 filtros com 5 x 5;
 3. Convolutacional 3: 384 filtros com 3 x 3;
 4. Convolutacional 4: 384 filtros com 3 x 3;
 5. Convolutacional 5: 256 filtros com 3 x 3.
1. Totalmente Conectado 1: 4096 neurônios;

2. Totalmente Conectado 2: 4096 neurônios;
3. Totalmente Conectado 3: 1000 neurônios.

No final do processo da AlexNet um dos neurônios vai estar mais ativado, possibilitando calcular qual é a maior probabilidade sobre classes de saída previstas para a imagem de entrada.

O processo citado anteriormente se caracteriza para as redes classificadoras, em que a imagem é transformada a um ponto que só resta um vetor de informações para ser usado na classificação. Em redes que tem a função de segmentação, o processo inicial se caracteriza com as redes classificadoras, com a aplicação das convoluções mas quando a imagem chega a uma certa "profundidade", é realizado a desconvolução desses dados, fazendo com que a saída da rede gere um mapa de segmentação com o destaque para as regiões de interesse (do inglês: *region of interest*) (ROI). Para esse tipo de rede, é necessário citar uma rede muito explorada na literatura, como a U-Net, desenvolvida por Ronneberger, Fischer e Brox (2015), conforme na Figura 14 (SULTANA; SUFIAN; DUTTA, 2020).

Figura 14 – Arquitetura da U-Net.



Fonte: adaptação dos autores (VOULODIMOS *et al.*, 2020)

A utilização dessas arquiteturas não depende só de coletar a imagem e passar por essa aprendizagem. As mesmas possuem etapas a serem seguidas como divisão dos dados, aplicações de técnicas de aumento de dados, função de perda, otimização e uma análise de desempenho através de métricas.

2.4.1 *Divisão dos dados*

Através do banco de dados é possível fazer uma seleção dos dados, que podem ser divididos em: conjunto de treino, conjunto de validação e conjunto de teste. Cada um desses subconjuntos tem uma função importante na criação do modelo treinado. É imprescindível que tenham a maior representatividade possível e sejam selecionados de forma arbitrária. Os subconjuntos geralmente são divididos entre 70%, para o conjunto de treino, 15%, para o conjunto de validação e 15%, para o conjunto de teste, mas pode variar dependendo do problema e das características dos dados. Cada conjunto tem informações distintas, por exemplo, uma imagem usada no treinamento não pode ser usada para validação ou teste, pois o modelo treinado já apreendeu com aquela imagem. Esses subconjuntos podem ser caracterizados como:

- **Conjunto de treino:** Dentre os outros conjuntos esse deve ser o maior, pois o modelo vai aprender com os dados e por conta disso deve contemplar boa parte das amostras. É o conjunto de dados que será usado para desenvolver o modelo;
- **Conjunto de validação:** Na etapa de treinamento o modelo vai aprendendo ao longo das épocas. Para cada época é possível retirar um valor de métrica e ao final de cada uma são aplicados os dados do conjunto. É possível utilizar o conjunto de validação para encontrar o modelo com menor erro em determinada época. Com isso, assegurando o melhor modelo treinado para determinada época. Também é através desse conjunto que podemos detectar *overfitting*, que é um problema identificado quando o modelo se torna especializado para dados de treinamento, com isso ele se torna menos capaz de generalizar para novos dados, resultando em um aumento no erro de generalização. Não é só a detecção de *overfitting* que é possível ser analisada com o conjunto de validação, podemos citar também *underfitting*, conjunto de treino ou de validação não representativo (STEVENS; ANTIGA; VIEHMANN, 2020; RUSSELL *et al.*, 2010; BROWNLEE, 2019; CHRISTOPHER *et al.*, 2018).
- **Conjunto de teste:** A base de testes estima a taxa de erro do modelo, também possibilita a visualização do resultado do modelo treinado.

2.4.2 *Aumento de dados para conjunto de treino*

O aumento de dados pode ser feito por diferentes técnicas e é usado para gerar “novas” amostras para o conjunto de treino a partir dos dados originais do conjunto. E ao mesmo

tempo garantindo que os rótulos dos dados não sejam alterados. O objetivo ao aplicar o aumento de dados é aumentar a generalização do modelo treinado. Não é aplicado um aumento de dados para conjunto de validação e teste, simplesmente é usado os dados originais sem modificações. A aplicação do aumento de dados se torna interessante pois a rede é capaz de aprender com recursos mais robustos. O processo de aumento funciona com base em transformações afins a partir da imagem original de forma aleatória, em que uma imagem pode gerar várias diferentes e as imagens com mudanças são aplicadas para o momento de treinamento da rede. Então o conjunto de treino original é só utilizado para gerar as imagens aumentadas, não participando diretamente do treinamento (SHORTEN; KHOSHGOFTAAR, 2019; ROSEBROCK, 2019).

Nesse projeto foram aplicados os seguintes conjuntos de transformações afins:

- Zoom: A imagem sofre uma ampliação;
- Rotação: Isso gira cada imagem até o ângulo especificado;
- Giro horizontal e vertical: Inverte aleatoriamente a imagem de entrada nas direções horizontal ou vertical, respectivamente;
- Deslocamento horizontal ou vertical: Isso resulta no deslocamento da imagem na direção horizontal ou vertical;
- Brilho: Produz imagens semelhantes às tiradas com diferentes condições de iluminação, mudanças exemplificadas na Figura 15 (Foi aplicado apenas no processo de criação dos modelos segmentadores).

Figura 15 – Aplicação da mudança de brilho.



Fonte: (KANG; SINGH, 2019)

2.4.3 Transformação dos dados

As transformações nos dados são comumente aplicadas a fim de ganho de desempenho no processo de acurácia do modelo. Essas transformações podem ser simples, como alteração dos valores de escala, como mudar a escala do RGB de [0-255] para [0,0-1,0], ou mais complexa como a normalização através da média e o desvio padrão dos dados.

$$C_{i,j}^R = \frac{R_{i,j}}{255} \quad (2.1)$$

$$C_{i,j}^G = \frac{G_{i,j}}{255} \quad (2.2)$$

$$C_{i,j}^B = \frac{B_{i,j}}{255} \quad (2.3)$$

$$Z_{i,j}^R = \frac{R_{i,j} - \mu(C_{i,j}^R)}{\sigma(C_{i,j}^R)} \quad (2.4)$$

$$Z_{i,j}^G = \frac{G_{i,j} - \mu(C_{i,j}^G)}{\sigma(C_{i,j}^G)} \quad (2.5)$$

$$Z_{i,j}^B = \frac{B_{i,j} - \mu(C_{i,j}^B)}{\sigma(C_{i,j}^B)} \quad (2.6)$$

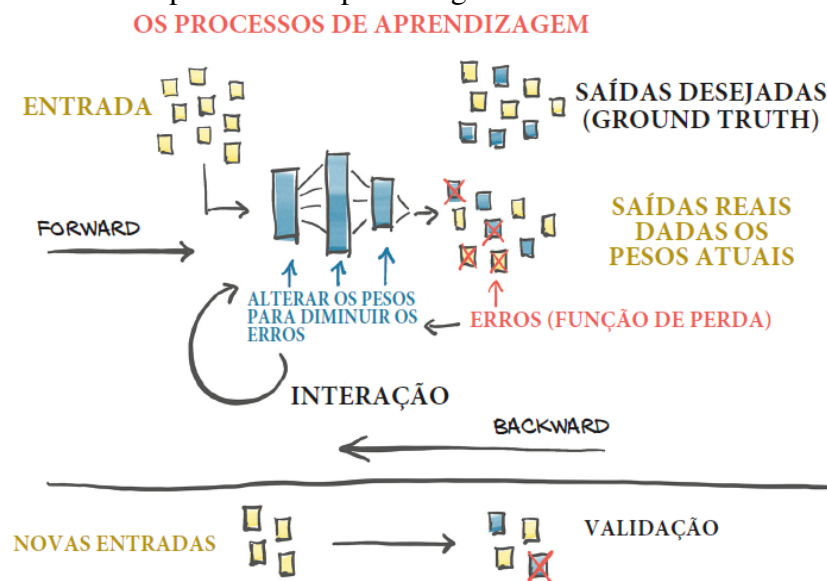
As equações 2.1 a 2.3 são referentes às mudanças de escala para uma determinada localização de *pixel* dos seus canais RGB da imagem, sendo sua posição determinada pelos valores de i, j . O seu resultado gera uma matriz do tamanho original da imagem com três canais, sendo eles $C_{i,j}^R, C_{i,j}^G$ e $C_{i,j}^B$. Já as equações 2.4 a 2.6, tem o papel de normalizar os dados, são mais comumente chamados de *z-scores*, μ retorna o valor médio de todos os *pixels* do canal referentes das equações 2.1 a 2.3 e já σ retorna o desvio-padrão de todos os *pixels* desses mesmos canais referentes. Os dados utilizados para calcular o μ e σ , são retirados do conjunto de treino para as imagens originais sem a aplicação do aumento de dados. Em seguida são aplicadas as equações 2.4 a 2.6 para o conjunto de treino aumentado, o conjunto de validação e conjunto de teste, garantido que os dados estão no mesmo plano.

2.4.4 Modelos

A mecânica de aprendizagem para criação do modelo de treinamento vai além de só ter uma arquitetura adequada para problemática, dados normalizados ou até ter dados em

enormes quantidades para o treinamento. As Secções 2.4.4.1 e 2.4.4.2 estão ligados a etapa de aprendizados dos dados. O processo de aprendizagem do modelo tem diversas etapas e funções, conforme esquematizado na Figura 16. Nesse processo temos os dados de entrada, que são usando para treinamento do modelo, e suas saídas desejadas (*Ground truth*), que são usada para avaliar as saídas reais dadas os pesos atuais. Para avaliar os dados de saídas reais dadas os pesos atuais com a saídas desejadas, é aplicado uma função de perda, que fornece os erros da saída nos dois dados. Em seguida, a retropropagado (*backpropagation*) dos erros e a atribuído ao gradiente para cada parâmetro ao longo do caminho. Para conjunto de função de perda e retropropagado, é possível utilizar um otimizador tem a função de atualizar parâmetros do modelo e minimizar funções de perda (STEVENS; ANTIGA; VIEHMANN, 2020). E ao final desse esquema são atualizados os pesos. Isso é feito até o final da interação. Depois é avaliado o modelo treinado no processo de validação com novos dados de entradas.

Figura 16 – Modelo processo de aprendizagem.



Fonte: adaptação (STEVENS; ANTIGA; VIEHMANN, 2020)[pág. 142]

2.4.4.1 Função de perda

Escolher a função de perda é uma etapa complexa, pois cada uma pode gerar uma melhoria para o treinamento ou espaço computacional entre outros fatores que ligados direta ou indiretamente aos dados. A função que foi utilizada para processo de segmentação e classificação foi a perda de entropia cruzada (do inglês: *cross-entropy loss*) (CCE), representada na equação 2.7. A CCE, que é a combinação de *LogSoftmax* e *NLLLoss* (STEVENS; ANTIGA; VIEHMANN, 2020; HO; WOOKEY, 2020; NATHANIEL, 2021).

$$CCE = -\frac{1}{N} \sum_{i=0}^N \sum_{j=0}^J y_j \cdot \log(\hat{y}_j) + (1 - y_j) \cdot \log(1 - \hat{y}_j) \quad (2.7)$$

A equação 2.7 tem y_j , que é correspondente à sua verdade absoluta, rótulo de classe ou regiões demarcadas por diferentes cores e o \hat{y}_j é a previsão correspondente dada pelo modelo treinado.

2.4.4.2 Otimização

Na aplicação do método de otimização foram escolhidos dois diferentes algoritmos, um para segmentação e outro para a classificação. Para a segmentação foi escolhido o algoritmo AdamW, dos autores Loshchilov e Hutter (2019), que é uma variação do algoritmo Adam proposto pelos autores Kingma e Ba (2017). Segundos os autores Loshchilov e Hutter (2019), o desempenho de generalização é substancialmente melhor do que a implementação comum de Adam.

Já para a classificação foi usado o algoritmo descida gradiente estocástica (do inglês: *gradient descent method*) (SGB), que para os autores Zhang, Wang e Zhang (2019) desenvolveram um aplicação de reconhecimento de cor de veículos. Tal recurso permitia uma convergência mais rápida do modelo (SUTSKEVER *et al.*, 2013).

2.4.5 Métricas

As aplicações das métricas de avaliação são utilizadas para medir a qualidade ou precisão do modelo já treinado, com a aplicação do conjunto de teste. Cada uma pode ser medida diferenciada e aplicada em diferentes situações, nesse caso tem métricas para avaliar de forma diferenciada a segmentação e a classificação.

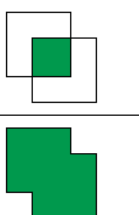
2.4.5.1 Segmentação

Alguns métodos bem explorados na literatura são: a Acurácia de *pixel*, que analisa *pixel a pixel* da imagem resultante do modelo e da imagem de rótulo. Não é um bom método a se aplicar em segmentação binárias pois pode dar uma falsa sensação de que o modelo está com precisão alta mas quando é analisado a imagem resultante só está presente uma classe; Coeficiente Sørensen-Dice (**Dice**), que tem sua própria fórmula para calcular a relação da imagem predita com a imagem rotulada, o coeficiente é mais criterioso do que a acurácia de *pixel*; o Jaccard *index* tem uma fórmula similar ao score de Dice mas tem resultados distintos e também é mais criterioso do que a acurácia de *pixel*. Para esse projeto foi escolhido o Jaccard *index*, sua metodologia é explicada na próxima subseção (Gu *et al.*, 2019; Nguyen *et al.*, 2019).

2.4.5.1.1 Jaccard *index*

Também conhecido como Intersecção sobre União (do inglês: *intersection over Union*) (IoU) ou coeficiente de similaridade de Jaccard, é um método estatístico que mede a similaridade entre dois conjuntos de amostras. Na Figura 17, tem-se a fórmula simplificada que é aplicada nesse método, e que se apresenta como é uma simples razão. No numerador é calculado a área de sobreposição entre a imagem rotulada (*ground truth*), estrutura feita manualmente, com a imagem predita, estrutura descoberta pela modelo treinado. No denominador é a área de união, essa área contém a junção da região da imagem rotulada e predita. Então quanto mais próximo tiverem essas regiões melhor será a segmentação. Esse tipo de métrica se popularizou para avaliar a acurácias das redes neurais convolucionais. Na Figura 18, sendo a pessoa como a região de interesse na imagem, tem-se os resultados aplicado em uma imagem, para numerador (área de sobreposição) e denominador (área de união) da fórmula, conforme na imagem da Figura 17.

Figura 17 – Fórmula do Jaccard *index*.

$$\text{IoU} = \frac{\text{área de sobreposição}}{\text{área de união}}$$


Fonte: adaptação dos autores (ROSEBROCK, 2020)

Figura 18 – Aplicação da fórmula Jaccard *index*.

(a) Área de sobreposição.



(b) Área de união.



Fonte: adaptação dos autores (KIZRAK, 2019).

2.4.5.2 Classificação

É possível obter valores estatísticos dos resultados da aplicação do modelo treinado em conjunto de teste, tais como: Acurácia, Precisão, *Recall*, *F1-Score* e outros. Cada uma tem sua particularidade. Para calcular essas métricas é possível usar a definição de matriz de confusão, que indica os acertos e os erros de um modelo, comparando assim os valores com o seu valor real. Na Tabela 2, representa a matriz de classificação.

Matriz de classificação é tem dois grupos: Valor Previsto e Valor verdadeiro. O valor previsto é o resultado, que é comparado com o valor verdadeiro, assim agregando a matriz.

Tabela 2 – Matriz de classificação.

		(Valor Verdadeiro)	
		Positivo	Negativo
(Valor Previsto)	Positivo	Verdeiros Positivos (VP)	Falsos Positivos (FP)
	Negativo	Falsos Negativo (FN)	Verdadeiros Negativos (VN)

Fonte: o autor.

$$Acurácia = \frac{Verdadeiros\ Positivos\ (VP) + Verdadeiros\ Negativos\ (VN)}{Total} \quad (2.8)$$

- Acurácia: apresenta uma precisão geral do classificador, seu valor é obtido através da razão do número de acertos de um conjunto de amostras por sua quantidade de dados que foram classificados, demonstrado na Equação 2.8. O Total é a quantidade total dos dados, $Total = Positivos\ (VP) + Verdadeiros\ Negativos\ (VN) + Falsos\ Positivos\ (FP) + Falsos\ Negativos\ (FN)$.

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos (VP)}}{\text{Verdadeiros Positivo (VP)} + \text{Falso Positivo (FP)}} \quad (2.9)$$

- *Precisão*: apresenta dentre a classificação que o modelo fez, para um tipo de classe quantas realmente estão corretas, demonstrado na Equação 2.9.

$$\text{Recall} = \frac{\text{Verdadeiros Positivos (TP)}}{\text{Verdadeiros Positivo (VP)} + \text{Falso Negativos (FP)}} \quad (2.10)$$

- *Recall*: apresenta dentre as situações possíveis de classificação de uma determinada classe como valor esperado, quantas foram definidas corretamente, demonstrado na Equação 2.10.

$$F1\text{-Score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (2.11)$$

- *F1-Score*: é uma média harmônica que envolve a precisão e o *recall*, demonstrado na Equação 2.11.

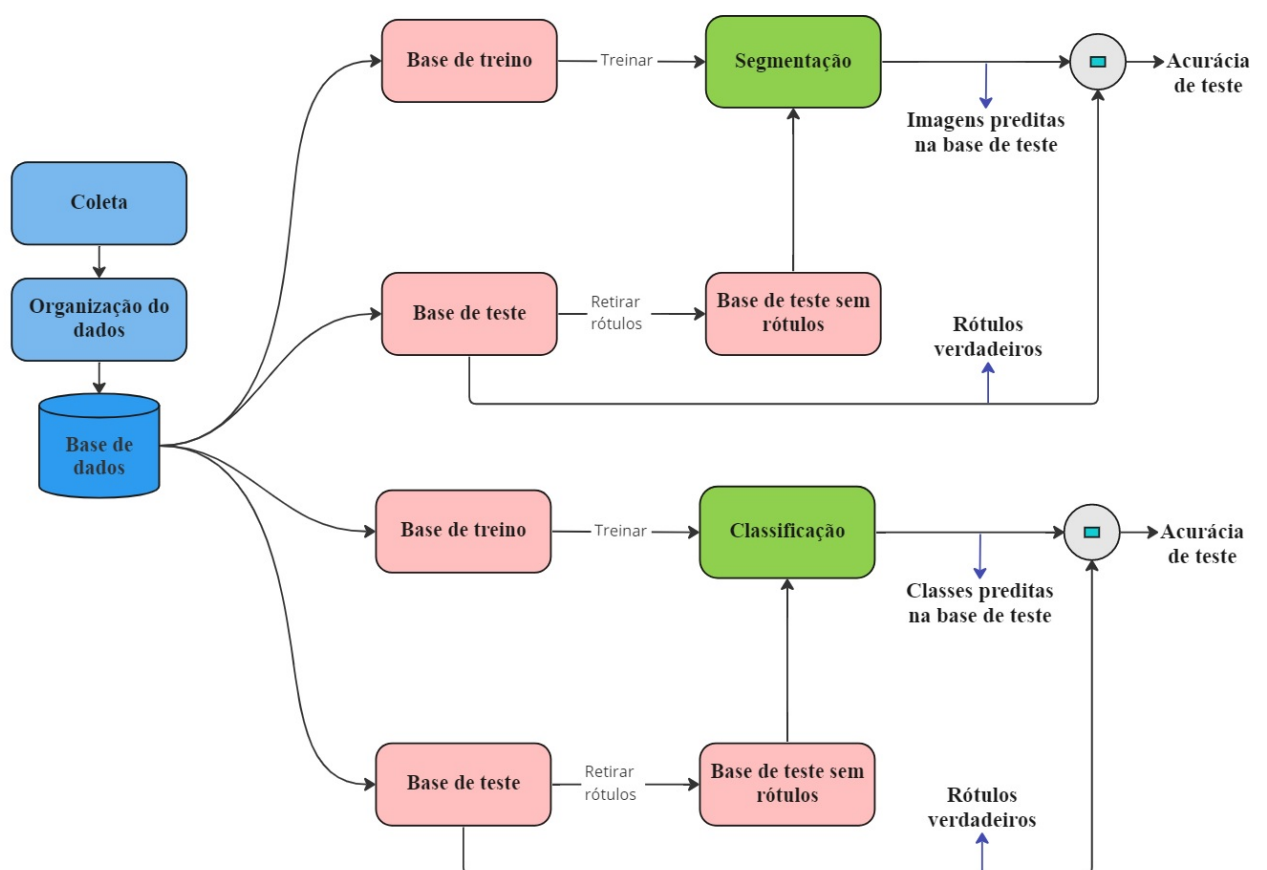
3 METODOLOGIA

O projeto consiste em três partes fundamentais, são elas:

- **Coleta de dados:** É uma estrutura essencial pois consiste na criação da base de dados de imagens. É a partir dela, que pode ser construído um sistema de segmentação e classificação, através dos dados que é possível entender o formato da imagem, o seu padrão entre outros pontos, até chegar ao sistema final. Com a precisão do teste laboratorial hematócrito é possível conhecer o estado real de anemia do animal presente nas imagens coletadas;
- **Segmentação:** Nas imagens é possível encontrar regiões desnecessárias, tais como: os dedos do tratador e o pelo do animal. São regiões que elevam a complexidade da segmentação, pois os dedos do tratador podem estar em algumas imagens com luva, ou sem e ter a mesma tonalidade da mucosa do olho do animal. Além disso os animais tem as pelagens diferentes. A região de interesse pode estar em diferentes locais da imagem, pode estar com diferentes tamanho, entre outros fatores, elevando ainda mais a complexidade da segmentação. Por essas problemáticas na segmentação, optamos por explorar quatro modelos de aprendizagem profunda (*Deep Learning*) para realizar tal função. Nesse caso, os modelos aplicados foram: ERFNet, U-Net, FCN e DeepLabV3, e para esses modelos foi adotada uma rotina buscando o melhor resultado para a base de dados criada. Também utilizar os melhores modelos treinados, aplicando como entrada imagens outra base de dados, afim de obter uma nova validação. Além disso, explorar o seu comportamento em uma aplicação *mobile*;
- **Classificação:** Para o reconhecimento da palidez da mucosa e sua classificação foi admitido também o uso de uma aprendizagem profunda. Nesse caso, os modelos aplicados foram: AlexNet, MobileNetV2, ResNet18, ResNet152 e ResNeXt. A dificuldade maior dessa classificação é encontrar uma configuração ou um próprio modelo que possam fazer uso do melhor dos dados para sua a classificação. Também testar os melhores modelos treinados outra base de dados. Também utilizar os melhores modelos treinados, aplicando como entrada imagens outra base de dados, afim de obter uma nova validação. Além disso, explorar o seu comportamento em uma aplicação *mobile*.

A Figura 19 representa o fluxo da metodologia, esquematizado respectivamente, como: coleta, que representa a obtenção dos dados brutos; Organização dos dados, que é ordenação dos dados de uma forma que possa criar um banco de dados estruturado; Banco de dados, que é possível ser usado no processo de criação dos modelos treinados de segmentação e classificação. O processo de treinamento e validação da segmentação e classificação são semelhantes, são eles: divisão da base de treino e a base de teste; Uso da base de treino para criação do modelo de inteligência; Utilização da base de teste sem rótulos para criação dos resultados preditos; Validação dos resultados preditos na base de teste com os rótulos verdadeiros, assim resultando em uma acurácia de teste. A diferença do processo de segmentação e classificação são os resultados preditos, um é uma imagem com a região de interesse e outro a classe predita da imagem.

Figura 19 – Fluxo da metodologia.

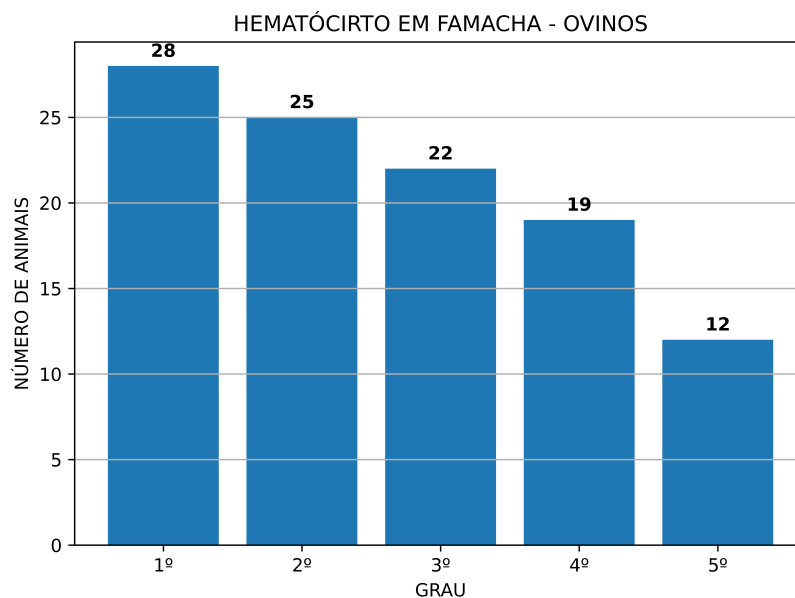


Fonte: criado pelo autor.

4 APLICAÇÃO E RESULTADOS EXPERIMENTAIS

O gráfico da Figura 11 nos permite observar um alto número de animais com grau 1 e 2, isso é algo que pode trazer complicações futuras nos testes, acarretando resultados tendenciosos para a classe com maior quantidade de dados. Para contornar esse desbalanceamento nos dados, optamos por usar apenas uma fração dos mesmos na realização dos experimentos, como apresentado no gráfico da Figura 20. Também nesse experimento, admitiu-se o uso das imagens coletadas pelos dispositivos *mobiles*, sem nenhuma alteração prévia nas imagens, usando a mesma estrutura de cenário da Figura 5c.

Figura 20 – Gráfico dos dados organizados para os experimentos.



Fonte: o autor.

Para termos resultados mais precisos sobre a estabilidade do algoritmo de segmentação e de classificação, adotamos uma rotina diversificada para criação do modelo já treinado. Assim, adotando uma estrutura de repetição:

- Seleccionava-se 1 imagem para teste;
- Usava-se o restante das imagens, 105 imagens, para separar os dados em conjunto de treino e de conjunto de validação, isso de forma aleatória.

Esse método de validação é chamado de *k-fold*, que pode ser interpretado como uma mistura do *leave-one-out* com o *leave-p-out*, sendo que cada imagem é usada em um determinado momento (LIU, 2019; BATTULA, 2021). Essa estratégia foi utilizada para 106 imagens, gerando 106 modelos já treinados. Porém, é possível usar todos esses modelos, então foi utilizado um

valor estatístico, apresentado na seção 2.4.5.1.1, extraído da imagem de teste, para selecionar apenas um modelo com a melhor estrutura, isso para o processo de segmentação.

Na classificação, foram adotados os rótulos como Anêmico (Graus 3 a 5) e Não anêmico (Graus 1 e 2), assumindo como base para conseguir o melhor resultado com os poucos dados, como no trabalho dos autores Demoliner e Alves (2017) que apresentaram análise relevante com essa divisão. Então, para encontrar o melhor modelo dos 106, foi utilizada a seguinte estratégia: foram usados apenas os testes que deram resposta correta. Dentre os mesmo, elegemos apenas os que acertaram mais imagens do conjunto de validação, e em caso de empate analisando o menor valor de perda para aquele determinado momento de melhor validação.

Essas estratégias para a segmentação e classificação são uma forma de analisar a metodologia. Isso porque algumas imagens podem ser mais bem utilizadas no treinamento do que na validação, por conta da sua qualidade, posição, cenário, entre outros pontos que interferem na qualidade das imagens.

Um dos métodos utilizados para a redução do tempo de processamento e custo computacional é o redimensionamento do tamanho das imagens de entrada para a rede convolucional. No caso da segmentação, as imagens originais foram reduzidas para um tamanho de 224x224 *pixels*, assim passando por todo o processo de treinamento e validação utilizando esse tamanho. E no processo de teste, as imagens de saída são redimensionadas para o tamanho original da imagem e, em seguida, é realizado o processo de análise da segmentação, com método de similaridade de Jaccard, apresentado na seção 2.4.5.1.1. Essa restauração final do tamanho é um processo para fazer a segmentação original dos *pixels*, assim resgatando os *pixels* sem nenhuma alteração.

Já na classificação, todas as segmentações tinham escalas distintas, umas maiores e outras menores, então foi adotado o critério em que todas as imagens seriam redimensionadas para o tamanho de 224x224 *pixels*.

O ambiente utilizado para execução dos algoritmos aprendizagem profunda foi em uma CPU, com processador Intel(R) Xeon(R) @2.20GHz, já para a Unidade Central de Gráfico (do inglês: *Graphics Processing Unit*) (GPU) foi uma placa gráfica Tesla V100-SXM2-16GB.

4.1 Segmentação

A separação dos dados para geração dos modelos na segmentação foi:

- **Conjunto de treino:** 84 imagens, mas foi aplicado uma estrutura de aumento de dados,

passando para 672 imagens;

- **Conjunto de validação:** 21 imagens;
- **Conjunto de teste:** 1 imagem.

4.1.1 Modelos

Os algoritmos aplicados em aprendizagem profunda podem ser utilizados na segmentação semântica, detecção de objeto ou segmentação de instância. Estes têm sua função de localização do objeto ou regiões de interesses a fim de destacar esses pontos do restante da imagem. Nesse projeto foi selecionado alguns desses algoritmos para uma criação do modelo.

4.1.1.1 *Efficient Residual Factorized Network (ERFNet)*

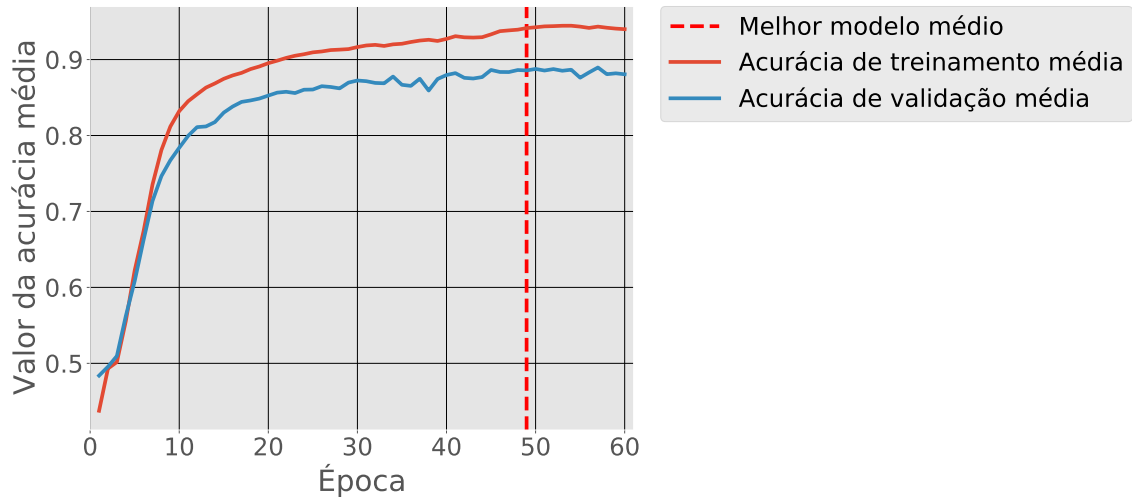
A realização da segmentação foi utilizando o modelo proposto pelos autores Romera *et al.* (2018), o algoritmo utilizado está disponível em Eromera (2019).

Analisando os gráficos das Figuras 21 e 22, foi possível chegar à conclusão a respeito da estabilidade média dos modelos. Não importando a sua estrutura aleatória de treinamento e validação, em determinado momento ele ainda consegue uma estrutura de 88,95% sem normalização e 89,22% com normalização de segmentação para as acurácias médias, no método de similaridade de Jaccard. Essa estabilidade também se repete nas perdas médias em que os valores se mantêm inferiores a 0,1.

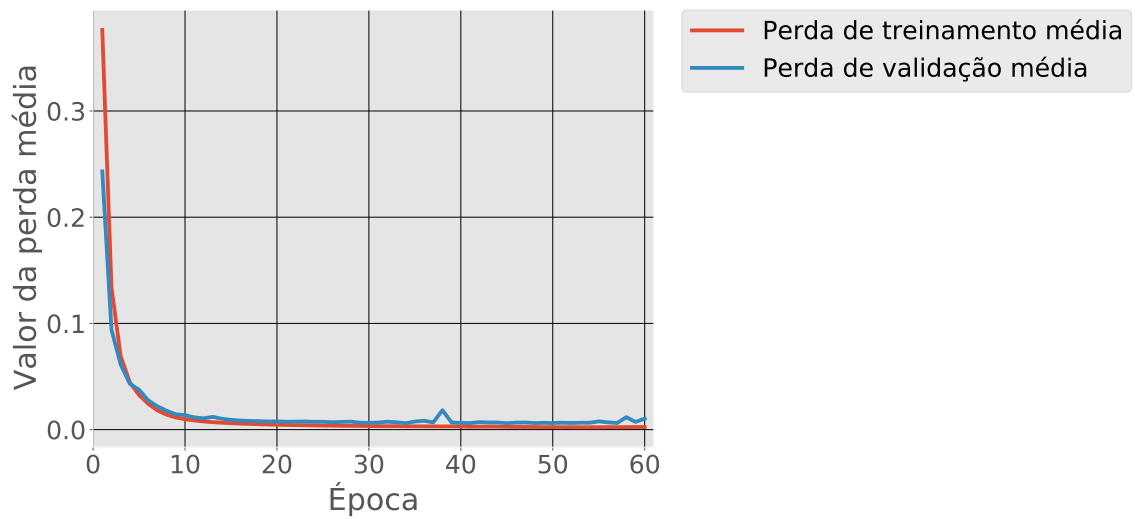
O melhor modelo obteve um valor de segmentação, de 96,04% sem normalização e 97,08% com normalização para a imagem de teste, no critério de similaridade de Jaccard. Os gráficos da Figura 23 e 24 foram obtidos do seu treinamento e validação, com o valor do seu melhor modelo de 90,35% sem normalização e 90,95% com normalização de segmentação para as acurácias média, no método de similaridade de Jaccard. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,1034 segundos sem normalização e 0,1152 segundos com normalização, já para a GPU foi de 0,0210 segundos sem normalização e 0,0234 segundos com normalização.

Figura 21 – Gráficos das acurácias e perdas médias obtidas nos modelos ERFNet, sem normalização.

(a) Acurácias médias.



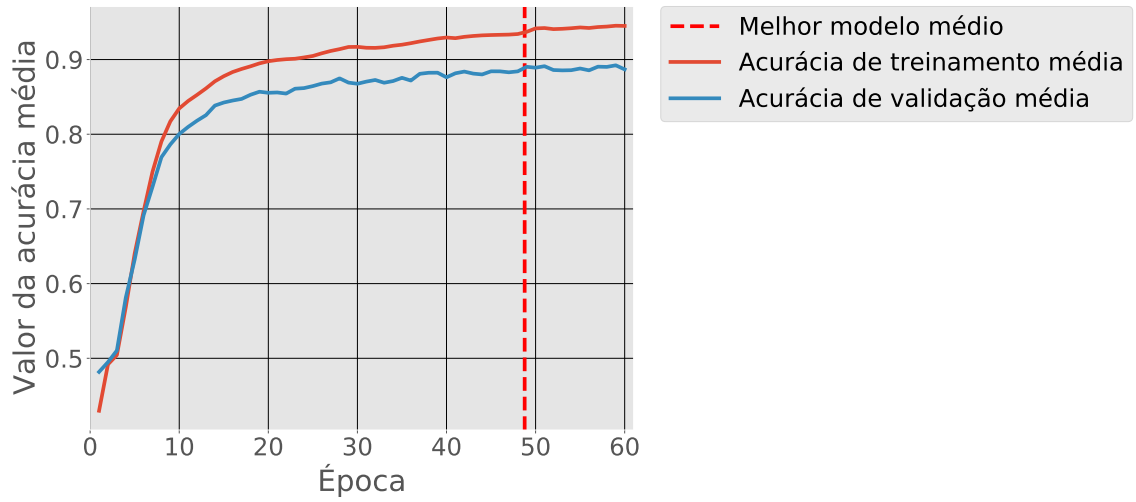
(b) Perdas médias.



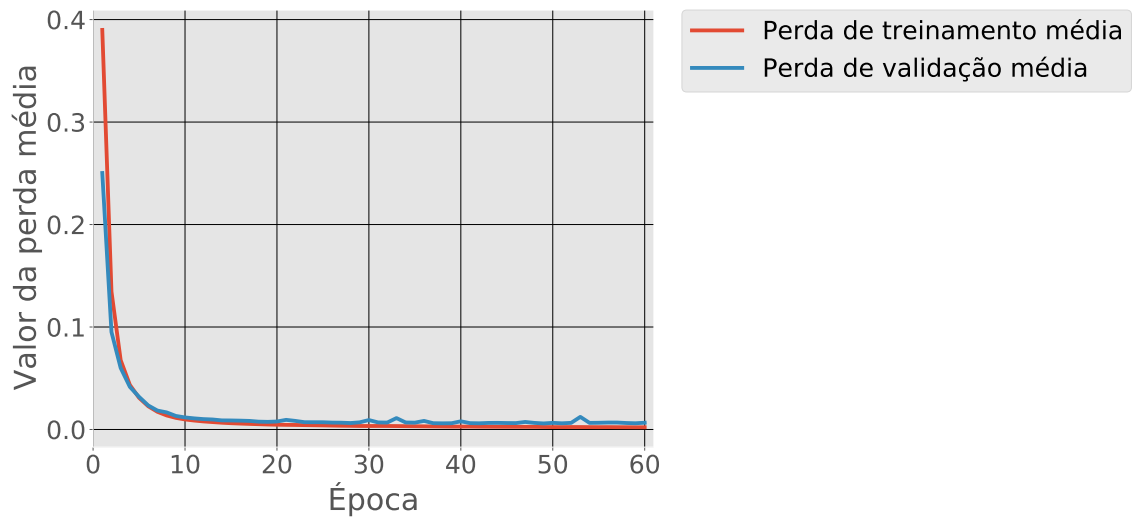
Fonte: o autor.

Figura 22 – Gráficos das acurácias e perdas médias obtidas nos modelos ERFNet, com normalização.

(a) Acurácias médias.

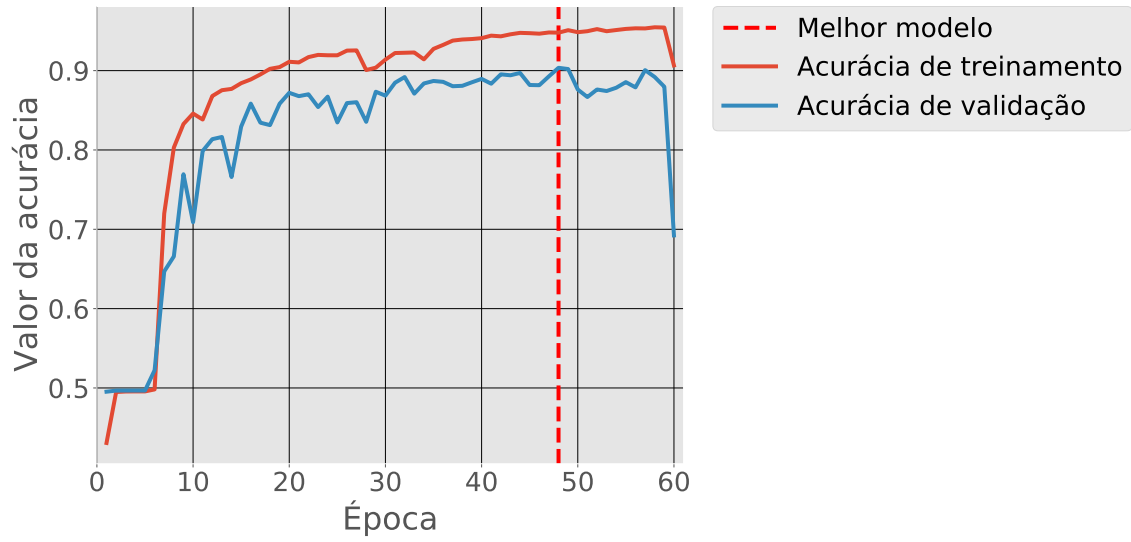


(b) Perdas médias.

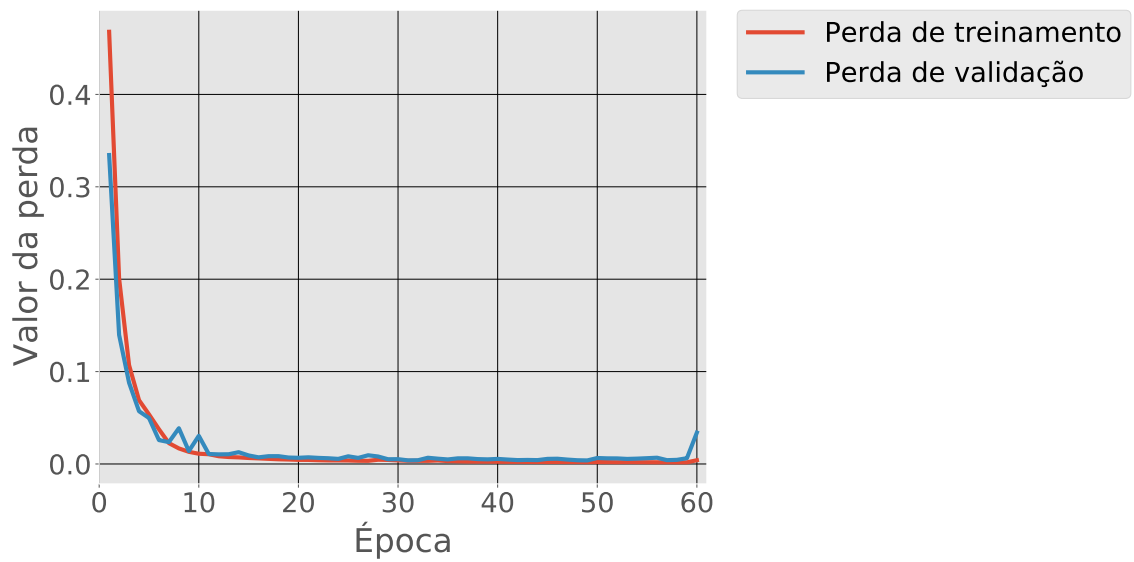


Fonte: o autor.

Figura 23 – Gráfico da acurácia e perda obtido pelo melhor modelo ERFNet, sem normalização.
(a) Acurácias.



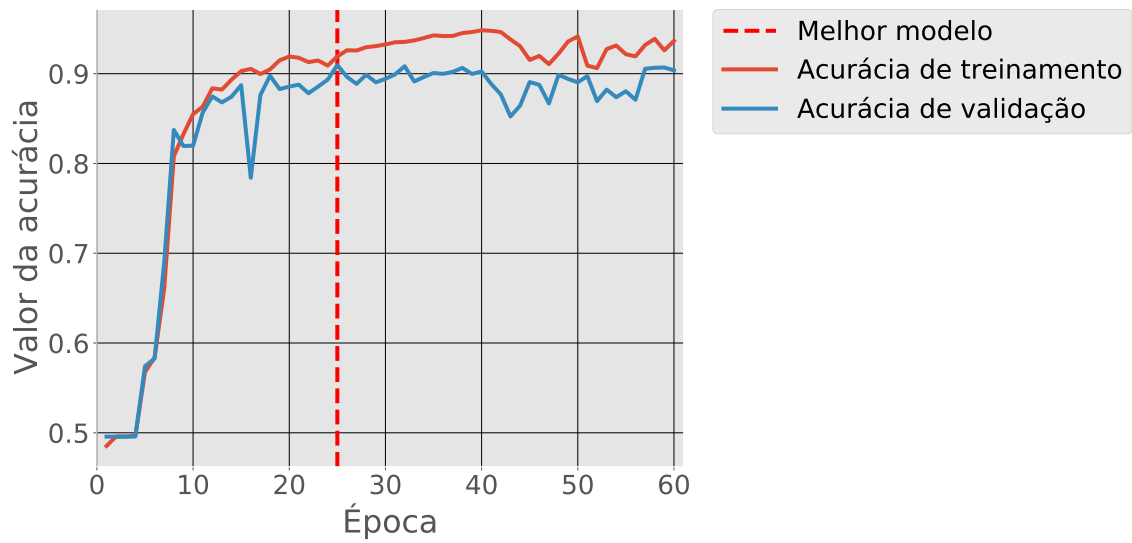
(b) Perdas.



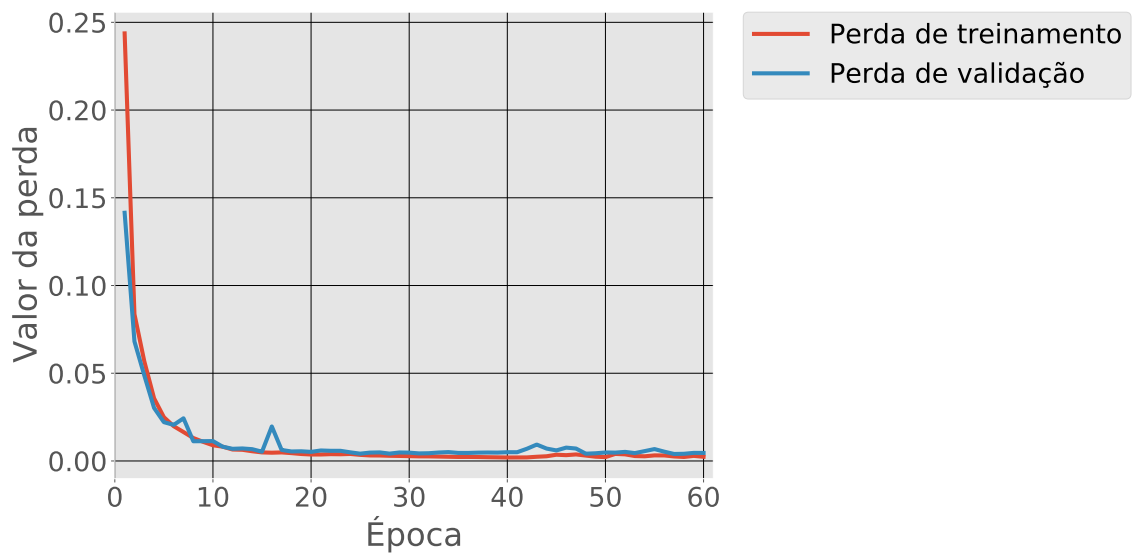
Fonte: o autor.

Figura 24 – Gráfico da acurácia e perda obtido pelo melhor modelo ERFNet, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.1.1.2 U-Net

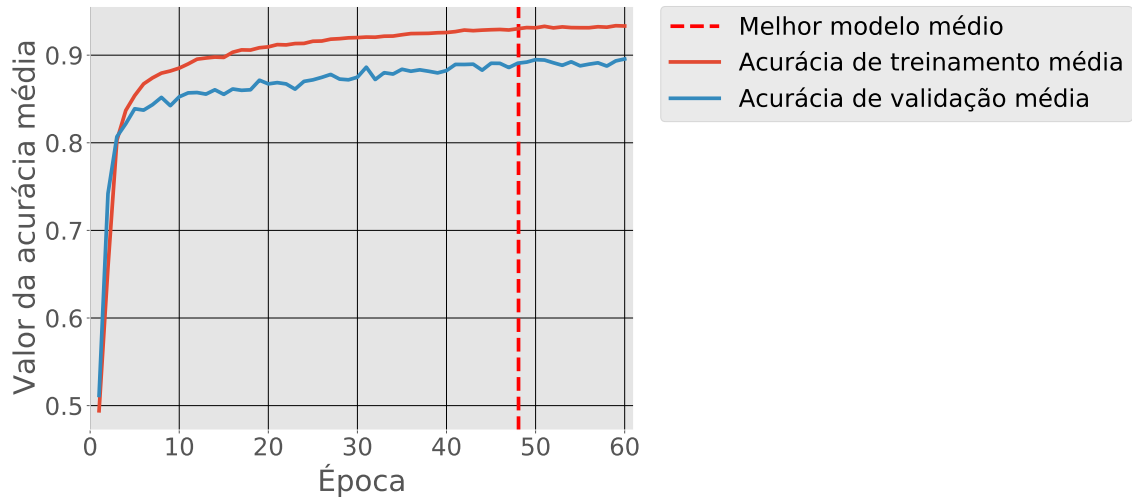
A realização da segmentação foi utilizando o modelo proposto pelos autores Ronneberger, Fischer e Brox (2015), o algoritmo utilizado está disponível em Alexandre (2017).

Analisando os gráficos das Figuras 25 e 26, foi possível chegar à conclusão a respeito da estabilidade média dos modelos. Não importando a sua estrutura aleatória de treinamento e validação, em determinado momento ele ainda consegue uma estrutura de 89,56% sem normalização e 89,79% com normalização de segmentação para as acurácias médias, no método de similaridade de Jaccard. Essa estabilidade também se repete nas perdas médias em que os valores se mantêm inferiores a 0,05.

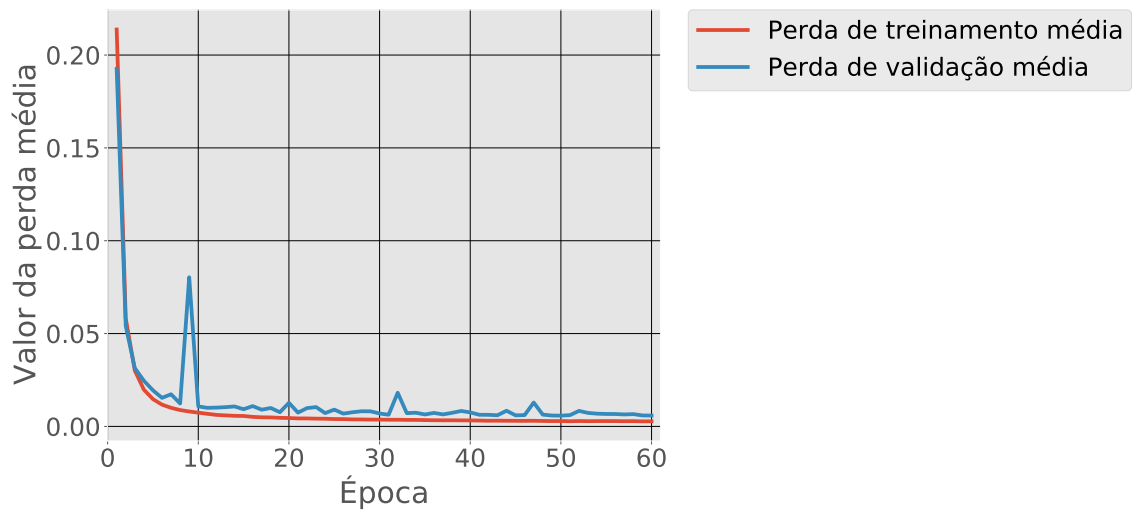
O melhor modelo obteve um valor de segmentação, de 97,20% sem normalização e 97,29% com normalização para a imagem de teste, no critério de similaridade de Jaccard. Os gráficos da Figura 27 e 28 foram obtidos do seu treinamento e validação, com o valor do seu melhor modelo de 92,07% sem normalização e 91,96% com normalização de segmentação para as acurácias média, no método de similaridade de Jaccard. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,5477 segundos sem normalização e 0,4433 segundos com normalização, já para a GPU foi de 0,0075 segundos sem normalização e 0,0074 segundos com normalização.

Figura 25 – Gráficos das acurácias e perdas médias obtidas nos modelos U-Net, sem normalização.

(a) Acurácias médias.



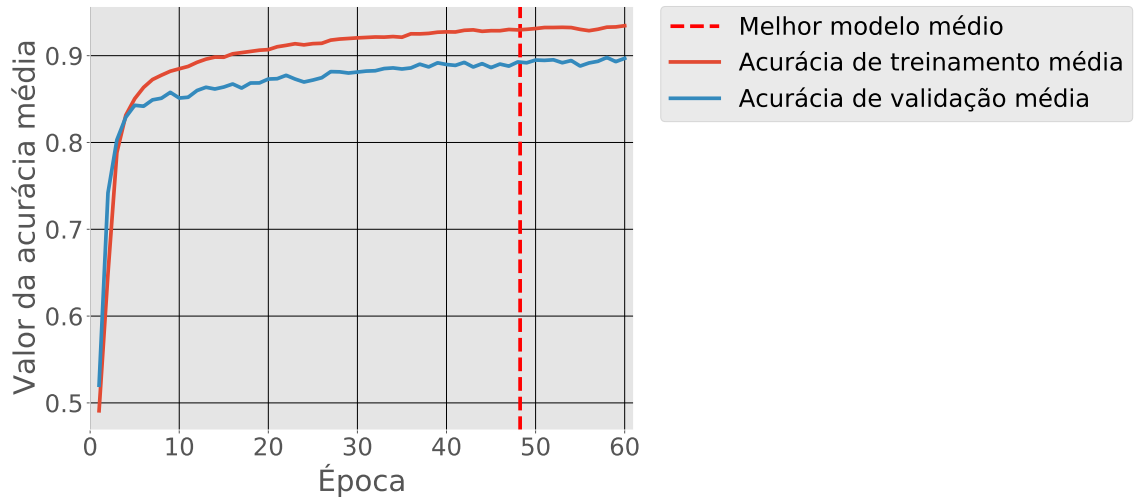
(b) Perdas médias.



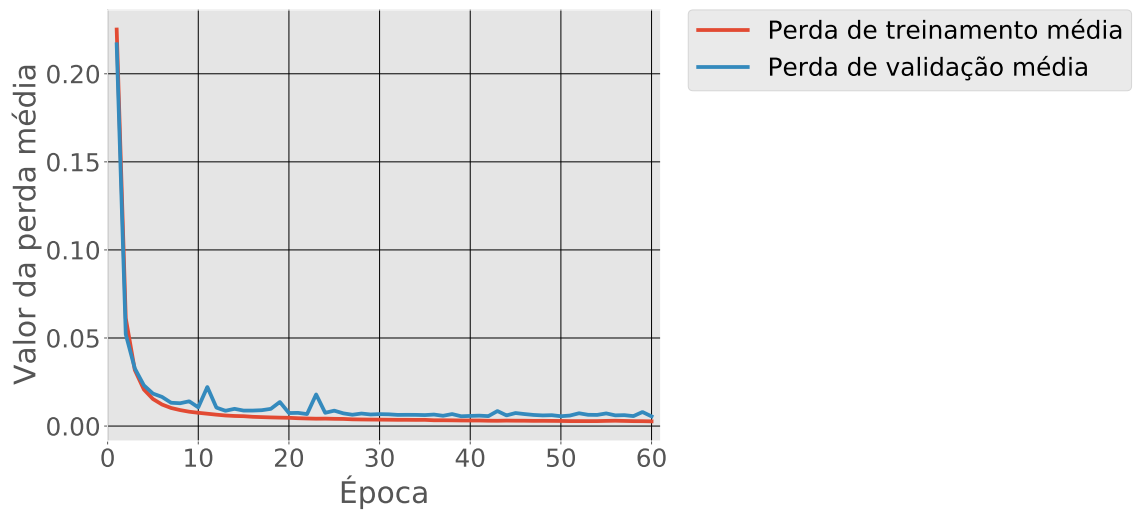
Fonte: o autor.

Figura 26 – Gráficos das acurácias e perdas médias obtidas nos modelos U-Net, com normalização.

(a) Acurácias médias.



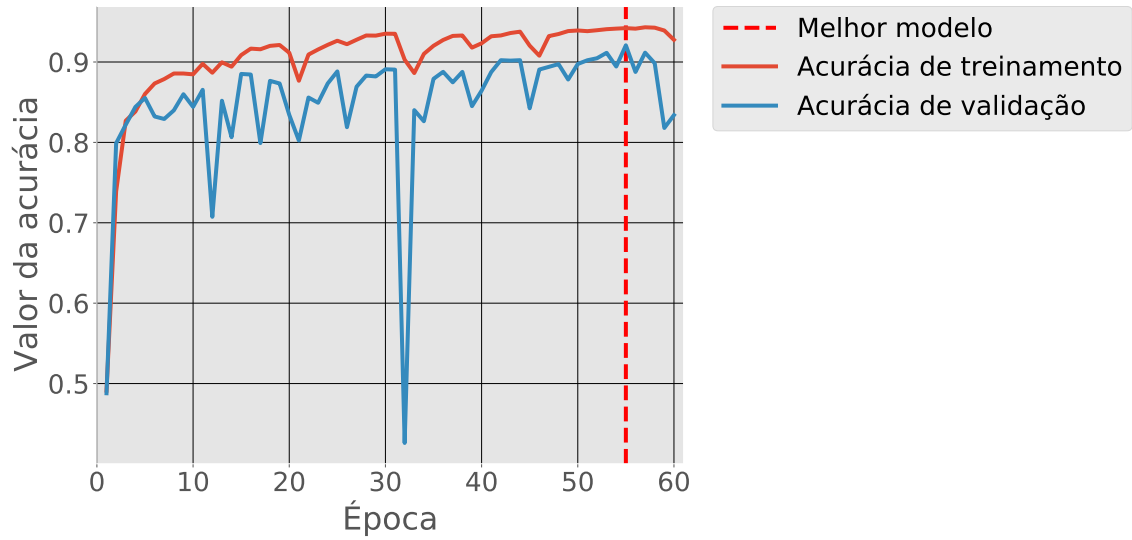
(b) Perdas médias.



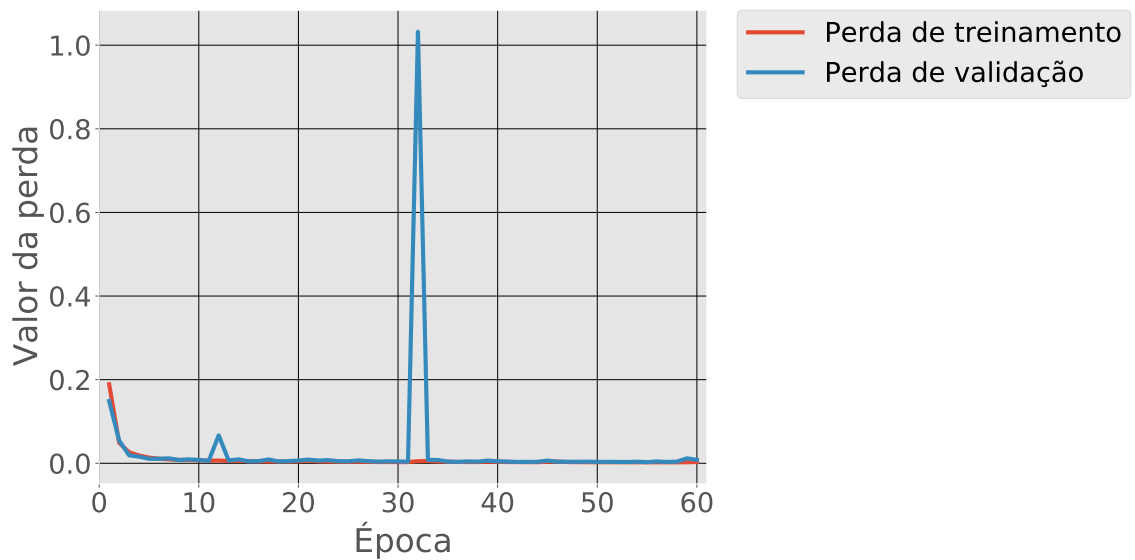
Fonte: o autor.

Figura 27 – Gráfico da acurácia e perda obtido pelo melhor modelo U-Net, sem normalização.

(a) Acurácias.



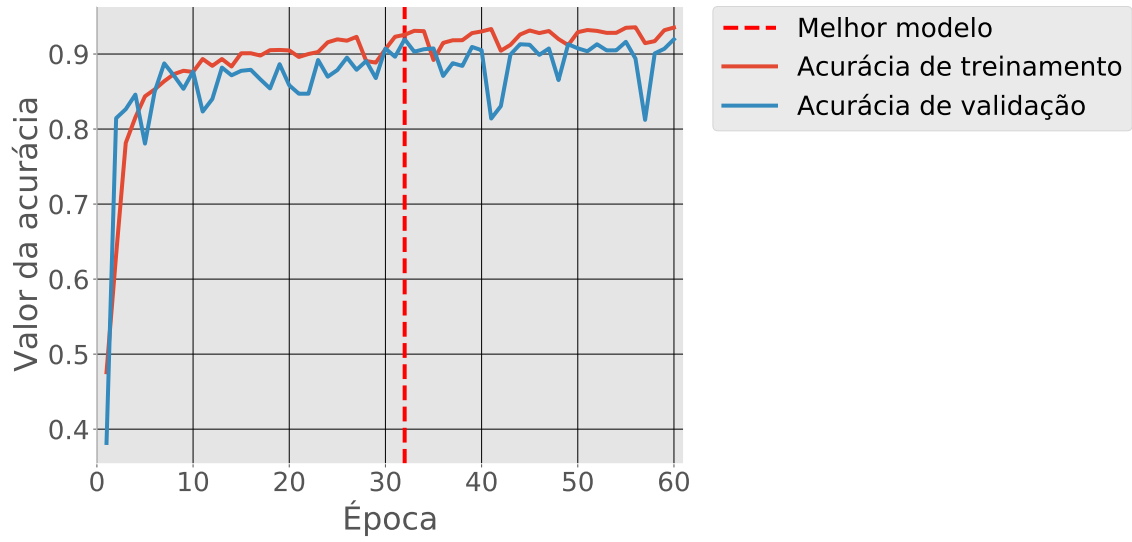
(b) Perdas.



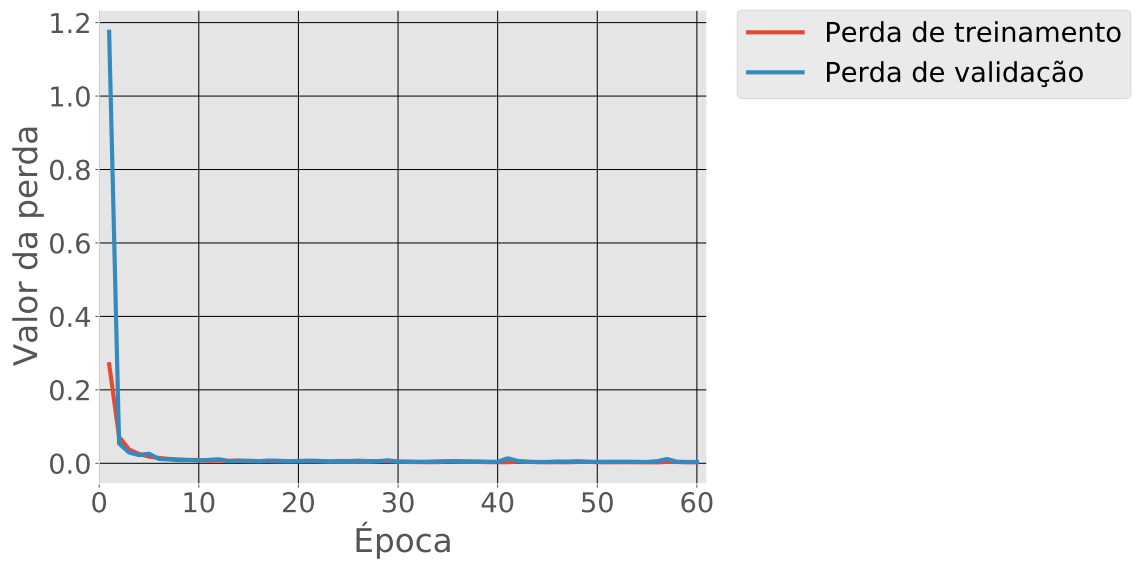
Fonte: o autor.

Figura 28 – Gráfico da acurácia e perda obtido pelo melhor modelo U-Net, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.1.1.3 Fully Convolutional Networks (FCN)

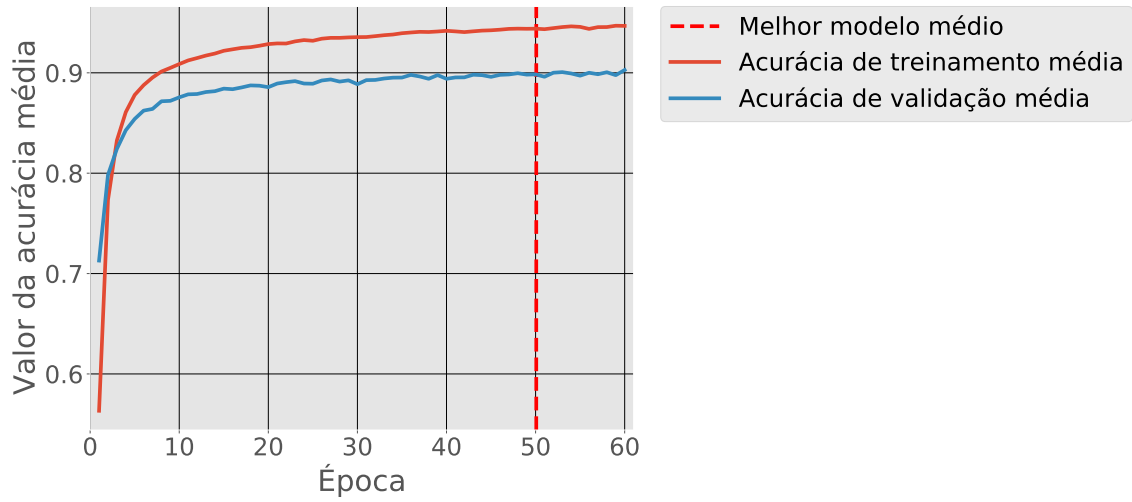
A realização da segmentação foi utilizando o modelo proposto pelos autores Shelhamer, Long e Darrell (2016), também conhecida como *Fully Convolutional Network* (FCN). Mas foi aplicada uma modificação no modelo de classificação utilizada por eles. Aqui assumiu-se o modelo proposto pelos autores He *et al.* (2016), usando a ResNet-50, o algoritmo utilizado está disponível em Pytorch (2019f).

Analisando os gráficos das Figuras 29 e 30, é possível chegar à conclusão a respeito da estabilidade média dos modelos. Não importando a sua estrutura aleatória de treinamento e validação, em determinado momento ele ainda consegue uma estrutura de 90,28% sem normalização e 89,70% com normalização de segmentação para as acurácias médias, no método de similaridade de Jaccard. Essa estabilidade também se repete nas perdas médias em que os valores se mantêm inferiores a 0,025.

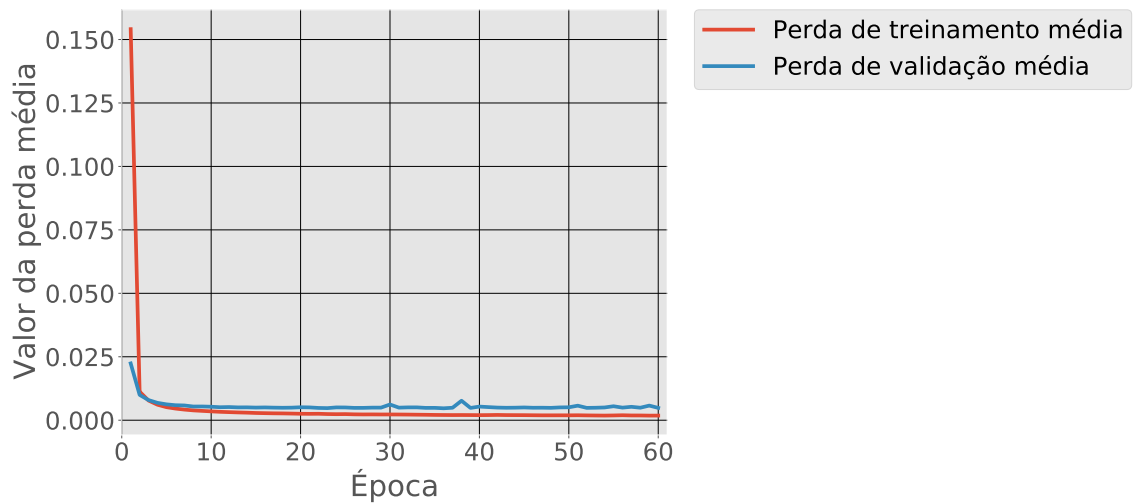
O melhor modelo obteve um valor de segmentação, de 96,65% sem normalização e 90,90% com normalização para a imagem de teste, no critério de similaridade de Jaccard. Os gráficos da Figura 31 e 32 foram obtidos do seu treinamento e validação, com o valor do seu melhor modelo de 92,38% sem normalização e 89,98% com normalização de segmentação para a acurácias média, no método de similaridade de Jaccard. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,4461 segundos sem normalização e 0,4549 segundos com normalização, já para a GPU foi de 0,0116 segundos sem normalização e 0,0132 segundos com normalização.

Figura 29 – Gráficos das acurácias e perdas médias obtidas nos modelos FCN, sem normalização.

(a) Acurácias médias.



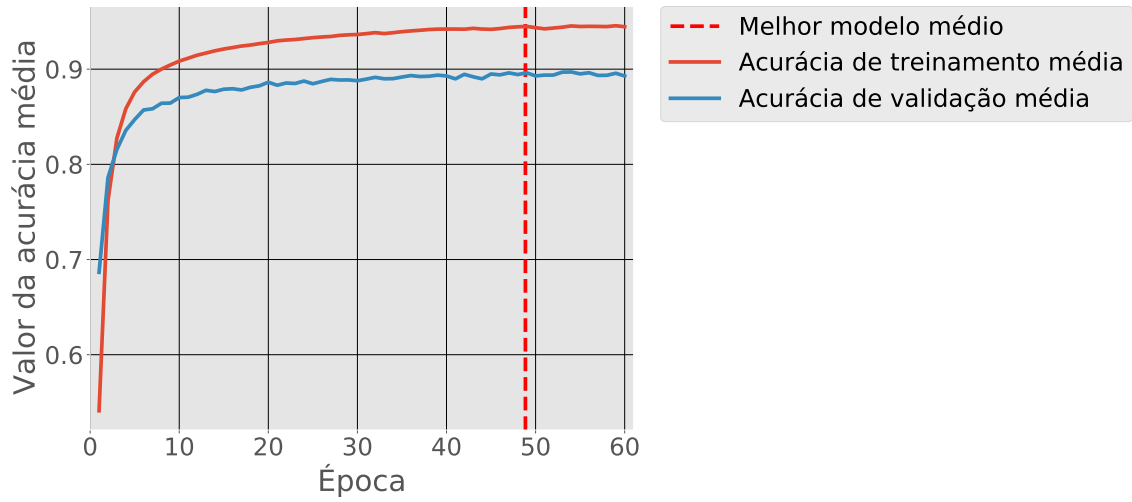
(b) Perdas médias.



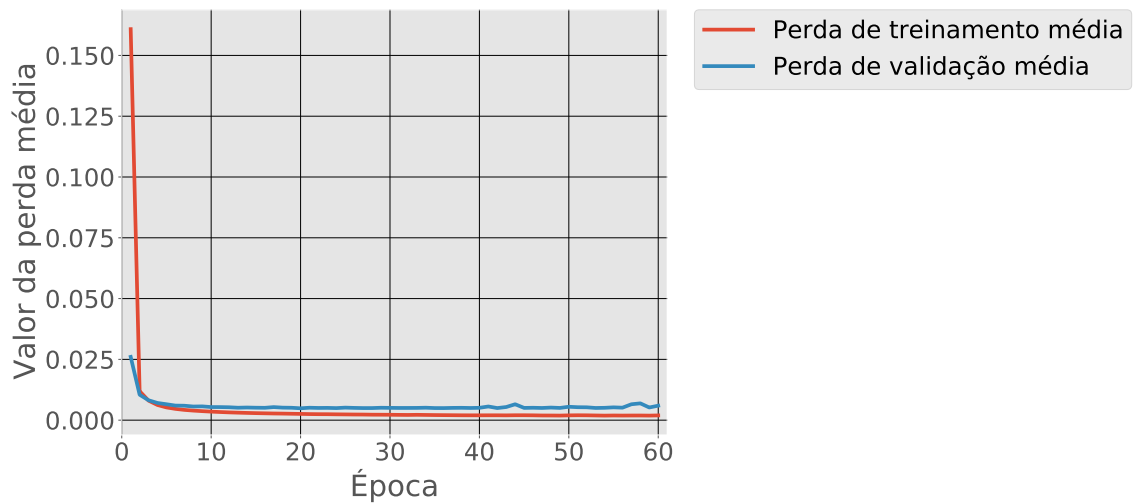
Fonte: o autor.

Figura 30 – Gráficos das acurácias e perdas médias obtidas nos modelos FCN, com normalização.

(a) Acurácias médias.



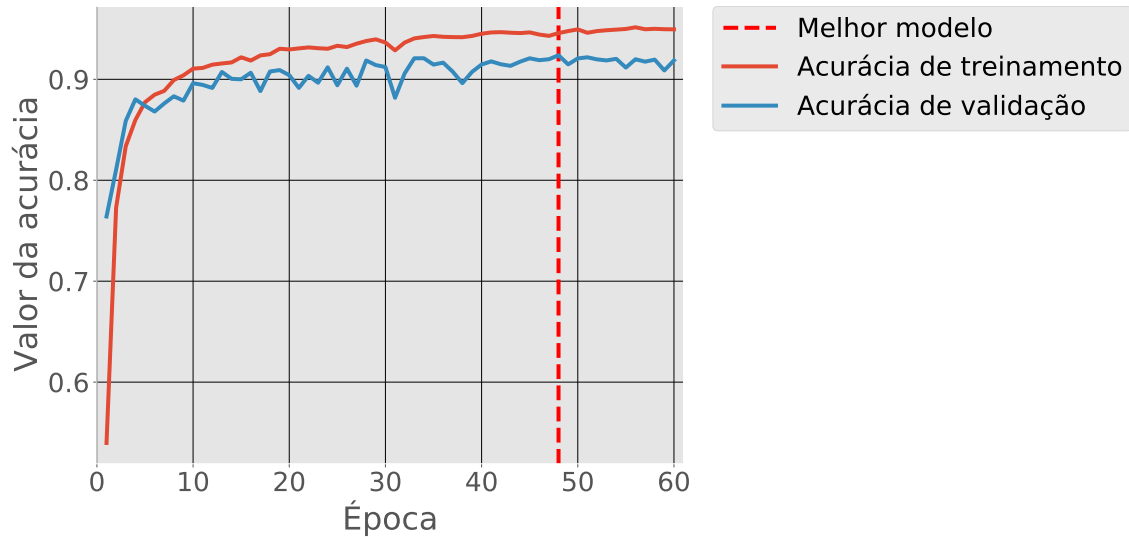
(b) Perdas médias.



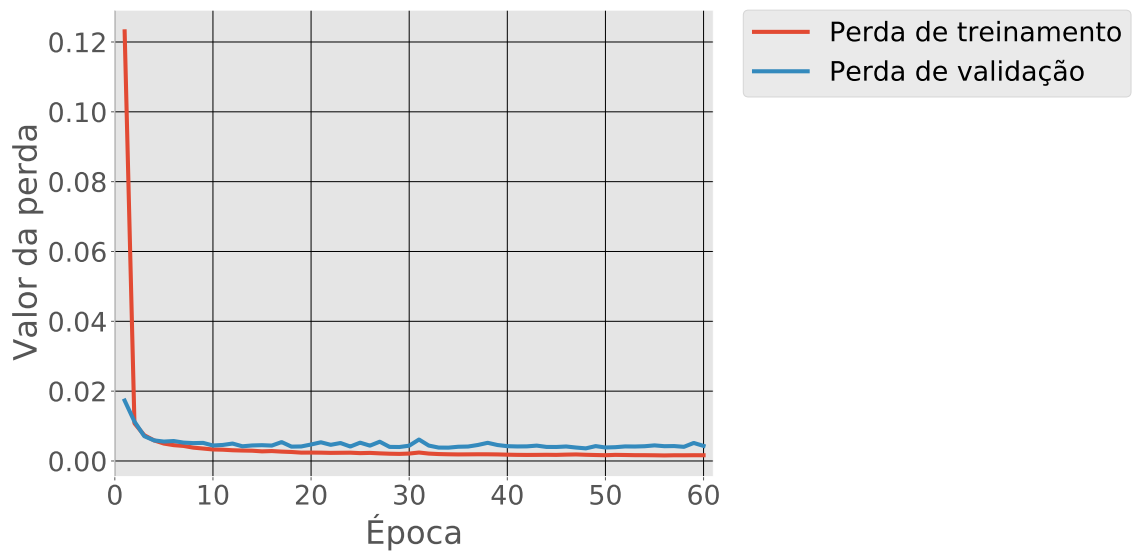
Fonte: o autor.

Figura 31 – Gráfico da acurácia e perda obtido pelo melhor modelo FCN, sem normalização.

(a) Acurácias.



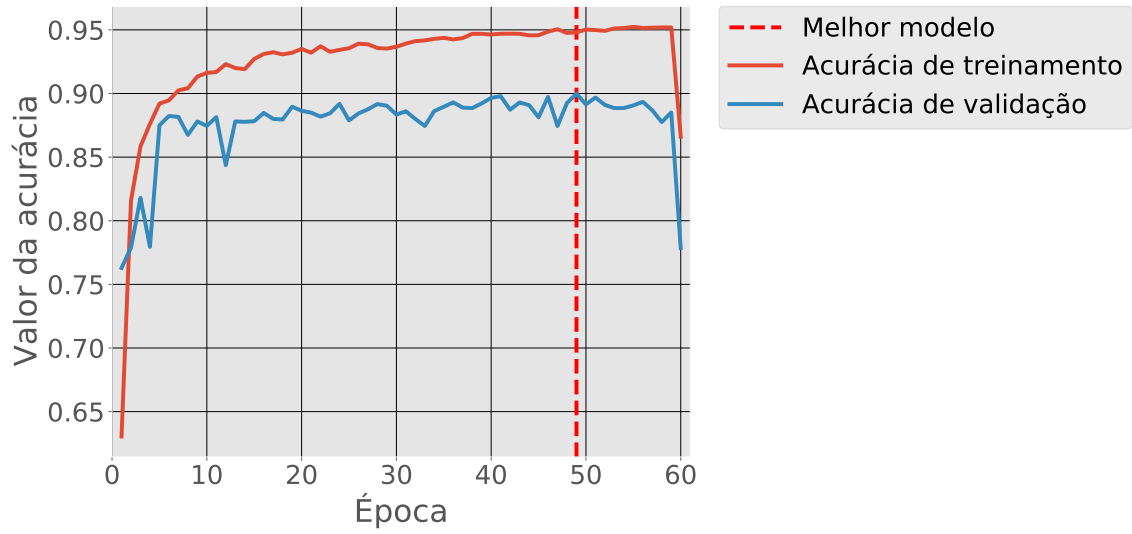
(b) Perdas.



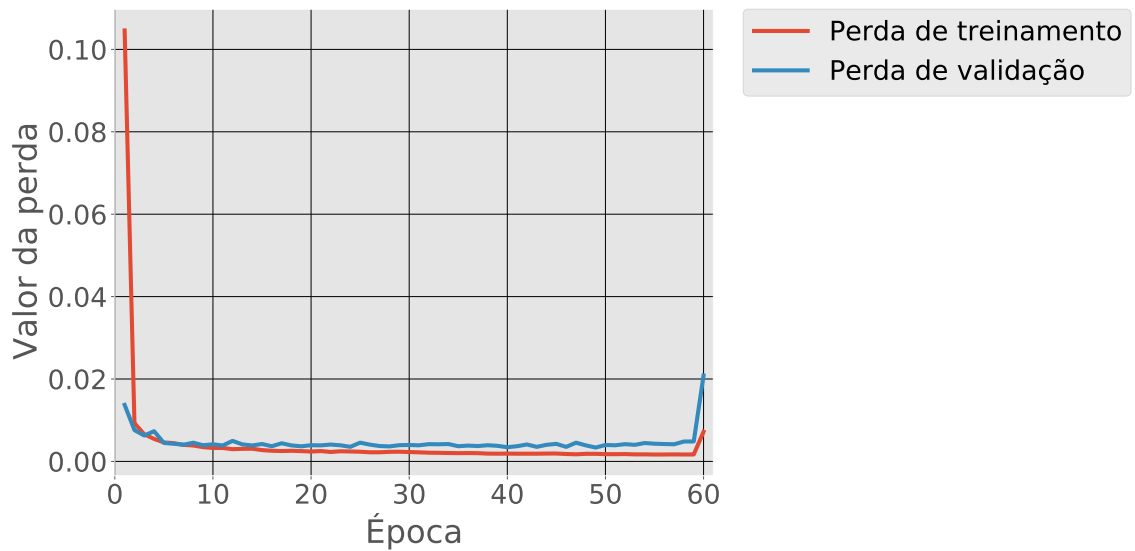
Fonte: o autor.

Figura 32 – Gráfico da acurácia e perda obtido pelo melhor modelo FCN, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.1.1.4 *DeepLabV3*

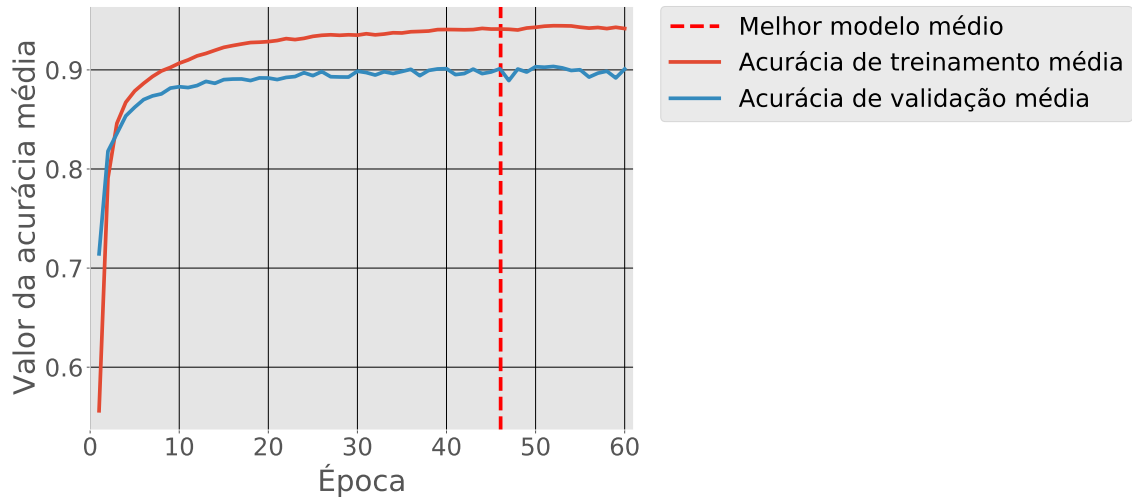
A realização da segmentação foi utilizando o modelo proposto pelos autores Chen *et al.* (2017), com sua nomenclatura conhecida como *DeepLabV3*. Mas foi aplicada uma modificação no modelo de classificação utilizada por eles. Aqui assumiu-se o modelo proposto pelos autores He *et al.* (2016), usando a ResNet-50, o algoritmo utilizado está disponível em Pytorch (2019g).

Analisando os gráficos das Figuras 33 e 34, é possível chegar à conclusão a respeito da estabilidade média dos modelos. Não importando a sua estrutura aleatória de treinamento e validação, em determinado momento ele ainda consegue uma estrutura de 90,34% sem normalização e 90,09% com normalização de segmentação para as acurácias médias, no método de similaridade de Jaccard. Essa estabilidade também se repete nas perdas médias em que os valores se mantêm inferiores a 0,2.

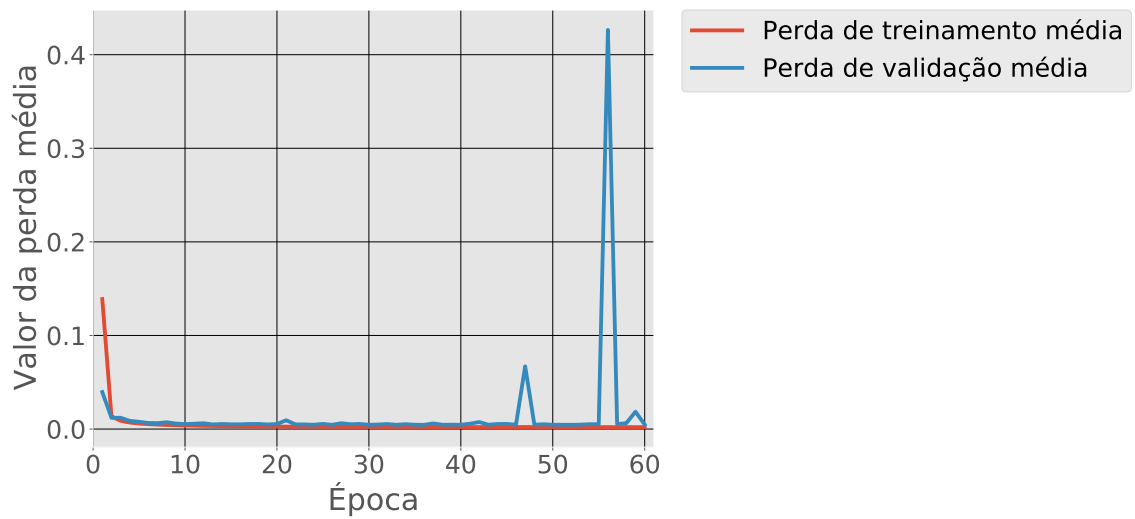
O melhor modelo obteve um valor de segmentação, de 96,46% sem normalização e 96,60% com normalização para a imagem de teste, no critério de similaridade de Jaccard. Os gráficos da Figura 35 e 36 foram obtidos do seu treinamento e validação, com o valor do seu melhor modelo de 90,62% sem normalização e 92,42% com normalização de segmentação para as acurácias médias, no método de similaridade de Jaccard. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,6014 segundos sem normalização e 0,5371 segundos com normalização, já para a GPU foi de 0,0149 segundos sem normalização e 0,0144 segundos com normalização.

Figura 33 – Gráficos médio das acurácias e perdas obtidas nos modelos DeepLabV3, sem normalização.

(a) Acurácias média.



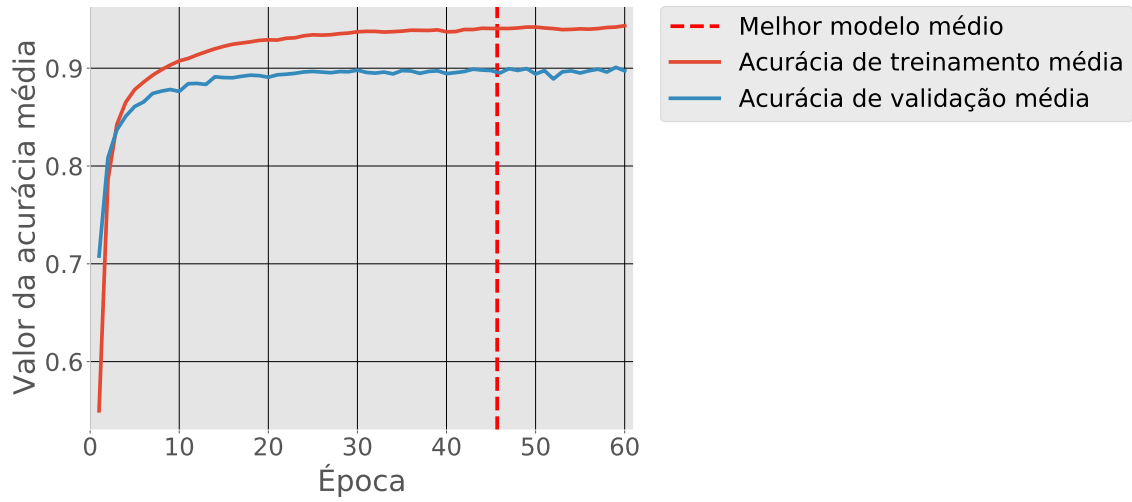
(b) Perdas média.



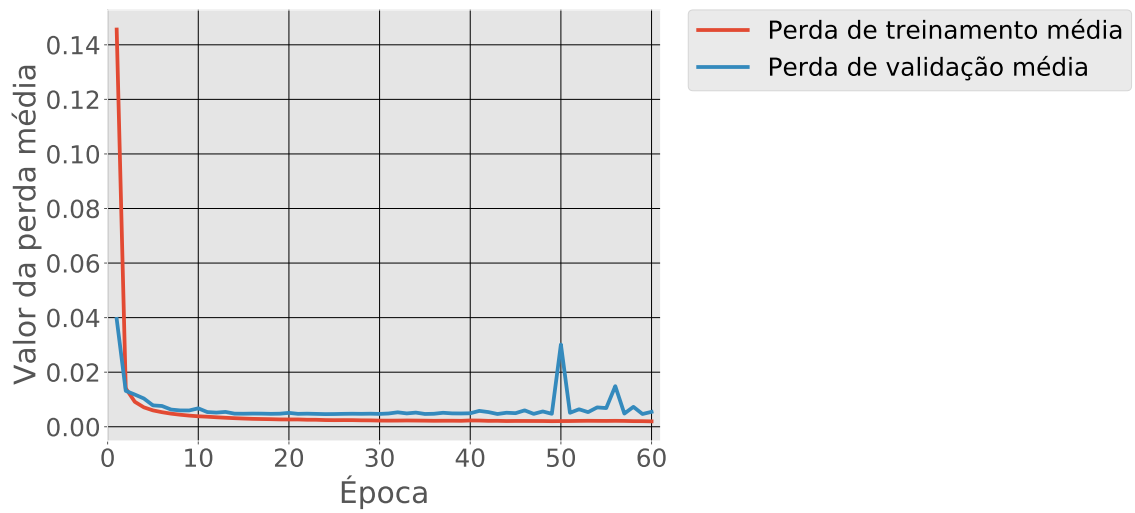
Fonte: o autor.

Figura 34 – Gráficos médio das acurácias e perdas obtidas nos modelos DeepLabV3, com normalização.

(a) Acurácias média.



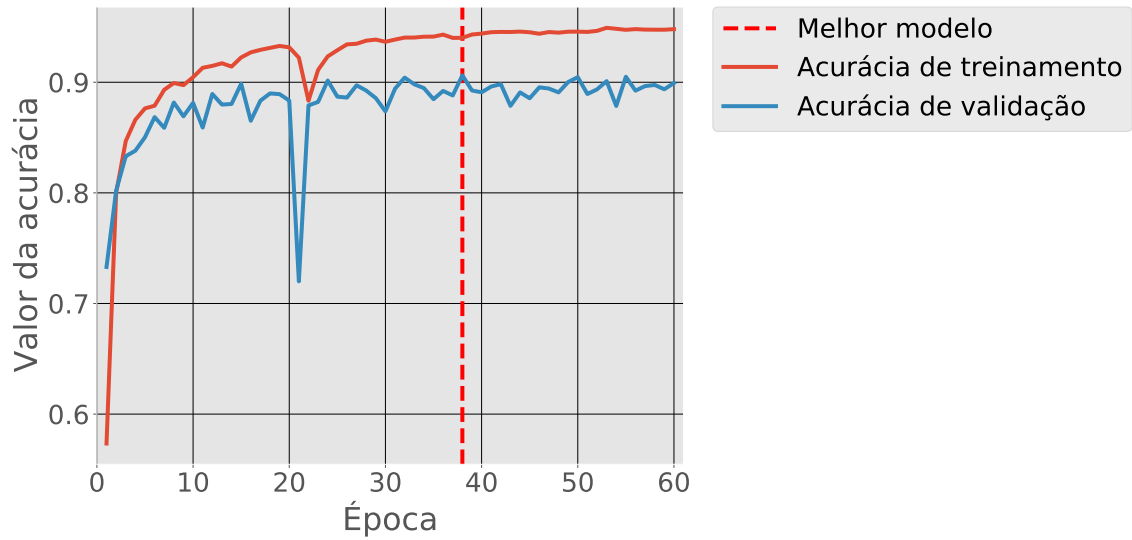
(b) Perdas média.



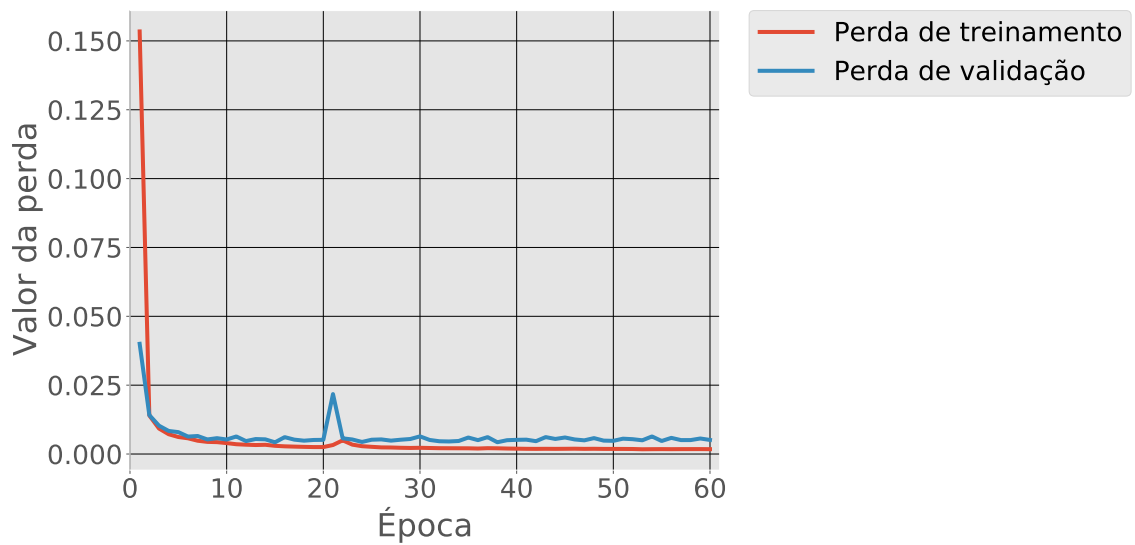
Fonte: o autor.

Figura 35 – Gráfico da acurácia e perda obtido do melhor modelo DeepLabV3, sem normalização.

(a) Acurácias.



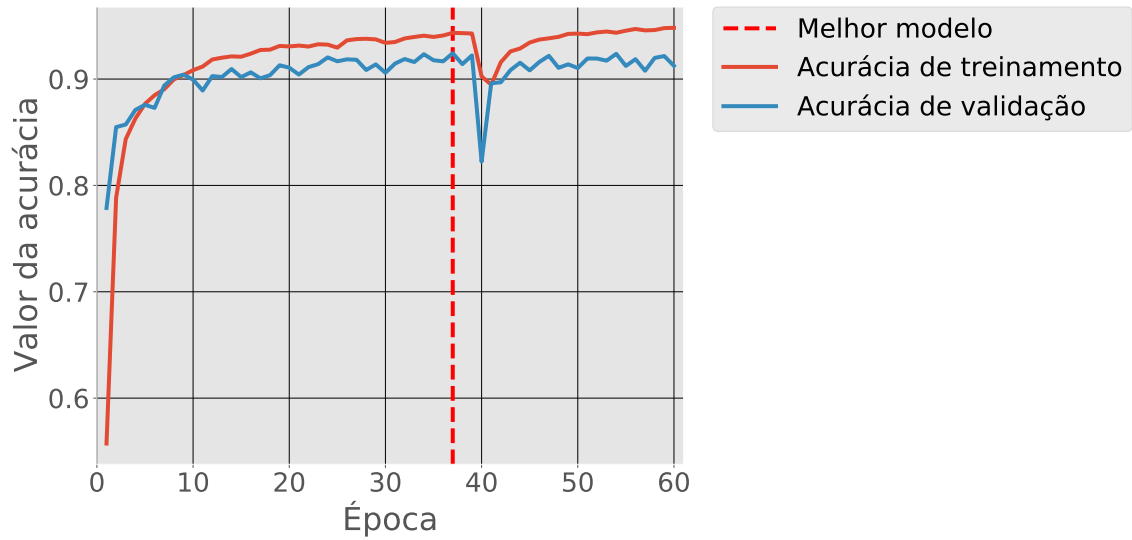
(b) Perdas.



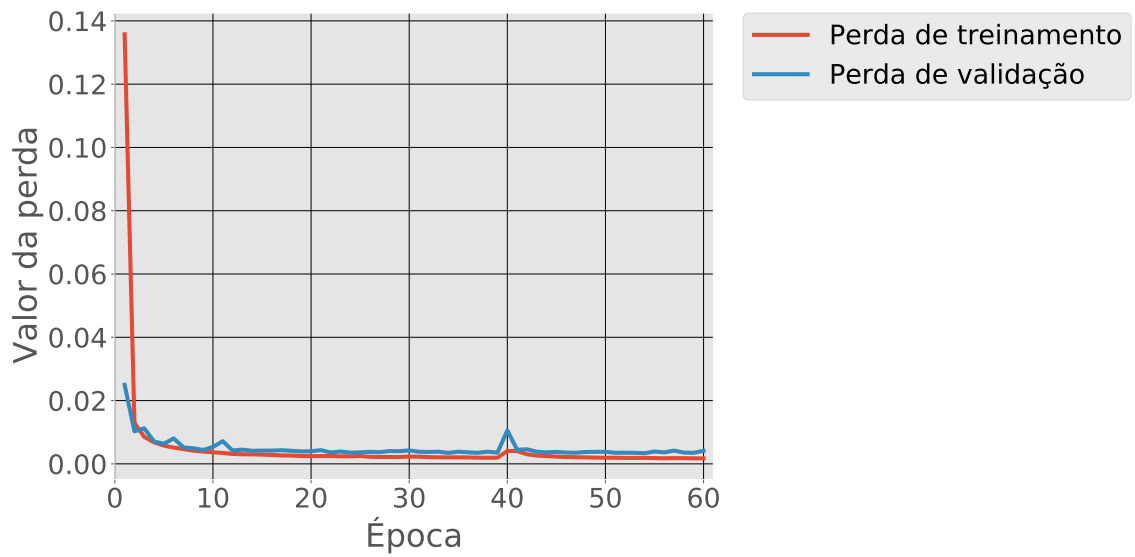
Fonte: o autor.

Figura 36 – Gráfico da acurácia e perda obtido do melhor modelo DeepLabV3, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

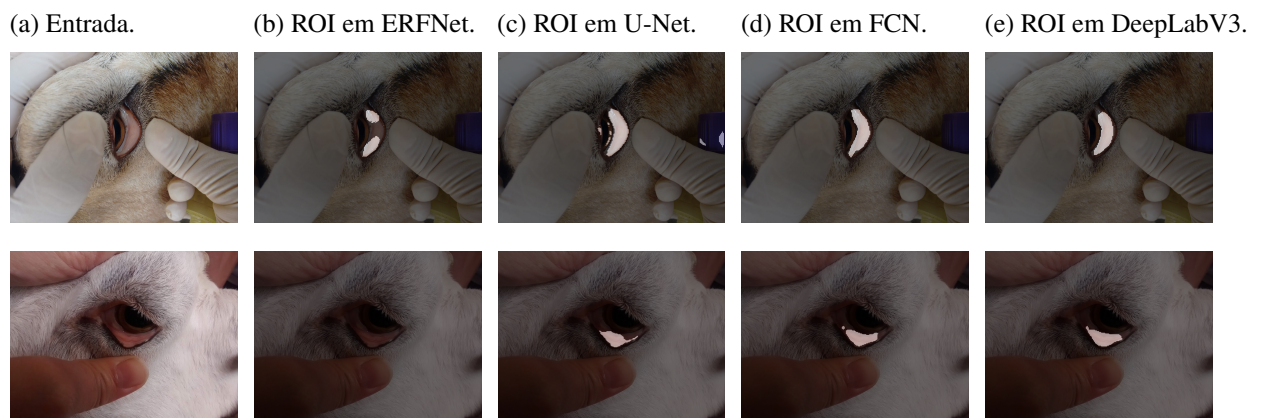
4.1.2 Aplicação em outra base de dados

Foram também realizados testes com outra base de dados cedida por pesquisadores da Universidade Federal do Maranhão (UFMA). O total de imagens utilizadas foi de 105, e essa base continha ovinos e caprinos. Alguns desses resultados são ilustrados nas Figuras 37 e 38, com amostras relacionadas a caprinos:

- A imagem da primeira coluna: Imagem de entrada para o modelo já treinado, Figura 37a e 38a;
- As imagens das demais colunas: Mesclagem da imagem de entrada com a imagem resultante do modelo, com destaque na ROI. São as Figuras 37b, 37c, 37d, 37e, 38b, 38c, 38d e 38e. A região mais clara, é o resultado da segmentação utilizando o modelo já treinado de uma aprendizagem profunda.

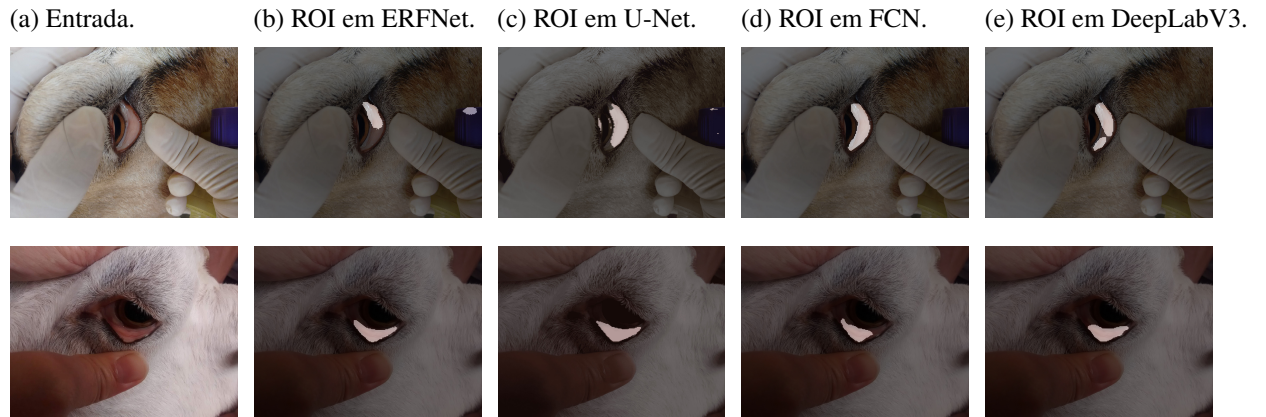
Ocorreram falhas em poucas amostras como, por exemplo, nenhuma região ser selecionada na imagem. E, ainda, algumas imagens que obtiveram as seleções enviesadas, tais como: a maior região não era a região da mucosa, seleção da pálpebra superior do olho do animal, seleção de parte do glóbulo ocular do animal. As falhas estão mais evidentes na exposição da mucosa do que na própria falha da rede, ou seja, na captura das imagens.

Figura 37 – Processo de segmentação de outra base de dados, sem normalização.



Fonte: o autor.

Figura 38 – Processo de segmentação de outra base de dados, com normalização.



Fonte: o autor.

A Tabela 3 contém os resultados das médias obtidas com os algoritmos segundo o critério apresentado na seção 2.4.5.1.1, para a segmentações das 105 imagens nessa base de dados. Foram utilizados os melhores modelos para ERFNet, U-Net, FCN DeepLabV3, sem ou com normalização.

Tabela 3 – Resultados médio da segmentação

Modelos	Sem normalização	Com normalização
ERFNet	69,07 %	73,06 %
U-Net	73,01 %	78,06 %
FCN	73,91 %	78,76 %
DeepLabV3	76,79 %	74,88 %

Fonte: o autor.

4.1.3 Aplicação no celular

Neste teste nós averiguamos a viabilidade de embarcar essa metodologia em uma aplicação móvel. Na Figura 39, é possível visualizar esse resultado, apenas com uma implementação mais simples sem o uso de funções de segmentação ou tratamento de imagem. Com a seguinte estrutura: a imagem de entrada, já configurada com o tamanho do modelo 224x224 *pixels*, e a imagem de saída, resultado do melhor modelo e o tempo de resposta do modelo. É imprescindível calcular o tempo de processamento pois muitas vezes o modelo demora muito a processar os dados de entrada até a informação de saída. Foi utilizado um celular Motorola Moto G5S Plus com um processador 8 Core e 2 GHZ. Analisando a Tabela 4, foi possível observar que os modelos ERFNet e U-Net obtiveram tempos aceitáveis na sua execução comparados aos modelos FCN e DeepLabV3. Diversos fatores podem impactar nesse comportamento, tais como:

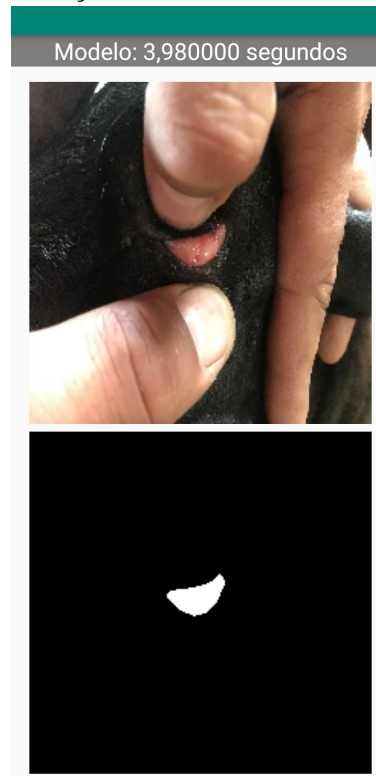
- A conversão dos modelos já treinados. Como não é possível usar diretamente os modelos salvos na plataforma Android por conta da incompatibilidade do sistema, foi utilizada uma função *TorchScript*, que realiza a função de serializar um modelo em um conjunto de instruções que podem ser invocadas independentemente da linguagem de programação Python, como programas C++ ou em dispositivos móveis (STEVENS; ANTIGA; VIEHMANN, 2020). Foi possível usar diretamente essa função no ERFNet e U-Net, mas na aplicação da FCN e DeepLabV3 foi necessário o uso de um artifício programacional, convertendo os dados em tuple, antes da serialização, o que pode ter ocasionado um gargalo ao modelo;
- A complexidade da rede. Os modelos ERFNet e U-Net são modelos simples e não tão profundos se comparados às estruturas do FCN e DeepLabV3. Esse comportamento pode ter influenciado no tempo de resposta pois quanto mais profundo for mais tempo leva para realização das conversões matemáticas do sistema operacional.

Tabela 4 – Tempo de resposta no dispositivo móvel.

Modelos	Tempo (s)	
	sem normalização	com normalização
ERFNet	0,375	0,383
U-Net	3,585	3,646
FCN	3,980	3,583
DeepLabV3	4,161	4,166

Fonte: o autor.

Figura 39 – Aplicação *mobile* do modelo gerado pela FCN sem normalização.



Fonte: o autor.

4.1.4 Análise dos resultados

No primeiro momento foi feita a escolha dos modelos justificando assim a sua aplicação:

- ERFNet é um algoritmo que tem seu destaque para uma segmentação em tempo real e rápida então foi utilizada para saber sua precisão e velocidade (ROMERA *et al.*, 2018; EROMERA, 2019);
- U-Net possui forte aplicação em imagens médicas então pode ter sua precisão elevada em imagens com uma só região a ser destacada (RONNEBERGER; FISCHER; BROX, 2015; ZHOU *et al.*, 2018; MCGLINCHY *et al.*, 2019);
- FCN e DeepLabV3 são abordadas em muitos estudos e aplicações na área de segmentação através de aprendizagem profunda (SHELHAMER; LONG; DARRELL, 2016; CHEN *et al.*, 2017; HE *et al.*, 2016; PYTORCH, 2019f; PYTORCH, 2019g).

A partir do comportamento dos gráficos de acurácia e perda foi possível retirar informações relevantes. Para os gráficos dos modelos médios, apresentado nas Figuras 21, 22, 25, 26, 29, 30, 33 e 34, é analisar os algoritmos com os dados e seu comportamento chegando à

conclusão de que as curvas de aprendizado do treinamento e de validação atingiram um bom ajuste. Analisando os resultados para os melhores modelos médios, foi possível avaliar que os melhores resultados eram obtidos acima da época 40 para o seu treinamento. O ajuste da curva de treinamento e validação para os modelos médios são bem similares aos melhores modelos, nas Figuras 23, 24, 27, 28, 31, 32, 35 e 36, os gráficos de perda do treinamento, diminuem até um ponto de estabilidade, e a validação tem uma pequena lacuna com a perda do treinamento. Enquanto a curva de treinamento e validação da acurácia cresceu ao longo das épocas.

Foi considerado o uso da normalização dos dados, como apresentado na Subseção 2.4.3, para o entendimento do seu comportamento, pois em algumas problemáticas a normalização dos dados tem um efeito de elevar a acurácia. Não foi o caso para esses dados, houve um aumento irrelevante da acurácia do modelo e outros instantes a normalização foi pior do que os dados não normalizado, nesse caso para FCN, como apresentado na Tabela 5.

Também foi analisada a estrutura de tempo de resposta do modelo já treinado através da imagem de teste. O que foi importante para saber se o modelo tem uma resposta rápida para uso da CPU e GPU. A normalização não afetou no desempenho de tempo como já era esperado, pois a normalização só afeta nos dados de entrada e não a quantidade de operações. Se for pressupor uma aplicação que funcione com seu processamento em nuvem e que disponha de uma GPU o melhor algoritmo seria a U-Net, pois tem o menor tempo de resposta. Agora para uma aplicação em um computador local que disponha de uma CPU o melhor algoritmo seria a ERFNet, pois tem o menor tempo de resposta, como apresentado na Tabela 5.

Para aplicação na Subseção 4.1.2, utilizando os melhores modelos já treinados obtidos para ERFNet, U-Net, FCN e DeepLabV3, assim tendo um melhor entendimento do comportamento dos modelos para as imagens de outra base de dados, que foram coletadas de forma diferente, pois nessas imagens não havia uma preocupação de mostrar o olho do animal. Já para nossa base de dados, a ideia era dar destaque maior a mucosa, escondendo assim o olho do animal, como apresentado na Seção 2.3. Os resultados nas Figuras 37 e 38, em alguns imagens apresentam ruídos mesmo assim a região da mucosa tem maior destaque.

O pequeno e médio produtor não têm acesso à internet, com isso impossibilitando uma aplicação que funcione com seu processamento em nuvem. O projeto também levou em consideração uma aplicação *mobile*, então na Subseção 4.1.3 foi apresentada uma parte dessa aplicação para analisar o tempo de resposta e o resultado da aplicação usando um celular, como apresentado na Tabela 4 e na Figura 39. Essa parte se torna interessante pois é possível ter uma

ideia de qual algoritmo é o mais rápido para realizar a etapa de segmentação em uma aplicação *mobile* e também levar em conta sua acurácia.

Essa averiguação do tempo foi considerada nas primeiras fases do estudo com aprendizagem profunda. Antes de aplicar uma imagem de entrada para o treinamento do modelo com tamanho de 224x224 *pixels* (foi adotado o tamanho porque é usada na documentação da biblioteca de aprendizado em Pytorch (2019h)). Aplicou-se um tamanho 800x800 *pixels*, o mesmo que foi adotado nas imagens coletadas pela aplicativo desenvolvida em *Android*, apresentada na Seção 2.3. O tamanho maior de entrada fazia com que o treinamento fosse mais lento, e também o tempo de resposta fosse elevado, por exemplo da FCN que tem um tempo na CPU entorno de 0,4 segundos passava para aproximadamente 10 segundos, quando maior a imagem mais tempo leva para o modelo processá-la. Na Figura 40 é ilustrada a segmentação realizada pelo modelo FCN sem normalização. Realizando o processo apresentado na Subseção 2.4.5.1.1 da imagem da Figura 40b (que foi demarcada a ROI de forma manual), com a imagem da Figura 40c (resultado da FCN, sem normalização) temos uma precisão de segmentação de 96,65%. A Figura 40c é a imagem resultante do modelo, com o seu redimensionamento para o tamanho original, caso tamanho da imagem da Figura 40a. O processo de redimensionamento para o tamanho original é interessante para a próxima etapa, nesse caso a segmentação da região, para coletar os dados originais sem mudanças da sua estrutura de qualidade. Isso é importante para a classificação, apresentado na Seção 4.2, pois a imagem pode conter pontos relevante com uma qualidade maior e sem previas alterações. A etapa final do processo de segmentação é a imagem da Figura 40d, em que é possível notar que foi realizada uma etapa de rotação em um sentido horizontal.

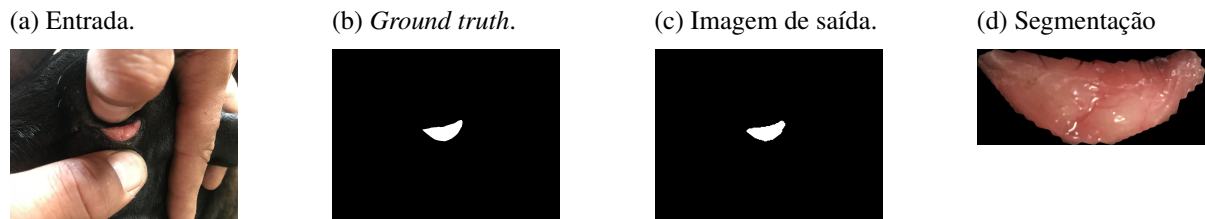
Tabela 5 – Resultados obtidos nos testes para os melhores modelos.

Modelos	Acurácia do Modelo (IoU)		Acurácia de Teste (IoU)		Tempo CPU (s)		Tempo GPU (s)	
	SN	CN	SN	CN	SN	CN	SN	CN
ERFNet	90,35%	90,95%	96,04%	97,08%	0,1034	0,1152	0,0210	0,0234
U-Net	92,07%	91,96%	97,20%	97,29%	0,5477	0,4433	0,0075	0,0074
FCN	92,38%	89,98%	96,65%	90,90%	0,4461	0,4549	0,0116	0,0132
DeepLabV3	90,62%	92,42%	96,46%	96,60%	0,6014	0,5371	0,0149	0,0144

Fonte: o autor.

Legenda: SN: sem normalização; CN: com normalização.

Figura 40 – Processo de segmentação para o melhor modelo FCN, sem normalização.



Fonte: o autor.

4.2 Classificação

A separação dos dados para geração dos modelos para classificação foi:

- **Conjunto de treino:** 84 imagens, mas foi aplicado uma estrutura de aumento de dados, passando assim 252 imagens;
- **Conjunto de validação:** 21 imagens;
- **Conjunto de teste:** 1 imagem.

No conjunto de validação também se preocupou em dividir as classes de forma ponderada e aleatória, assim 10 imagens de uma classe e 11 imagens de outra classe.

4.2.1 Modelos

Existe diversos algoritmos de classificação, se for comparar aos de segmentação. Para esse projeto foi selecionado apenas alguns desses algoritmos para criação dos modelos.

4.2.1.1 AlexNet

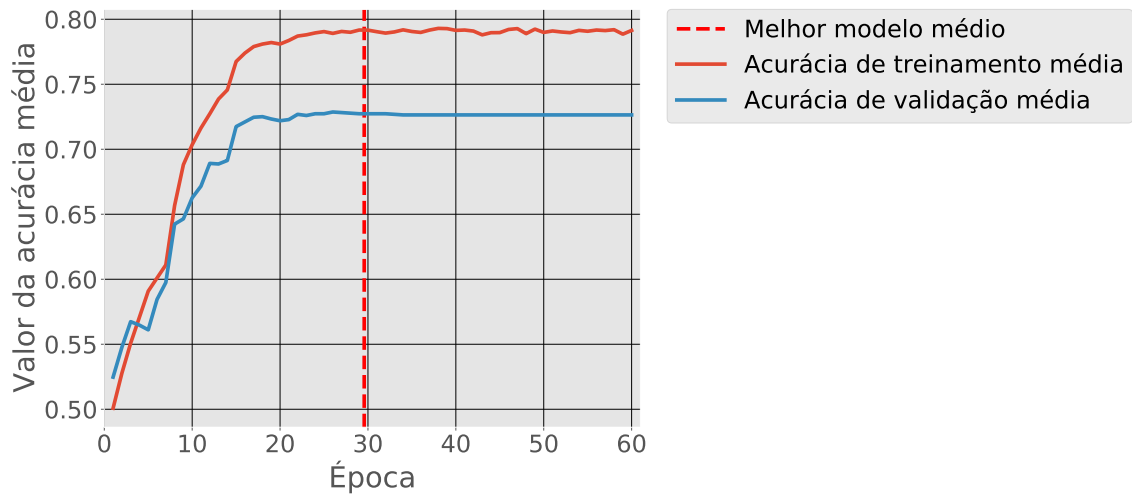
A realização da classificação foi utilizando o modelo proposto pelos autores Krizhevsky (2014), o algoritmo utilizado está disponível em Pytorch (2019a).

Analisando os gráficos das Figuras 41 e 42, é possível chegar à conclusão que as médias de acerto da validação dos modelos se estabiliza entre valores de 70,00% a 75,00%, isso com e sem a normalização. A máxima para as médias da curva de validação do treinamento foi de 72,00%, sem normalização e 74,66%, com normalização. As curvas das acurácias médias e das perdas médias mostraram uma distância elevada. No gráfico das perdas médias, apresentado nas Figuras 41b e 42b, a distância dos valores de treinamento e validação foi de 0,1139, isso sem normalização e 0,0331, com normalização, valores retirados para a época 29, sem normalização, e época 33 com normalização.

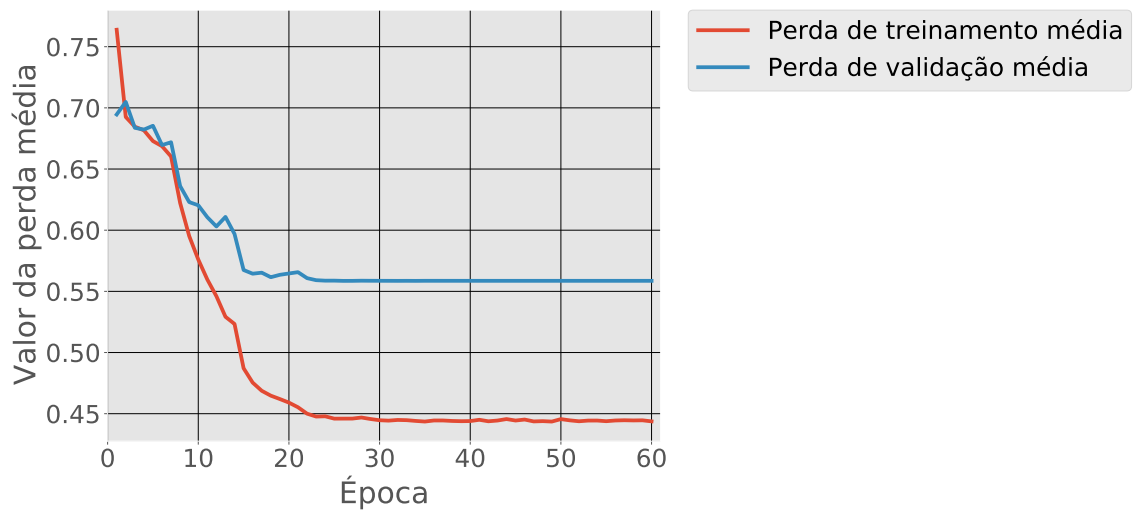
Quando isolamos o melhor modelo ele se encontra numa situação oposta aos dados médios, gráficos não são tão comportados. Os gráficos das Figuras 43 e 44, foram obtidos do melhor modelo, em que obteve uma resposta correta no conjunto de teste e o maior valor de acerto para o conjunto de validação, em caso de empate foi analisando o menor valor de perda para aquele determinado momento de melhor validação. Para o gráfico da Figura 43, estrutura sem normalização, o valor máximo da acurácia que corresponde à época 11, para esse modelo foi de 95,23%. Sua distância na perda para essa época foi de 0,1242. Já para o gráfico da Figura 44, estrutura com normalização, o valor da acurácia da época 60 desse modelo foi de 90,47%. Sua distância na perda para essa época foi de 0,0897. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,027782 segundos sem normalização e 0,024549 segundos com normalização, já para a GPU foi de 0,001506 segundos sem normalização e 0,001276 segundos com normalização.

Figura 41 – Gráficos das acurácias e perdas médias obtidas nos modelos AlexNet, sem normalização.

(a) Acurácias médias.



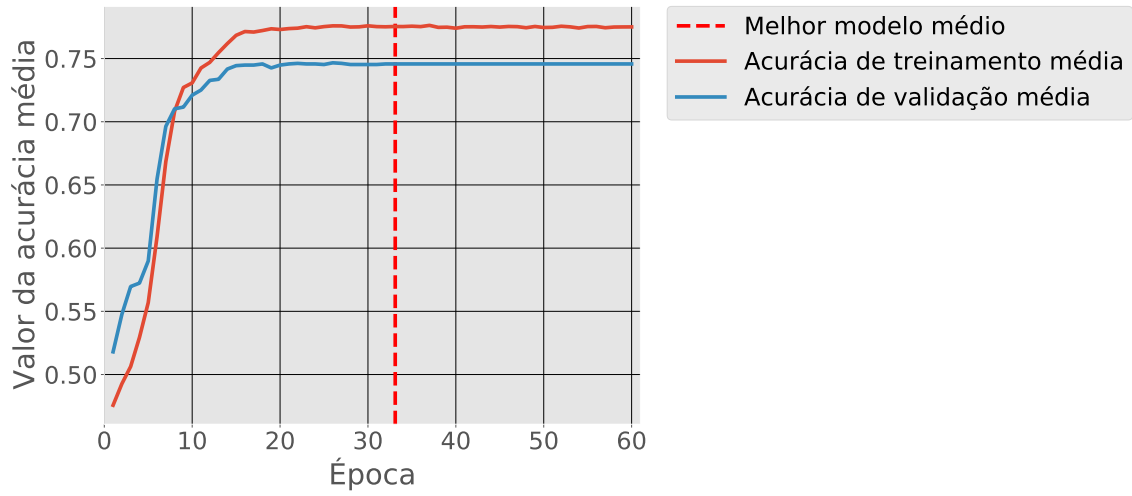
(b) Perdas médias.



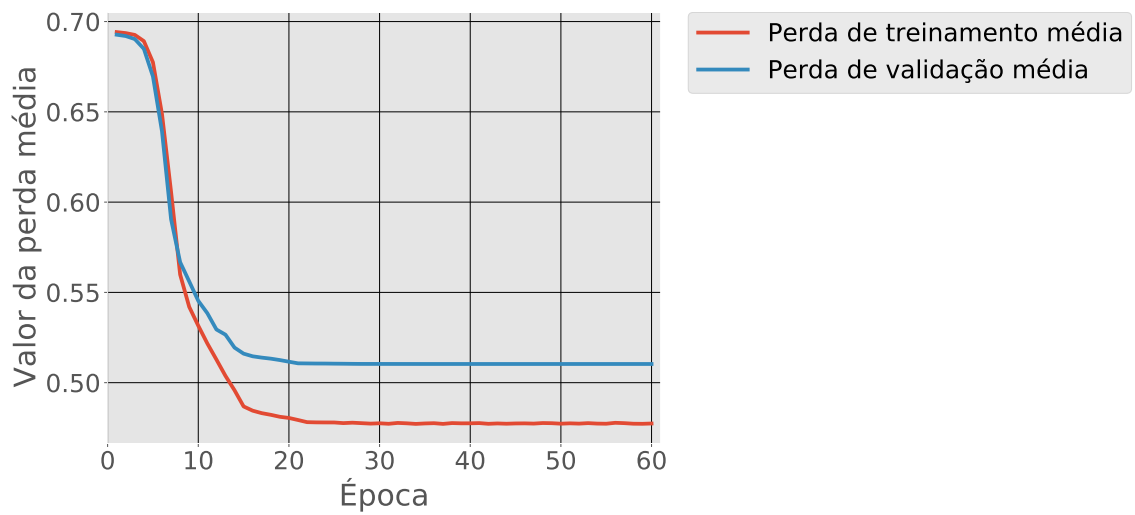
Fonte: o autor.

Figura 42 – Gráficos das acurácias e perdas médias obtidas nos modelos AlexNet, com normalização.

(a) Acurácias médias.



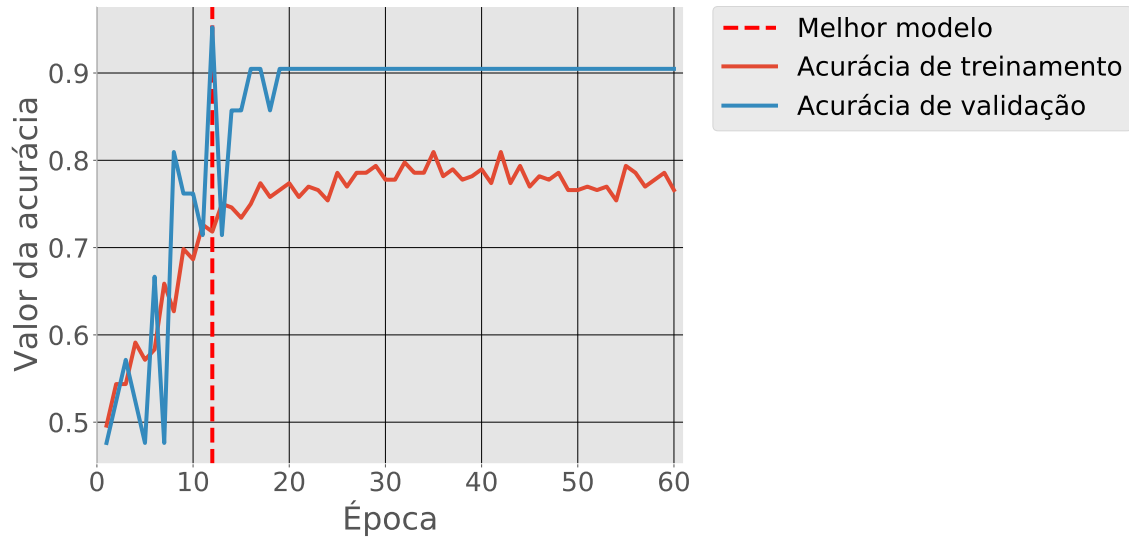
(b) Perdas médias.



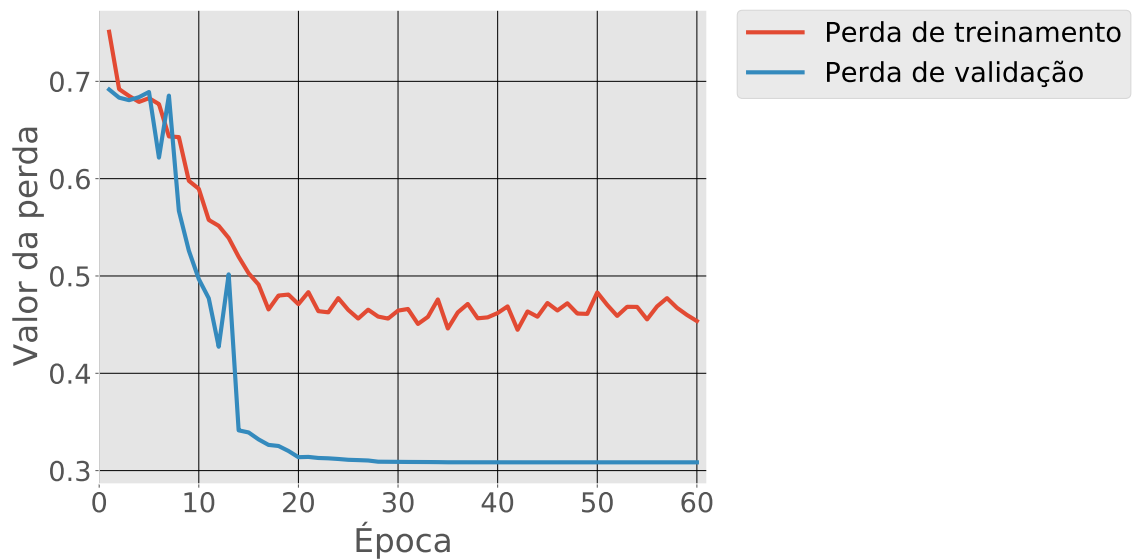
Fonte: o autor.

Figura 43 – Gráfico da acurácia e perda obtido pelo melhor modelo AlexNet, sem normalização.

(a) Acurácias.



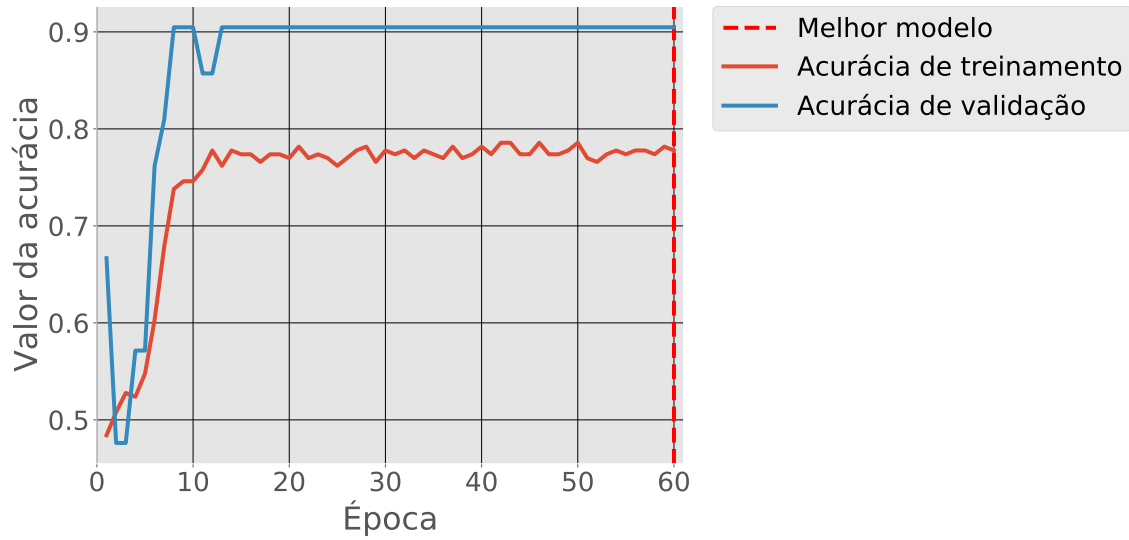
(b) Perdas.



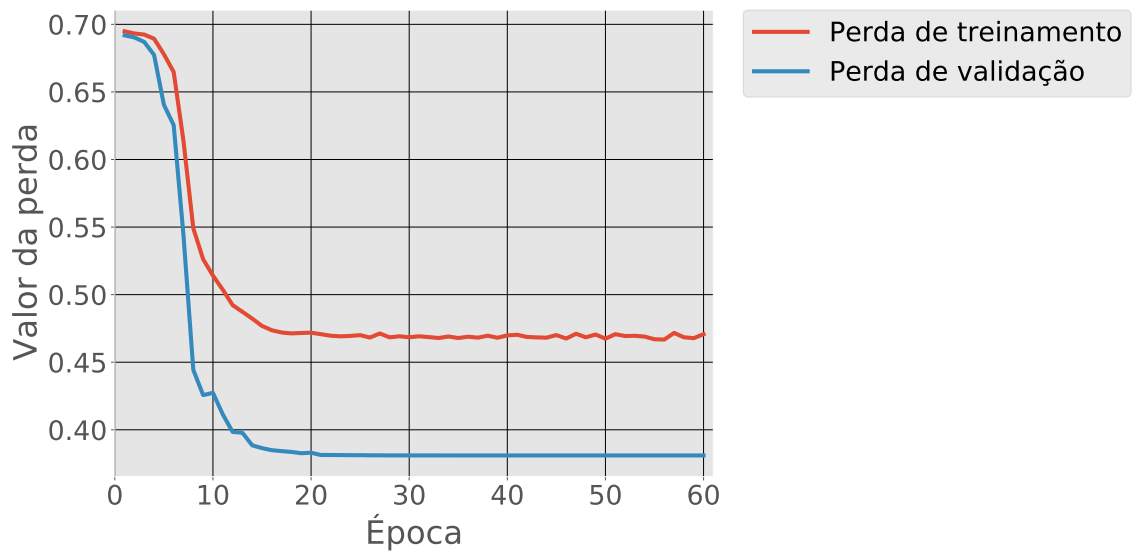
Fonte: o autor.

Figura 44 – Gráfico da acurácia e perda obtido pelo melhor modelo AlexNet, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.2.1.2 MobileNetV2

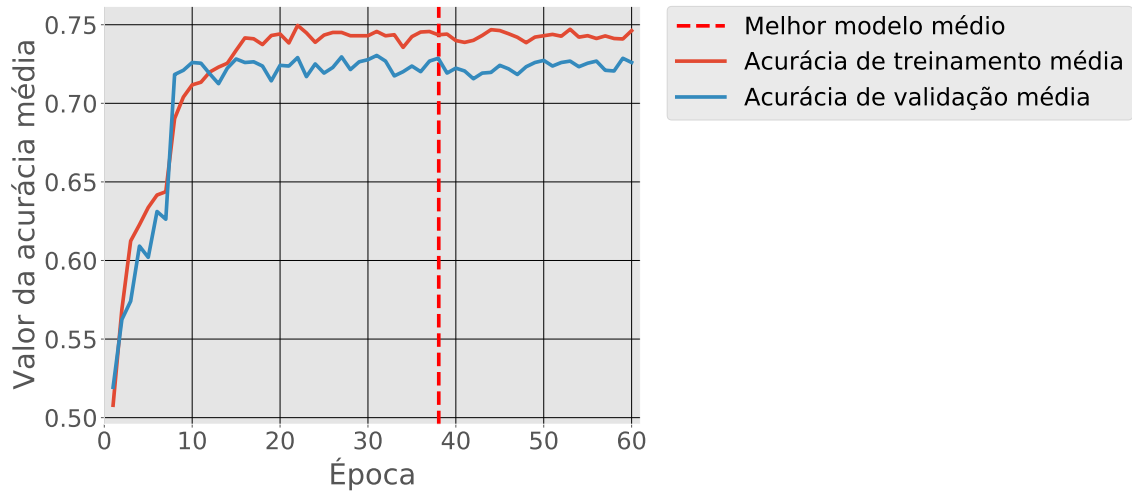
A realização da classificação foi utilizando o modelo proposto pelos autores Sandler *et al.* (2018), o algoritmo utilizado está disponível em Pytorch (2019b).

Analisando os gráficos das Figuras 45 e 46, é possível chegar à conclusão que as médias de acerto da validação dos modelos se estabiliza entre valores de 70,00% a 75,00%, isso com e sem a normalização. A máxima para as médias da curva de validação do treinamento foi de 73,04%, sem normalização e 73,22%, com normalização. As curvas da acurácias médias e das perdas médias mostraram uma distância elevada. No gráfico médio das perdas, apresentado nas Figuras 41b e 42b, a distância dos valores de treinamento e validação foi de 0,0658, isso sem normalização e 0,0596, com normalização, valores retirados para a época 38, sem e com normalização.

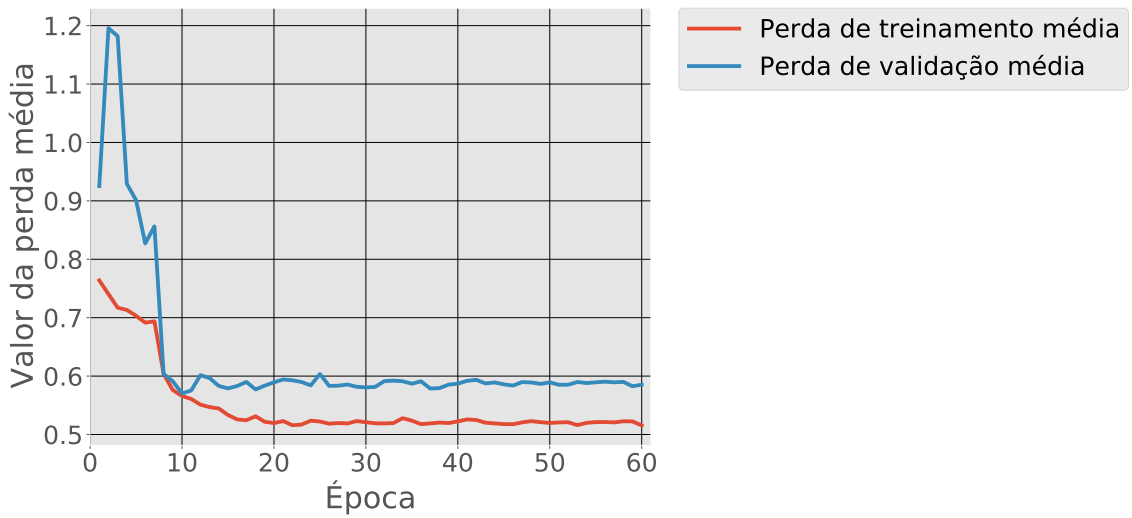
Quando isolamos o melhor modelo ele se encontra numa situação oposta aos dados médios, gráficos não são tão comportados. Os gráficos das Figuras 47 e 48, foram obtidos do melhor modelo, em que obteve uma resposta correta no conjunto de teste e o maior valor de acerto para o conjunto de validação, em caso de empate foi analisando o menor valor de perda para aquele determinado momento de melhor validação. Para o gráfico da Figura 47, estrutura sem normalização, o valor máximo da acurácia que corresponde à época 59, para esse modelo foi de 95,23%. Sua distância na perda para essa época foi de 0,2630. Já para o gráfico da Figura 48, estrutura com normalização, o valor da acurácia da época 14 desse modelo foi de 95,23%. Sua distância na perda para essa época foi de 0,1065. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,034852 segundos sem normalização e 0,035160 segundos com normalização, já para a GPU foi de 0,001276 segundos sem normalização e 0,007482 segundos com normalização.

Figura 45 – Gráficos das acurácias e perdas médias obtidas nos modelos MobileNetV2, sem normalização.

(a) Acurácias médias.



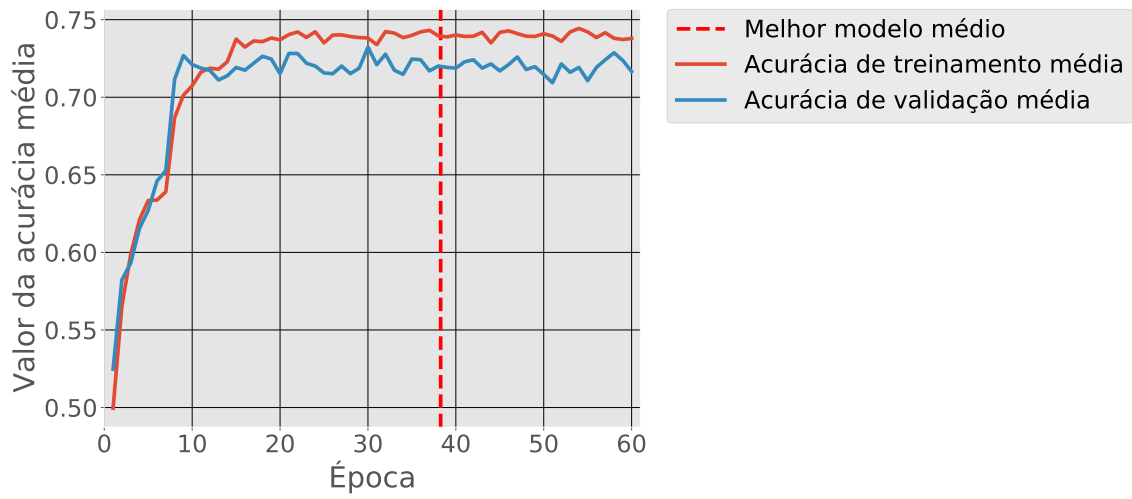
(b) Perdas médias.



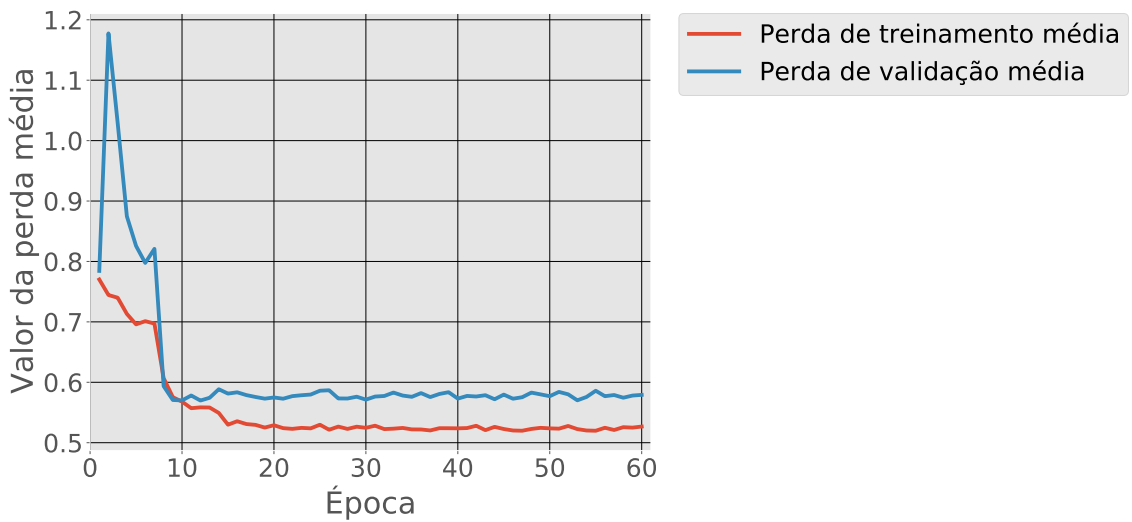
Fonte: o autor.

Figura 46 – Gráficos das acurácias e perdas médias obtidas nos modelos MobileNetV2, com normalização.

(a) Acurácias médias.



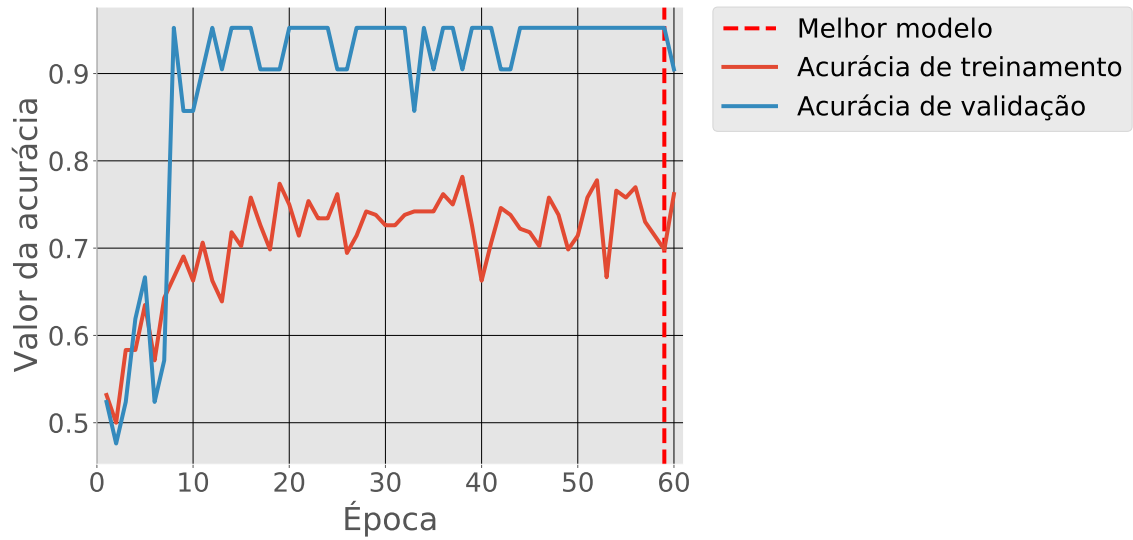
(b) Perdas médias.



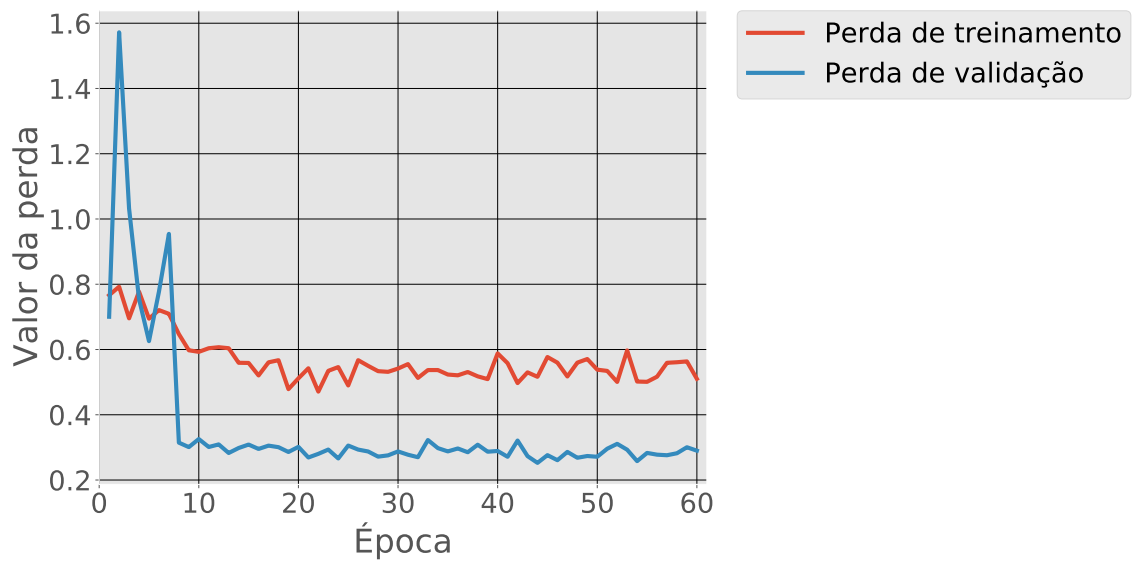
Fonte: o autor.

Figura 47 – Gráfico da acurácia e perda obtido pelo melhor modelo MobileNetV2, sem normalização.

(a) Acurácias.



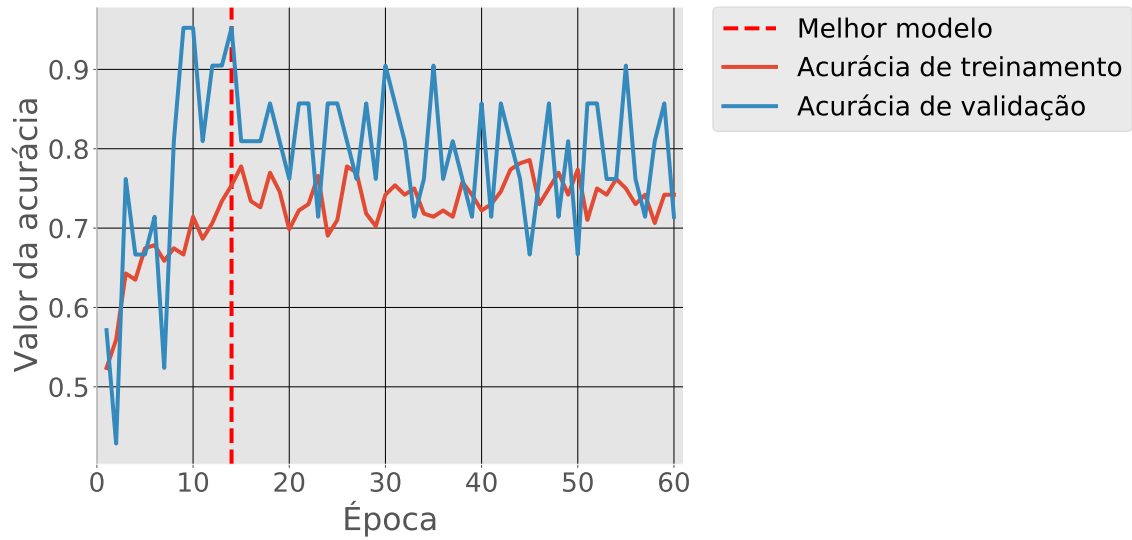
(b) Perdas.



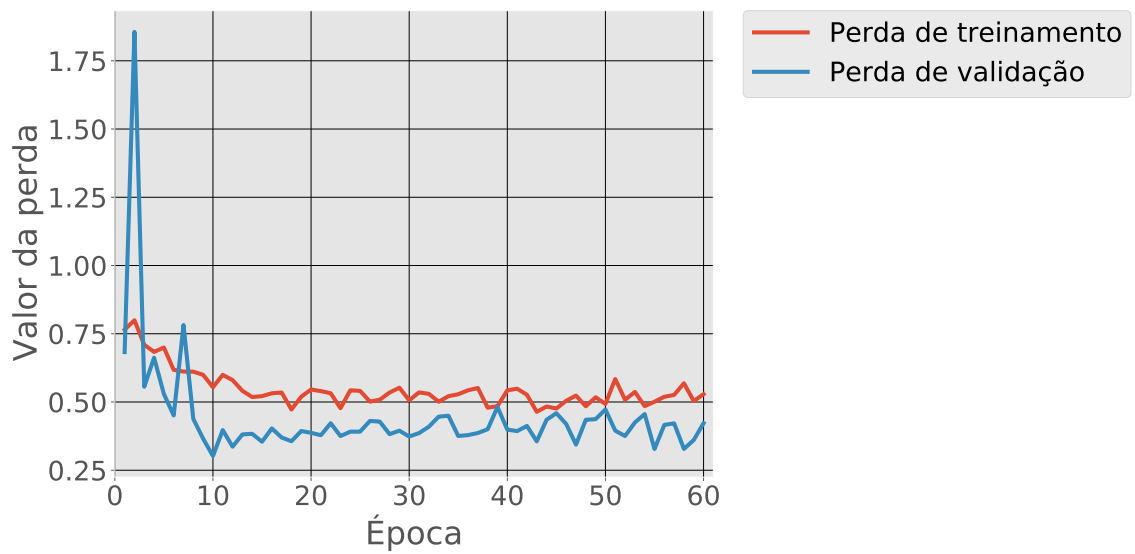
Fonte: o autor.

Figura 48 – Gráfico da acurácia e perda obtido pelo melhor modelo MobileNetV2, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.2.1.3 Deep Residual Learning (ResNet)

É possível encontrar diversas modificações para o modelo Deep Residual Learning (ResNet), por isso foi testada apenas duas dessas modificações, a ResNet18, apresentada na Subseção 4.2.1.3.1 e ResNet152, apresentada na Subseção 4.2.1.3.2 (HE *et al.*, 2016; PYTORCH, 2019c; PYTORCH, 2019d).

4.2.1.3.1 ResNet18

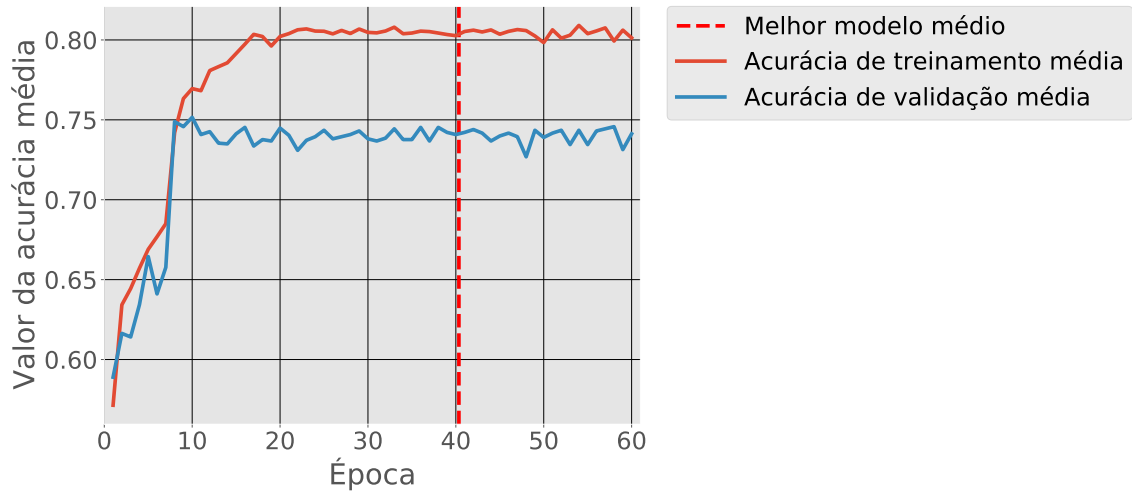
A realização da classificação foi utilizando o modelo proposto pelos autores He *et al.* (2016), o algoritmo utilizado está disponível em Pytorch (2019c).

Analisando os gráficos das Figuras 49 e 50, é possível chegar à conclusão que as médias de acerto da validação dos modelos se estabiliza entre valores de 73,00% a 82,00%, isso com e sem a normalização. A máxima para as médias da curva de validação do treinamento foi de 75,15%, sem normalização e 73,76%, com normalização. As curvas da acurácias médias e da perdas médias obtiveram uma distância elevada. No gráfico médio das perdas, apresentado nas Figuras 49b e 50b, a distância dos valores de treinamento e validação foi de 0,1193, isso sem normalização e 0,1447, com normalização, valores retirados para a época 40, sem normalização, e época 33, com normalização.

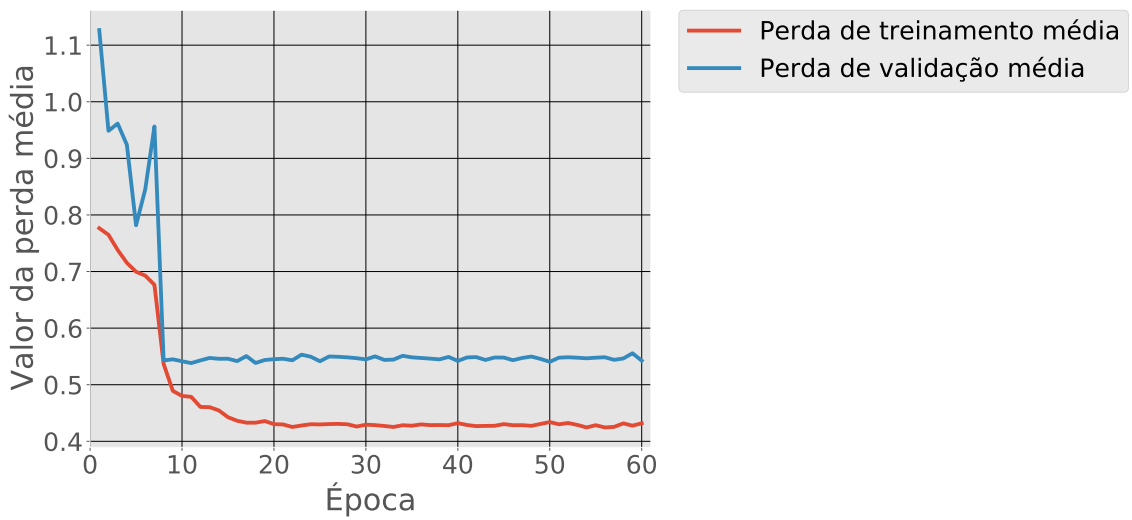
Quando isolamos o melhor modelo ele se encontra numa situação oposta aos dados médios, gráficos não são tão comportados. Os gráficos das Figuras 51 e 52, foram obtidos do melhor modelo, em que obteve uma resposta correta no conjunto de teste e o maior valor de acerto para o conjunto de validação, em caso de empate foi analisando o menor valor de perda para aquele determinado momento de melhor validação. Para o gráfico da Figura 51, estrutura sem normalização, o valor máximo da acurácia que corresponde à época 6, para esse modelo foi de 95,23%. Sua distância na perda para essa época foi de 0,3944. Já para o gráfico da Figura 52, estrutura com normalização, o valor da acurácia da época 14 desse modelo foi de 95,23%. Sua distância na perda para essa época foi de 0,5644. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,052170 segundos sem normalização e 0,053361 segundos com normalização, já para a GPU foi de 0,005635 segundos sem normalização e 0,004359 segundos com normalização.

Figura 49 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet18, sem normalização.

(a) Acurácias médias.



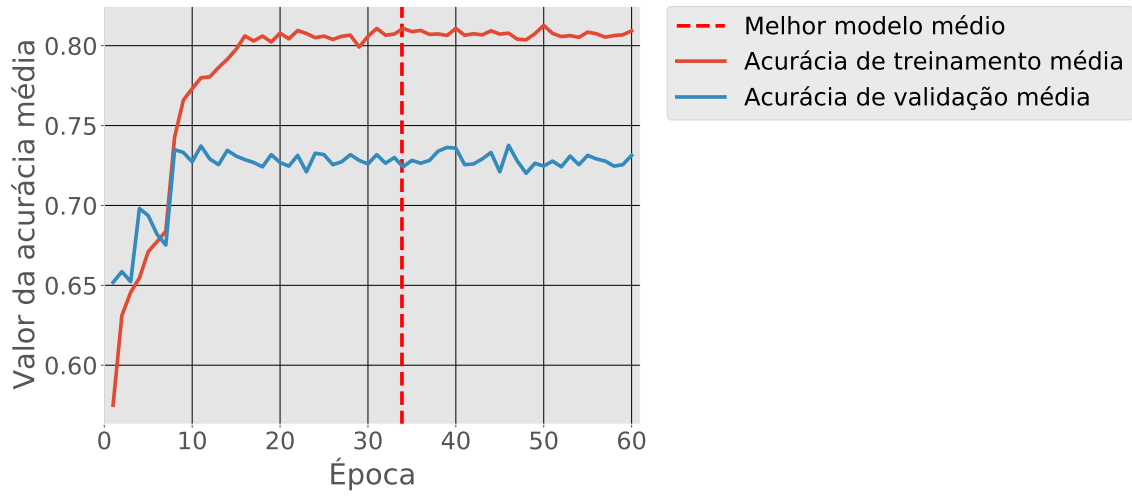
(b) Perdas médias.



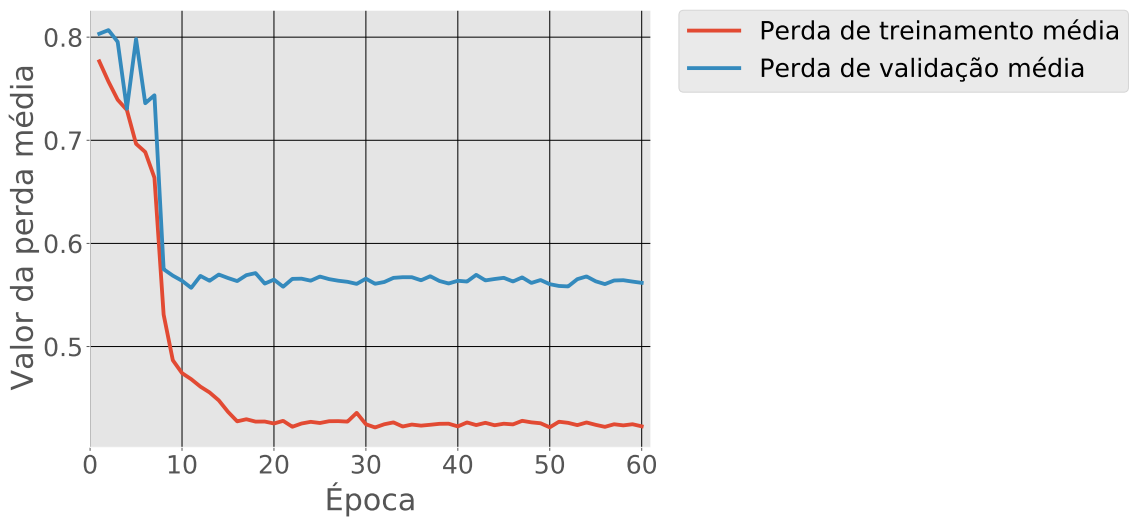
Fonte: o autor.

Figura 50 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet18, com normalização.

(a) Acurácias médias.



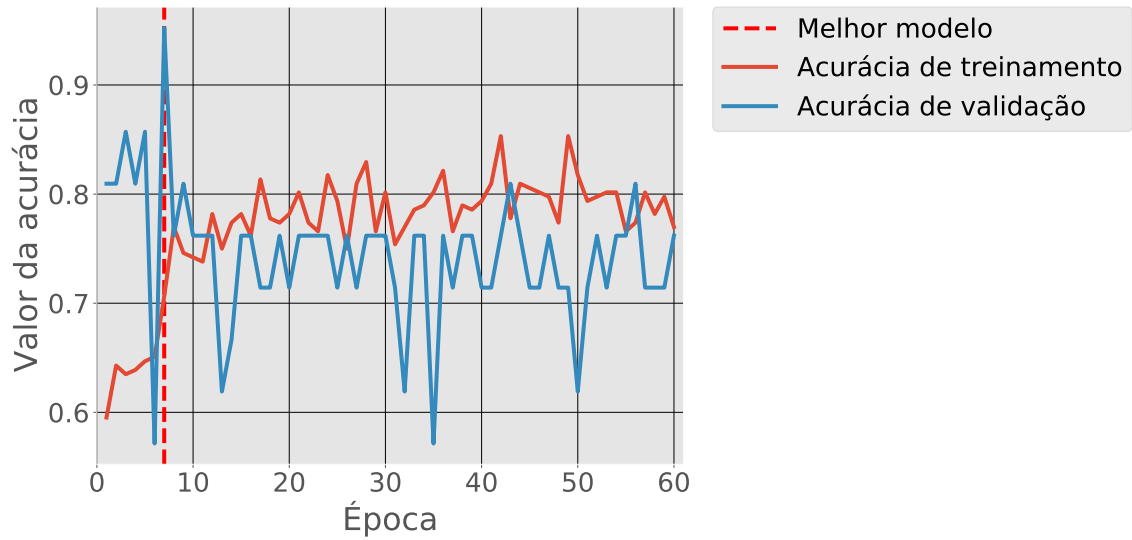
(b) Perdas médias.



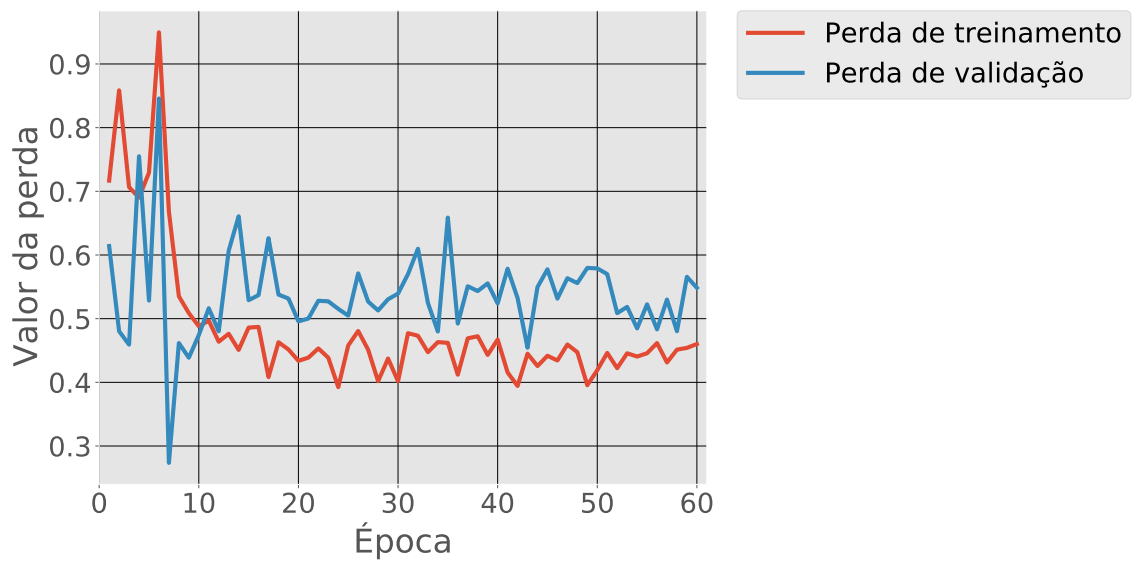
Fonte: o autor.

Figura 51 – Gráfico da acurácia e perda obtido pelo melhor modelo ResNet18, sem normalização.

(a) Acurácias.



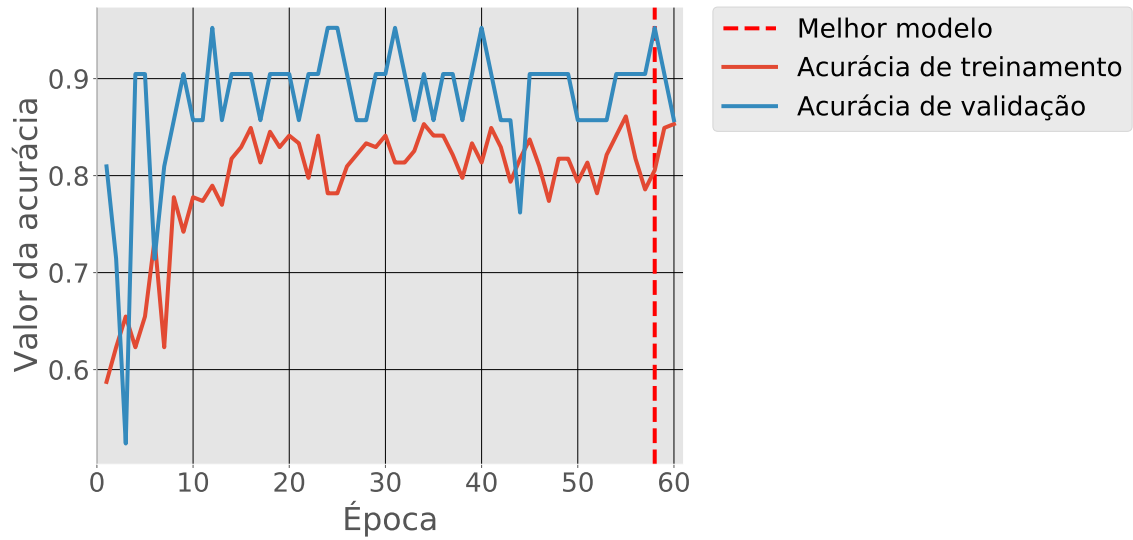
(b) Perdas.



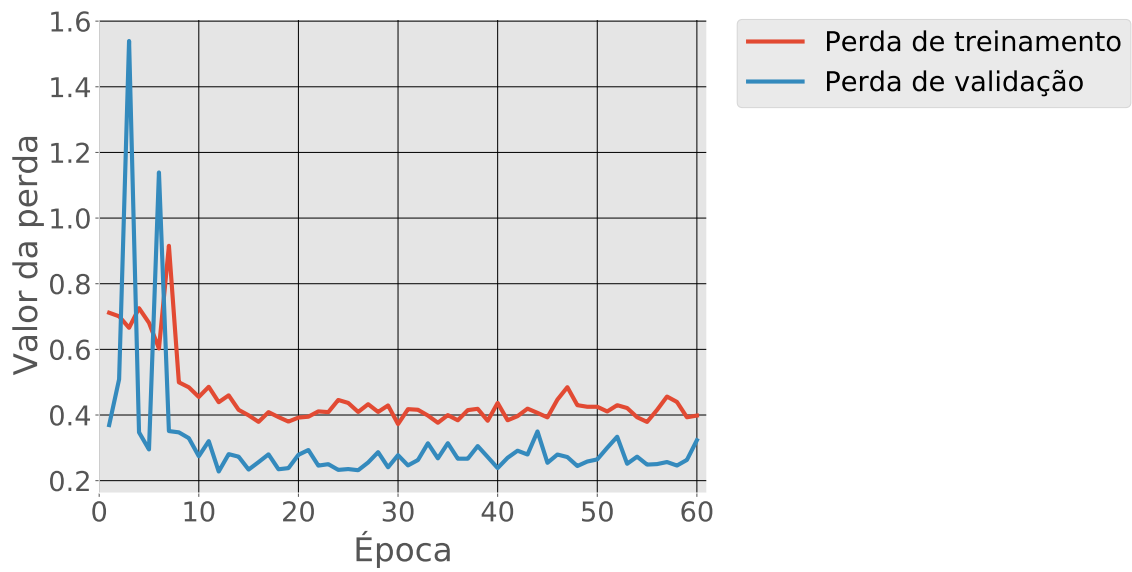
Fonte: o autor.

Figura 52 – Gráfico da acurácia e perda obtido pelo melhor modelo ResNet18, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.2.1.3.2 ResNet152

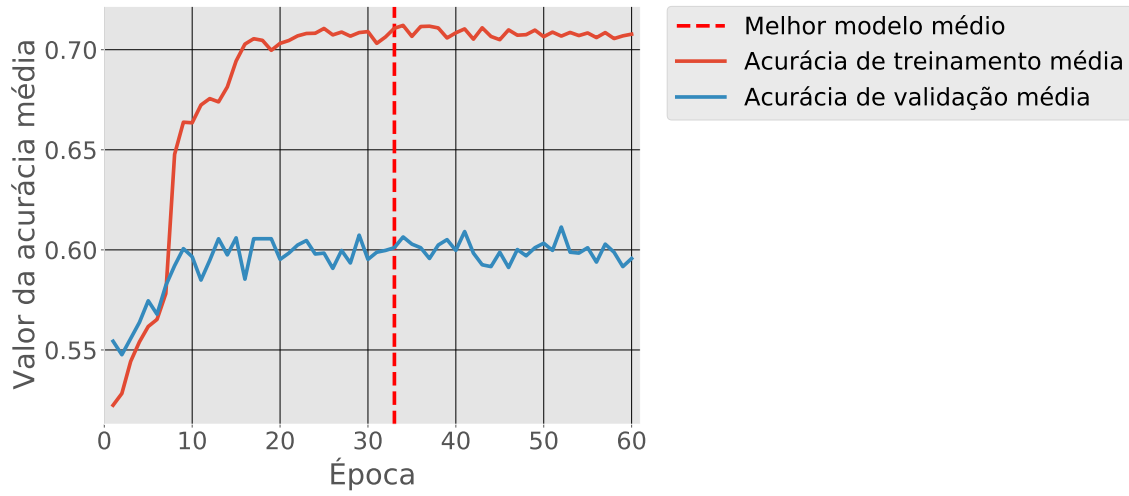
A realização da classificação foi utilizando o modelo proposto pelos autores He *et al.* (2016), o algoritmo utilizado está disponível em Pytorch (2019d).

Analisando os gráficos das Figuras 53 e 54, é possível chegar à conclusão que a média de acerto da validação dos modelos se estabiliza entre valores de 58,00% a 73,00%, isso com e sem a normalização. A máxima para a média da curva de validação do treinamento foi de 61,14%, sem normalização e 63,11%, com normalização. As curvas da acurácias médias e das perdas médias mostraram uma distância elevada. No gráfico médio da perda, apresentado nas Figuras 53b e 54b, a distância dos valores de treinamento e validação foi de 0,3470, isso sem normalização e 0,3279, com normalização, valores retirados para a época 33, sem e com normalização (valor obtido do gráfico médio da acurácia para o melhor modelo médio, apresentado nas 53a e 54a).

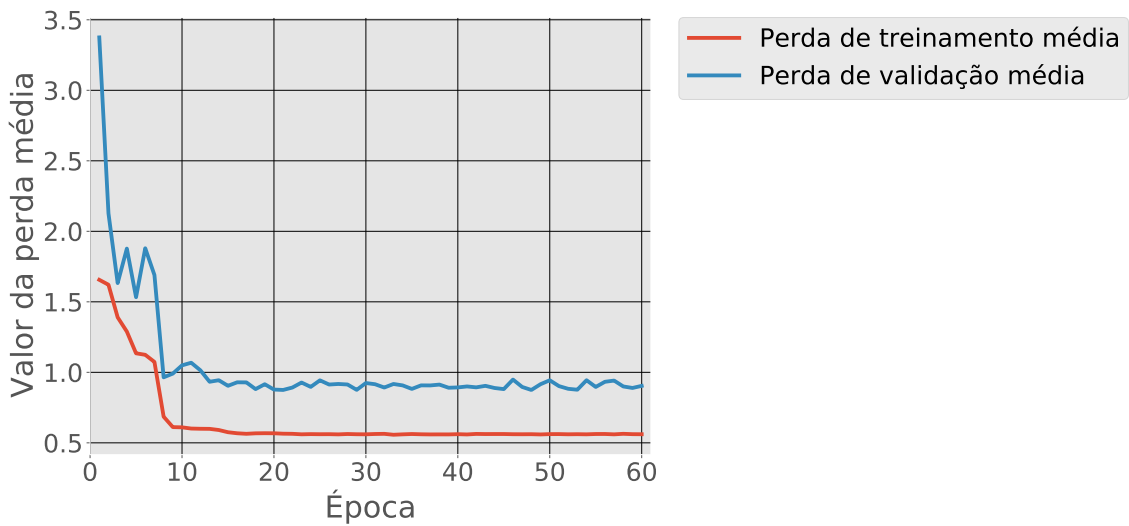
Quando isolamos o melhor modelo ele se encontra numa situação oposta aos dados médios, gráficos não são tão suavizados. Os gráficos das Figuras 55 e 56, foram obtidos do melhor modelo, em que obteve uma resposta correta no conjunto de teste e o maior valor de acerto para o conjunto de validação, em caso de empate foi analisando o menor valor de perda para aquele determinado momento de melhor validação. Para o gráfico da Figura 55, estrutura sem normalização, o valor máximo da acurácia que corresponde à época 46, para esse modelo foi de 85,71%. Sua distância na perda para essa época foi de 0,7862. Já para o gráfico da Figura 56, estrutura com normalização, o valor da acurácia da época 56 desse modelo foi de 90,47%. Sua distância na perda para essa época foi de 0,9497. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,262074 segundos sem normalização e 0,278390 segundos com normalização, já para a GPU foi de 0,023424 segundos sem normalização e 0,023852 segundos com normalização.

Figura 53 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet152, sem normalização.

(a) Acurácias médias.



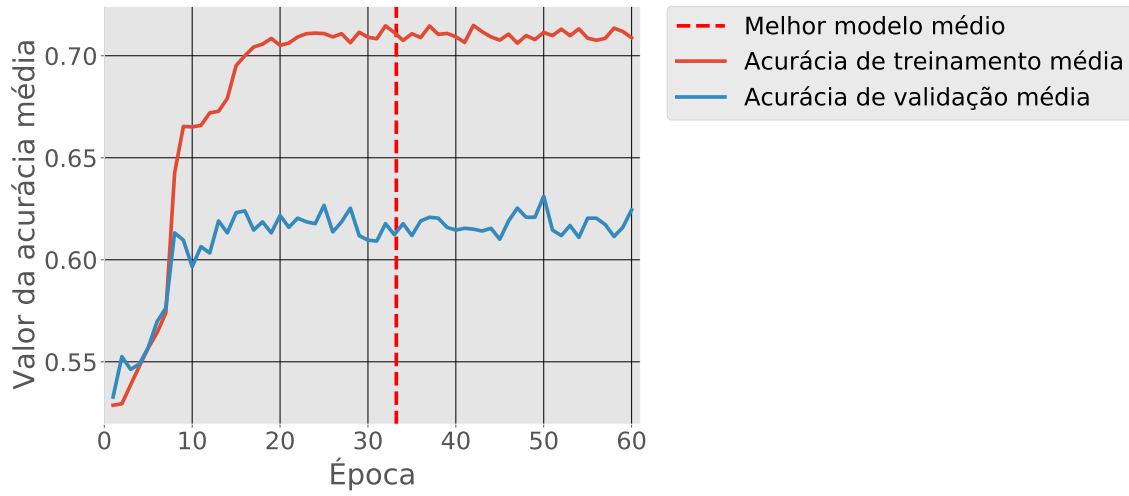
(b) Perdas médias.



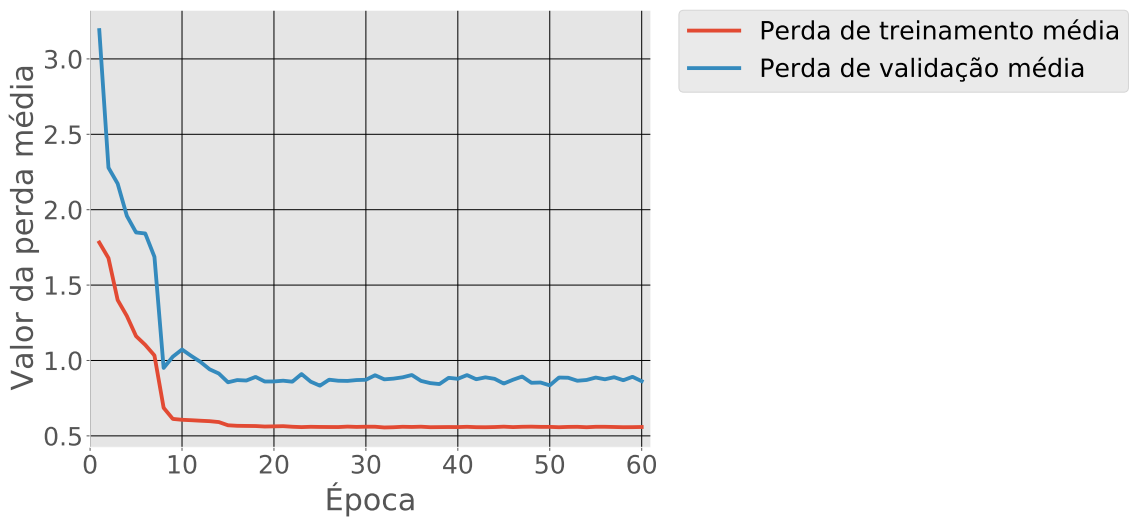
Fonte: o autor.

Figura 54 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNet152, com normalização.

(a) Acurácias médias.

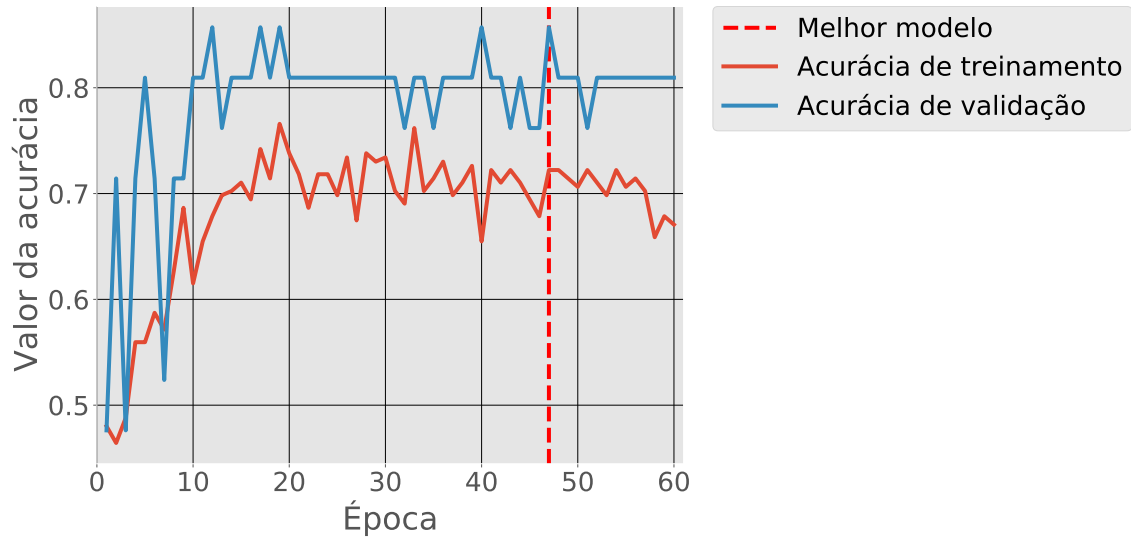


(b) Perdas médias.

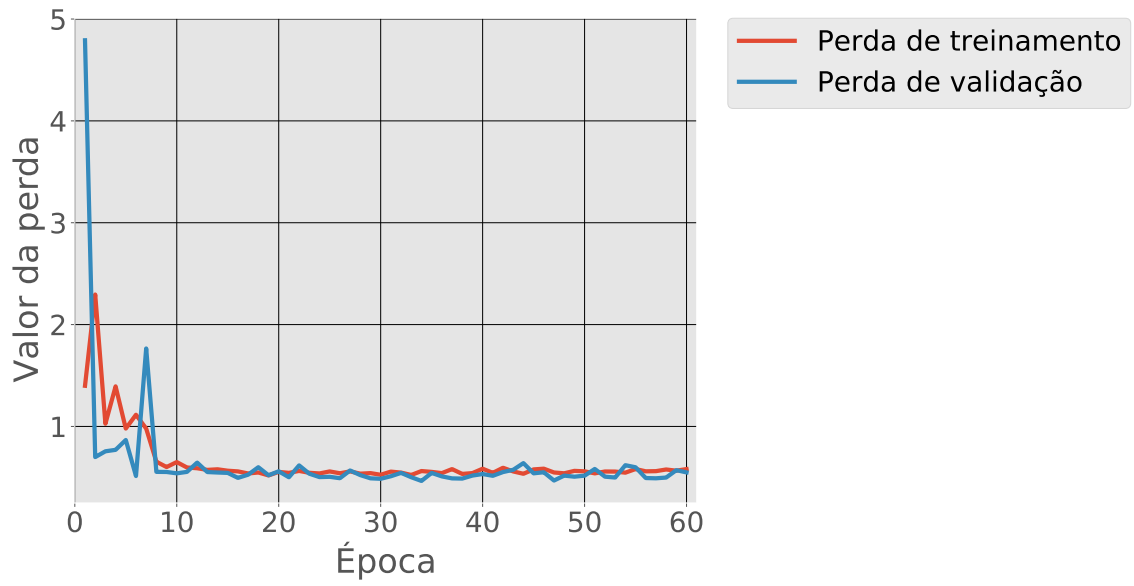


Fonte: o autor.

Figura 55 – Gráfico da acurácia e perda obtido do melhor modelo ResNet152, sem normalização.
(a) Acurácias.



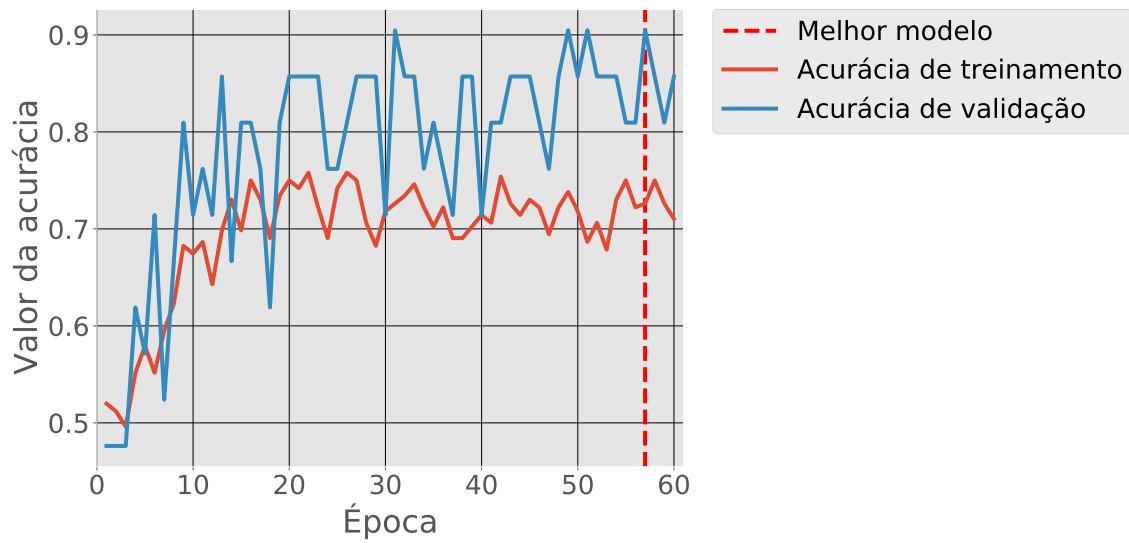
(b) Perdas.



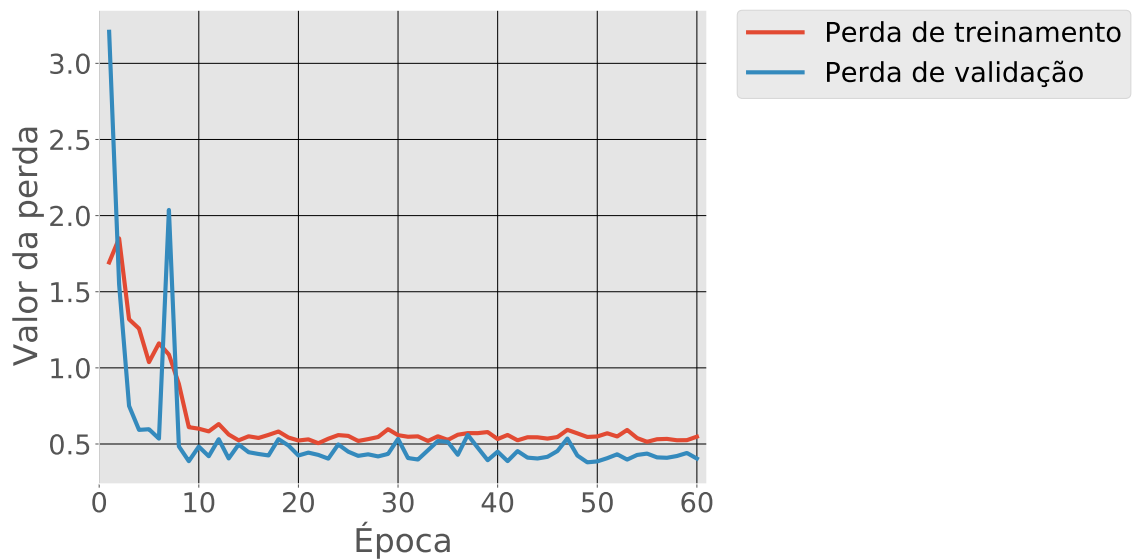
Fonte: o autor.

Figura 56 – Gráfico da acurácia e perda obtido do melhor modelo ResNet152, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

4.2.1.4 *Aggregated Residual Transformations for Deep Neural Networks (ResNeXt)*

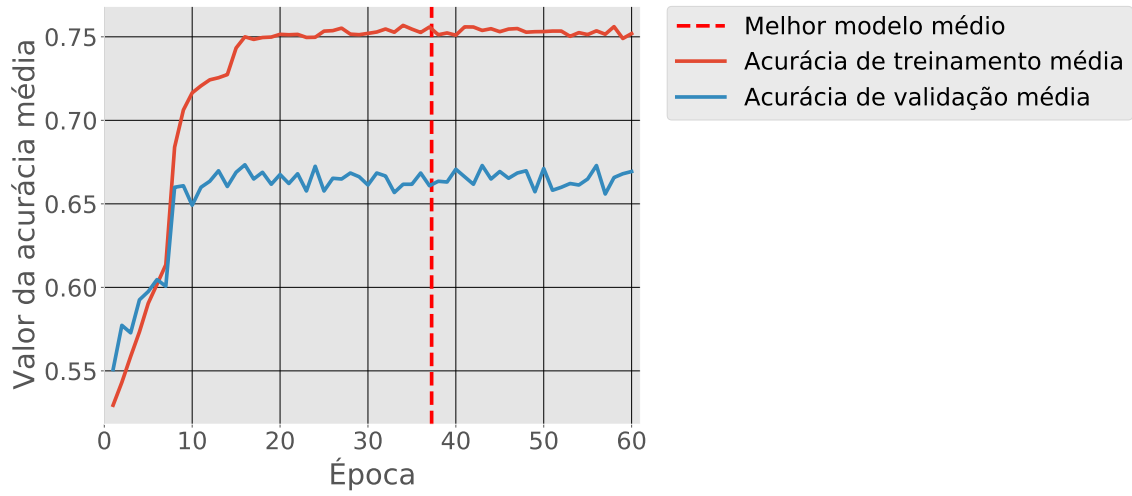
A realização desta função de classificação foi utilizado o modelo proposto pelos autores Xie *et al.* (2017), o algoritmo utilizado está disponível em Pytorch (2019e).

Analisando os gráficos das Figuras 57 e 58, é possível chegar à conclusão que a média de acerto da validação dos modelos se estabiliza entre valores de 65,00% a 75,00%, isso com e sem a normalização. A máxima para a média da curva de validação do treinamento foi de 67,34%, sem normalização e 68,82 %, com normalização. As curvas da acurácias médias e da perdas médias mostraram uma distância elevada. No gráfico médio da perda, apresentado nas Figuras 57b e 58b, a distância dos valores de treinamento e validação foi de 0,3164, isso sem normalização e 0,4244, com normalização, valores retirados para a época 37, sem normalização, e época 38, com normalização (valor obtido do gráfico médio da acurácia para o melhor modelo médio, apresentado nas 57a e 58a).

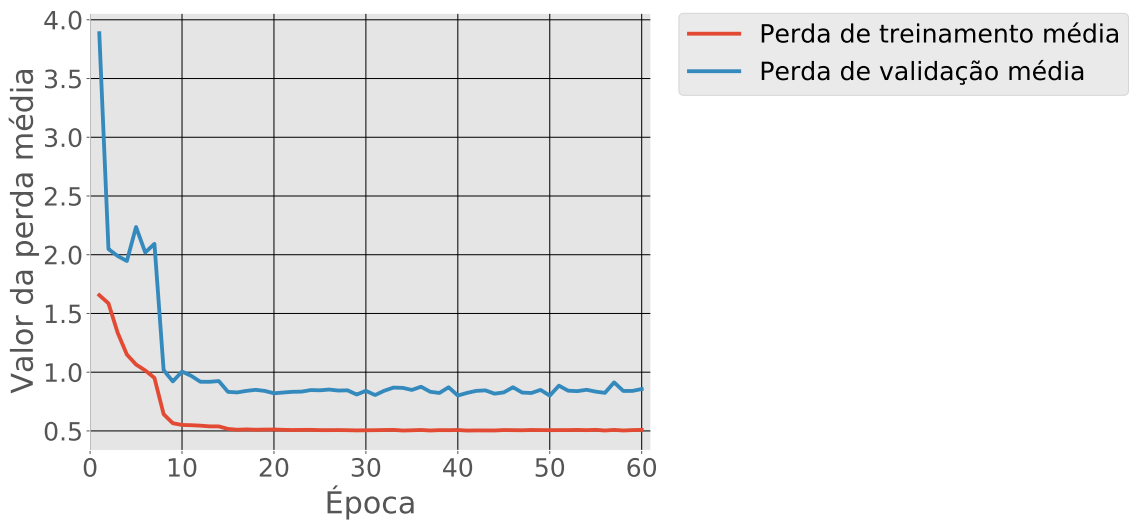
Quando isolamos o melhor modelo ele se encontra numa situação oposta aos dados médios, gráficos não são tão comportados. Os gráficos das Figuras 59 e 60, foram obtidos do melhor modelo, em que obteve uma resposta correta no conjunto de teste e o maior valor de acerto para o conjunto de validação, em caso de empate foi analisando o menor valor de perda para aquele determinado momento de melhor validação. Para o gráfico da Figura 59, estrutura sem normalização, o valor máximo da acurácia que corresponde à época 49, para esse modelo foi de 95,23%. Sua distância na perda para essa época foi de 0,1040. Já para o gráfico da Figura 60, estrutura com normalização, o valor da acurácia da época 6 desse modelo foi de 90,47%. Sua distância na perda para essa época foi de 0,9545. Também foi calculado o tempo de resposta do modelo para CPU, que foi de 0,439697 segundos sem normalização e 0,419737 segundos com normalização, já para a GPU foi de 0,042580 segundos sem normalização e 0,040185 segundos com normalização.

Figura 57 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNeXt, sem normalização.

(a) Acurácias médias.



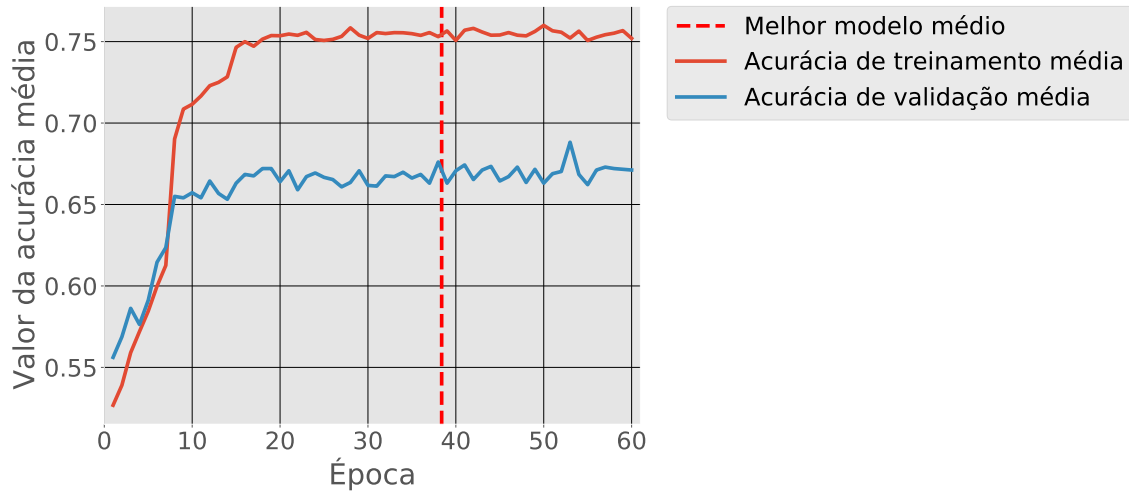
(b) Perdas médias.



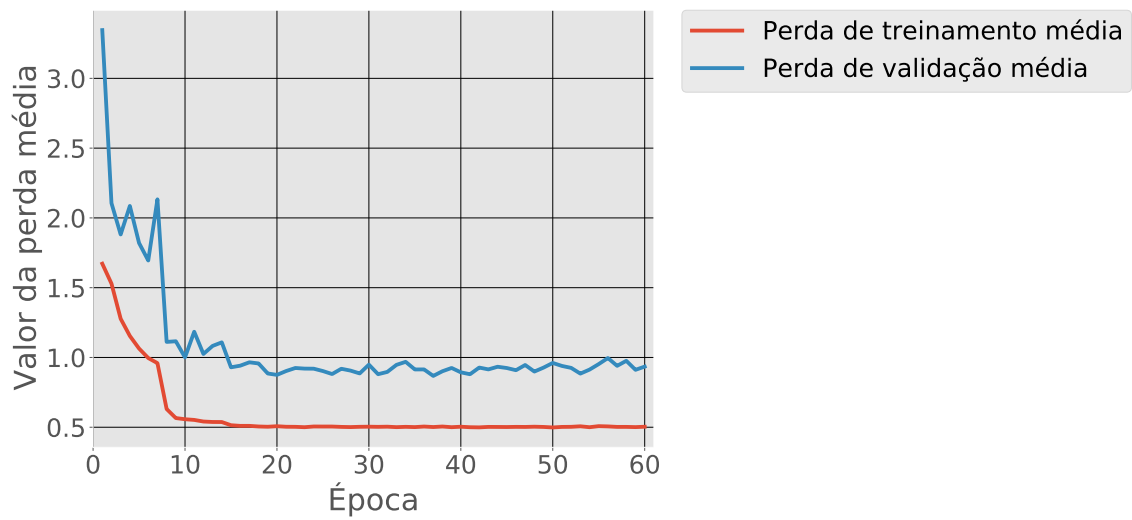
Fonte: o autor.

Figura 58 – Gráficos das acurácias e perdas médias obtidas nos modelos ResNeXt, com normalização.

(a) Acurácias médias.



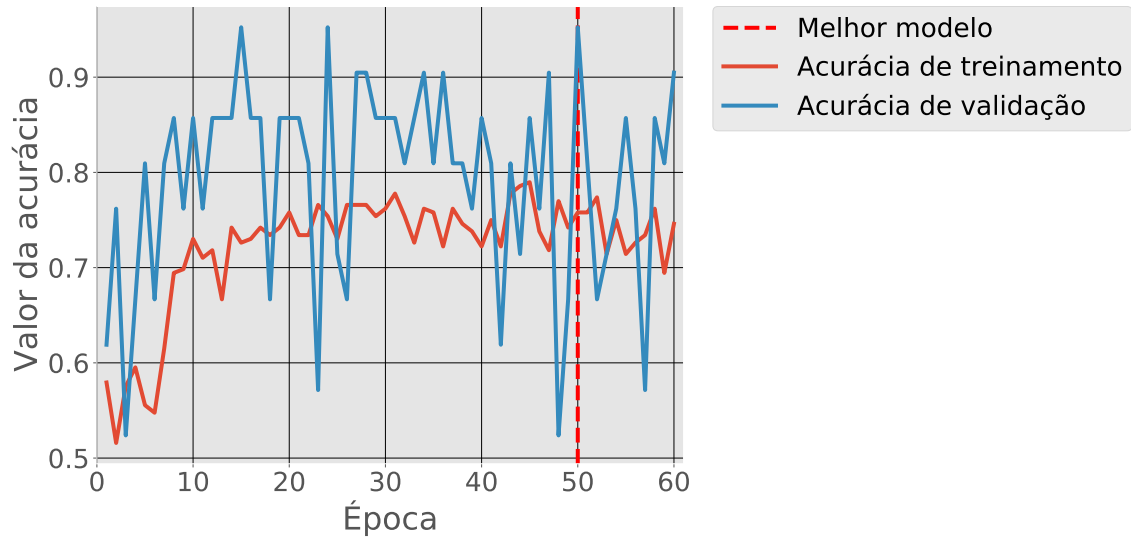
(b) Perdas médias.



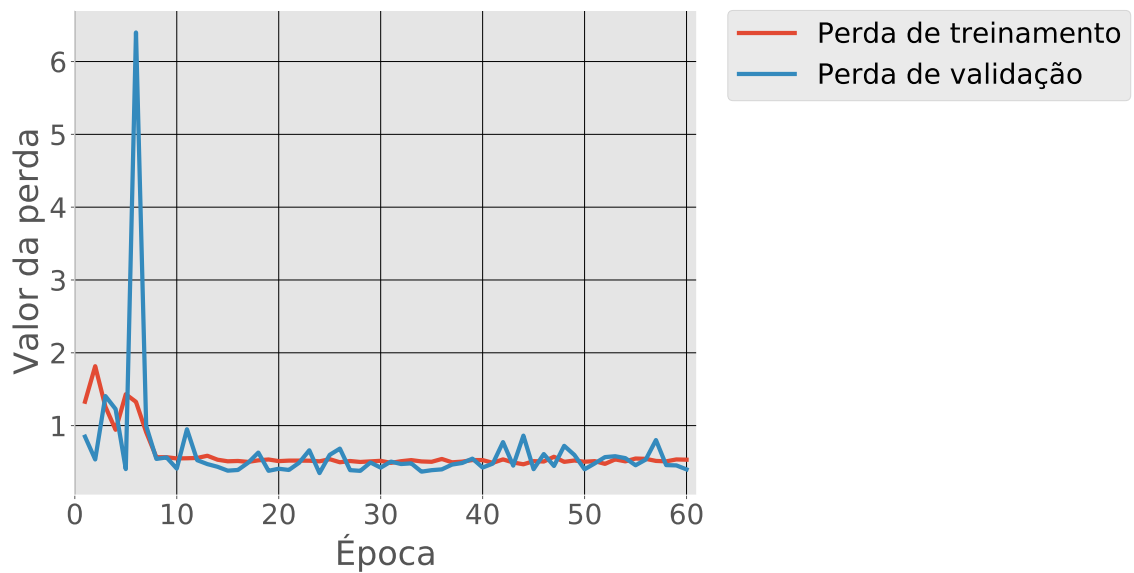
Fonte: o autor.

Figura 59 – Gráfico da acurácia e perda obtido do melhor modelo ResNeXt, sem normalização.

(a) Acurácias.



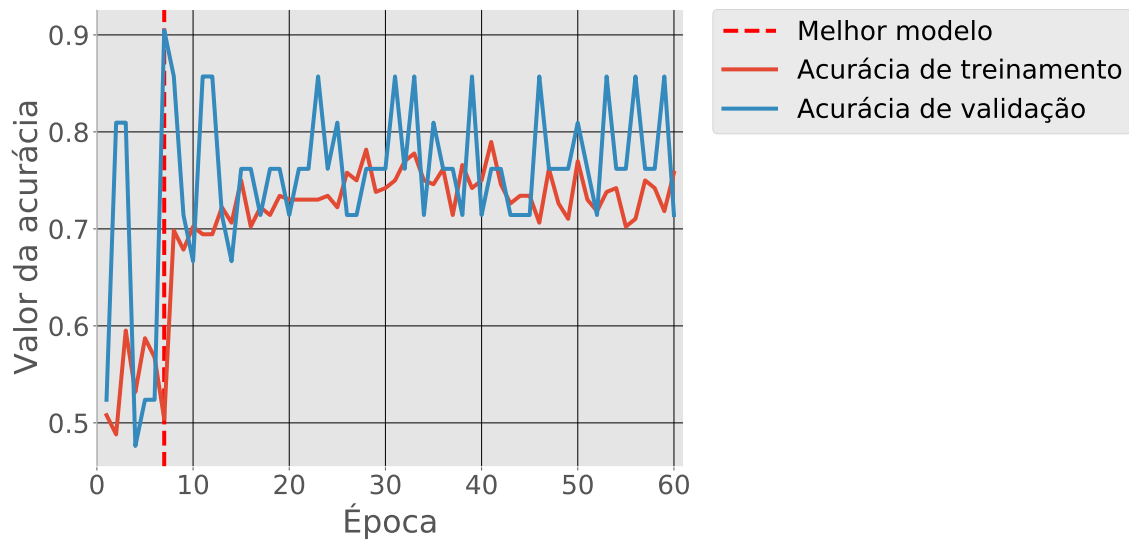
(b) Perdas.



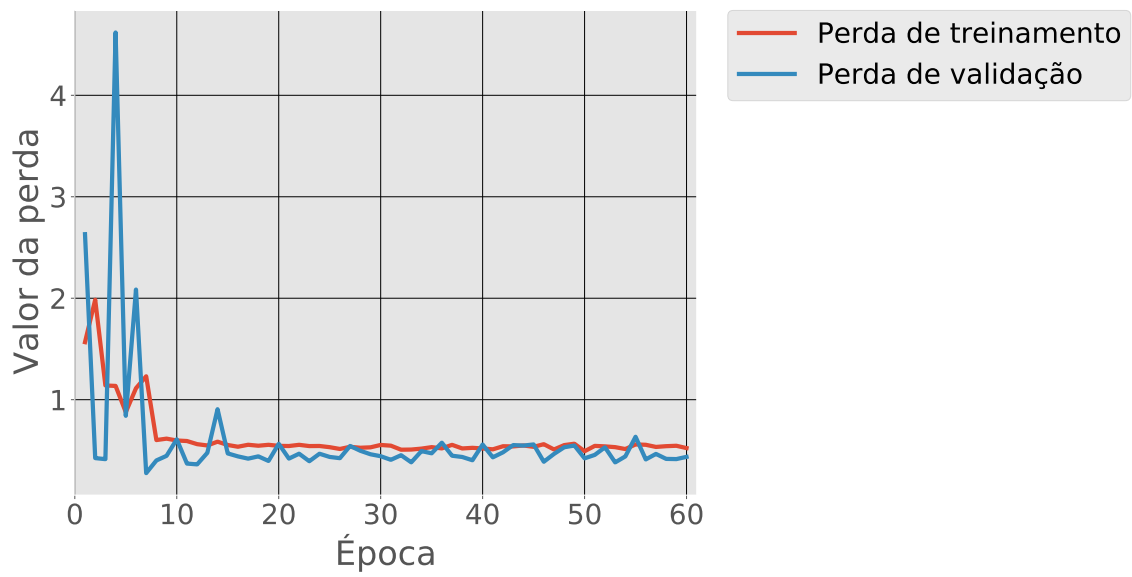
Fonte: o autor.

Figura 60 – Gráfico da acurácia e perda obtido do melhor modelo ResNeXt, com normalização.

(a) Acurácias.



(b) Perdas.



Fonte: o autor.

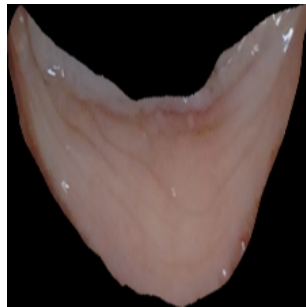
4.2.2 Aplicação em outra base de dados

Como foi apresentado na Subseção 4.1.2,

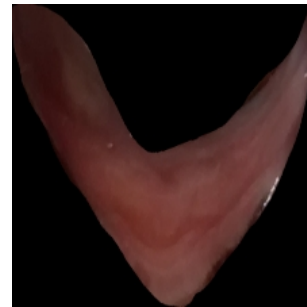
Foi aplicado nos modelos já treinados uma outra base de dados de imagens, como apresentado na Subseção 4.1.2. Na base de dados da UFMA, eles coletaram além das imagens, dados com Ht e o grau receptivo do FAMACHA declarado por um tratador. Utilizando as informações do Ht essa base de dados contém 78 imagens de animais saudáveis e 27 doentes. As imagens na Figura 61 são algumas que foram aplicadas no teste. A Figura 61a, tem um Ht de 24, então segundo o exame esse animal está saudável. O grau FAMACHA desse animal é 5°, portanto o animal está doente. A Figura 61b, tem um Ht de 16, conforme o exame esse animal está doente, para o tratador, o grau FAMACHA desse animal é 4°, então segundo esse método o animal está doente. As informações de saudável e doentes foram retiradas da Tabela 1.

Figura 61 – Imagens de teste da base da UFMA.

(a) Ht 24.



(b) Ht 16.



Fonte: o autor.

Tabela 6 – Resultados para a Figura 61a

Modelos	Sem normalização	Com normalização
Alexnet	Doente	Doente
MobileNetV2	Doente	Doente
ResNet18	Doente	Doente
ResNet152	Doente	Doente
ResNeXt	Doente	Doente

Fonte: o autor.

Tabela 7 – Resultados para a Figura 61b

Modelos	Sem normalização	Com normalização
Alexnet	Doente	Saudável
MobileNetV2	Doente	Saudável
ResNet18	Saudável	Doente
ResNet152	Saudável	Saudável
ResNeXt	Doente	Doente

Fonte: o autor.

As Tabelas 6 e 7 os respectivos resultados das imagens na Figura 61. Foi possível observar que na imagem da Figura 61a e os resultados dos modelos apresentado na Tabela 6 que a rede chegou a mesma conclusão que tratador. Em que pela coloração esbranquiçada da mucosa o animal, o mesmo poderia estar doente, porém pelo exame Ht o animal está saudável. Já para a imagem da Figura 61b e os resultados dos modelos apresentado na Tabela 6 demonstra resultados distintos sobre a mucosa do animal, provável essa confusão em função da mucosa tem relação com parte da coloração avermelhada em certos pontos da imagem.

A Tabela 8, contém os resultados estatísticos dos modelos, critérios apresentados na Seção 2.4.5.2. Esses dados são importantes, pois nem sempre um modelo que tem uma acurácia elevada é o melhor para uma aplicação em alta escala. Segundo essa tabela o melhor modelo foi a ResNet18, de acordo com a acurácia. Analisando a Precisão, *Recall* e *F1-Score*, é possível concluir que ele acertou mais as imagens saudáveis do que doente tornando o modelo tendencioso. O modelo que teve um resultado mais equilibrado foi MobileNetV2, sem normalização.

Tabela 8 – Estatísticas dos modelos.

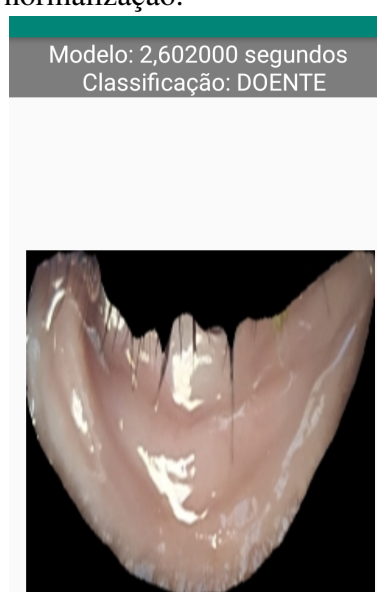
Modelos	Rótulos	Sem normalização				Com normalização			
		Acurácia	Precisão	Recall	F1-Score	Acurácia	Precisão	Recall	F1-Score
Alexnet	Saudável	40,95%	0,79	0,28	0,42	49,52%	0,76	0,47	0,58
	Doente		0,27	0,78	0,40		0,27	0,56	0,36
MobileNetV2	Saudável	56,19%	0,79	0,56	0,66	53,33%	0,75	0,56	0,64
	Doente		0,31	0,56	0,39		0,26	0,44	0,33
ResNet18	Saudável	64,76%	0,75	0,79	0,77	49,52%	0,80	0,42	0,55
	Doente		0,27	0,22	0,24		0,30	0,70	0,42
ResNet152	Saudável	59,05%	0,77	0,64	0,70	56,19%	0,75	0,62	0,68
	Doente		0,30	0,44	0,36		0,27	0,41	0,32
ResNeXt	Saudável	43,81%	0,74	0,37	0,50	49,52%	0,79	0,44	0,56
	Doente		0,26	0,63	0,37		0,29	0,67	0,40

Fonte: o autor.

4.2.3 Aplicação no celular

Os esclarecimentos discutidos na Subseção 4.1.3 podem também ser aplicados na parte da classificação. Quanto maior a complexidade da rede, maior é o tempo de processamento, como foi apresentado na Tabela 9. A Figura 62 é o resultado da aplicação do modelo no dispositivo *mobile* em *Android*. O tempo de processamento não se fez diferente entre com e sem normalização.

Figura 62 – Aplicação *mobile* do modelo gerado pela ResNeXt, sem normalização.



Fonte: o autor.

Tabela 9 – Tempo de resposta celular android.

Modelos	Tempo (s)	
	sem normalização	com normalização
Alexnet	0,121	0,121
MobileNetV2	0,175	0,174
ResNet18	0,244	0,247
ResNet152	1,807	1,716
ResNeXt	2,602	2,592

Fonte: o autor.

4.2.4 Análise dos resultados

No primeiro momento foi feita a escolha dos modelos justificando assim a sua aplicação:

- AlexNet é um algoritmo que ganhou destaque com um dos primordiais no conceito de aprendizagem profunda aplicada à classificação de imagem (KRIZHEVSKY, 2014; PYTORCH, 2019a);
- MobileNetV2 é um algoritmo que tem sua arquitetura voltada para aplicações *mobiles* (SANDLER *et al.*, 2018; PYTORCH, 2019b);
- ResNet é um algoritmo com amplas aplicações, que tem diversas modificações e derivações de camadas, por conta disso foi aplicado aqui duas dessa modificações ResNet18 e ResNet152 (HE *et al.*, 2016; PYTORCH, 2019c; PYTORCH, 2019d);
- ResNeXt dos modelos apresentado na biblioteca Pytorch (2019h) esse algoritmo foi o que teve melhor resultado (XIE *et al.*, 2017; PYTORCH, 2019e).

Para o comportamento dos gráficos de acurácia e perda é possível retirar informações relevantes. Para os gráficos dos modelos médios, apresentado nas Figuras 41, 42, 45, 46, 49, 50, 53, 54, 57 e 58 é possível ter uma análise maior dos algoritmos com os dados e seu comportamento chegando à conclusão que os modelos não obtiveram um bom desempenho, pois a curva de aprendizado do treinamento não chegou a valores de acurácia altos, por consequência da dificuldade em entender parte dos dados, isso também se mostrou na curva de validação. Além disso, a relação das curvas de aprendizado do treinamento e de validação apresentavam uma distância significativa. Analisando os gráficos médios dos modelos o que obteve maior faixa de valor foi a Resnet18, apresentada na Subseção 4.2.1.3.1, com uma faixa de valor entre 73,00% a 82,00%.

Já para os melhores modelos, apresentado nas Figuras 43, 44, 47, 48, 51, 52, 55, 56, 59 e 60, graficamente os resultados não são tão significativo, a acurácia do conjunto de treinamento não foi tão alta e seus valores de perdas não foram tão baixo, já para valores do conjunto de validação a sua curva ficou mais próximo da curva de treinamento mas com altos picos de acurácia obtendo valores de acerto de 90,00%, e as curvas de perdas do treinamento e validação estiveram mais próximas e mais estável, porém o seus valores não eram tão baixo, se comparados com os gráficos dos modelos apresentados na Seção 4.1.

Também foi aplicada uma normalização nos dados, como apresentado na Subseção 2.4.3, para o entendimento do seu comportamento. Pois com a normalização pode ocorrer um

efeito de elevar a acurácia dos dados. Não foi o caso para esses algoritmos, o aumento da acurácia do modelo no momento de validação e do teste não foram tão significativa e em outros instantes a normalização foi pior do que não normalizado, nesse caso para AlexNet e ResNeXt, o único modelo que teve resultados elevados com a normalização foi a ResNet152, como apresentado na Tabela 10.

Tabela 10 – Resultados obtidos nos testes para os melhores modelos.

Modelos	Acurácia do Modelo		Tempo CPU (s)		Tempo GPU (s)	
	SN	CN	SN	CN	SN	CN
Alexnet	95,23%	90,47%	0,027782	0,024549	0,001506	0,001276
MobileNetV2	95,23%	95,23%	0,034852	0,035160	0,001276	0,007482
ResNet18	95,23%	95,23%	0,052170	0,053361	0,005635	0,004359
ResNet152	85,71%	90,47%	0,262074	0,278390	0,023424	0,023852
ResNeXt	95,35%	90,47%	0,439697	0,419737	0,042580	0,040185

Fonte: o autor.

Legenda: SN: sem normalização; CN: com normalização.

Conforme a Tabela 10 também foram analisados a estrutura de tempo de resposta do modelo já treinado através da imagem de teste, isso é importante para saber se o modelo tem uma resposta rápida para uso da CPU e GPU. A normalização não afetou no desempenho de tempo como já era esperado, pois a normalização só afeta nos dados de entrada e não a quantidade de operações. Dependendo do modelo, quanto maior sua complexidade maior é o tempo na CPU e GPU. Em termos de velocidade, AlexNet mostrou-se mais ágil e ResNeXt mais lento

Para aplicação na Subseção 4.2.2, foi utilizado os melhores modelos já treinados obtidos para Alexnet, MobileNetV2, ResNet18, ResNet152 e ResNeXt, assim tendo um melhor entendimento do comportamento dos modelos para as imagens de outra base de dados. Através da Tabela 8, é possível chegar à conclusão que o melhor resultado segundo a acurácia foi a ResNet18, sem normalização, mas ela não obteve um resultado tão bom com a Precisão, *Recall* e *F1-Score*, provavelmente esse modelo se tornou bem criterioso, com a coloração da mucosa. Mesmo o animal estando doente, segundo o exame Ht, a mucosa poderia conter partes avermelhadas, então a classificação obtinha um resultado saudável ou então a mucosa estava totalmente branca e o animal estando saudável, segundo o exame Ht, então a classificação do animal resultava em doente.

Na Subseção 4.2.3 foi apresentado uma aplicação para analisar o tempo de resposta e o resultado usando um celular, como apresentado na Tabela 9 e na Figura 62. Essa parte se torna interessante pois é possível ter uma ideia de qual algoritmo é o mais rápido para realizar a

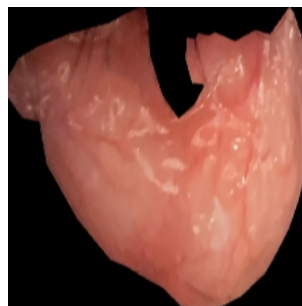
etapa de classificação em uma aplicação *mobile* e levar em conta sua acurácia. Para os modelos Alexnet, MobileNetV2, ResNet18 o tempo de resposta torna-os plausível para uma aplicação real voltada a um sistema embarcado, enquanto a ResNet152 e ResNeXt o tempo é mais elevado devido à complexidade da rede.

Figura 63 – Imagens usadas como conjunto de teste para criação dos modelos da ResNeXt, sem normalização.

(a) Teste acertado Ht 26.



(b) Teste errado Ht 20.



Fonte: o autor.

O erro na classificação das imagens da base de dados, no caso do rótulo não anêmica, pode ocorrer por conta da coloração da mucosa mais esbranquiçada, assim causando o engano. O processo oposto ocorre nas imagens com rótulo anêmica, assim a mucosa apresenta uma coloração mais avermelhada. Por exemplo, na Figura 63b, a imagem tinha um Ht de 20, segundo a Tabela 1, essa imagem estava anêmica, mas a resposta do modelo disse que essa imagem era não anêmica. Essa variância nas imagens no processo de classificação se torna evidente nos gráficos apresentados nessa seção, pois as suas curvas não conseguiram uma acurácia elevada se comparamos a seção anterior, Seção 4.1.

5 CONCLUSÃO

Neste capítulo serão apresentadas as conclusões do projeto, em seguida será feita uma discussão sobre os objetivos iniciais do trabalho.

5.1 Conclusões

Neste trabalho foi proposta desde o início a criação de uma base de dados ampla que leva em consideração a qualidade, quantidade e a precisão dos dados de imagem com laboratório, isso tudo para criação de uma metodologia de reconhecimento de anemia do animal em ovinos. Outro ponto importante foi o cuidado com a diversidade das imagens, que foram retiradas de várias câmeras, de ambientes diferentes, de formas distintas. Esses fatores foram primordiais para levantar uma metodologia de um sistema de segmentação e classificação que seja confiável para o pequeno e médio produtor. Ressaltamos que a base de dados deve ser aperfeiçoada caso sejam futuramente utilizadas na classificação da saúde através da imagem de ovinos.

Destarte, o método aplicado atingiu resultados significativos para a detecção de anemia. Podendo assim analisar os seus resultados em partes. A etapa de segmentação foi capaz de achar a região de interesse da mucosa mesmo com a presença do dedo do tratador em alguns imagens, a região de interesse posições distintas, a mudança do local de coleta de imagem. Isso com aplicação dos modelos: ERFNet, FCN, U-Net e DeepLabV3. Esses modelos treinados com a base de dados tiveram desempenho em torno de 90,00%, para critério IoU. Não ocorreram mudanças utilizando a normalização dos dados. Os resultados foram satisfatórios com aplicação de outra base de dados que teve um padrão de coleta diferente. É possível uma aplicação *mobile* utilizando modelo proposto para fazer a segmentação, analisando o tempo de resposta. Levando em considerando o tempo e valor do critério IoU, há dois modelos favoráveis a uma aplicação real, a U-Net e FCN. Até o dado momento, nenhum trabalho que proponha o reconhecimento da anemia dos animais utilizou uma segmentação automática utilizando aprendizagem profunda, passo promissor no estado da arte de segmentação automática da mucosa do animal.

A classificação alcançaram resultados de acurácia não tão elevados no processo para as duas classes, anêmico e não anêmico. Tal etapa ainda exige uma maior quantidade de dados para tornar o sistema confiável para o pequeno e médio produtor. Para tornar os modelos de classificação mais robustos e mais confiante nos resultados, faz-se necessário equilibrar as amostras de classes de animais doentes e saudáveis, aumentando o conjunto de treinamento

e validação. Ademais, é possível alcançar algumas conclusões a respeito dos modelos aqui utilizados, são eles: AlexNet, MobileNetV2, ResNet18, ResNet152 e ResNeXt. Com aplicação de outra base de dados, com um padrão de coleta diferente, concluímos que o melhor modelo corresponde a ResNet18, sem normalização, isso para valores de acurácia. Enquanto para precisão, *Recall* e *F1-Score*, levando em consideração um resultado com um menor desvio-padrão foi na MobileNetV2, sem e com normalização.

No decorrer da pesquisa, foi analisada a possibilidade de uma aplicação *mobile* utilizando modelo proposto para fazer a classificação, analisando o tempo de resposta. Considerando as métricas de tempo e acurácia, há dois modelos favoráveis, a uma aplicação *mobile*, ResNet18 (sem normalização) e MobileNetV2. A aplicação *mobile* ainda tem passos a serem seguidos para um modelo com precisão mais alta e confiabilidade na quantidade de imagens utilizadas no desenvolvimento, levando em consideração o estudo dos autores Macedo *et al.* (2014), que avaliaram a precisão com os valores de Ht, e a classificação pelo método FAMACHA, chegando a um resultado de 79,91 %. Um dos passos é o aumento de dados para os animais realmente doentes, grau 3° e grau 4°.

5.1.1 Verificação dos objetivos propostos

Este projeto foi realizado tendo como objetivos os tópicos expostos na seção 1.2. A reflexão abaixo descreve se os objetivos foram atingidos:

1. Para o primeiro objetivo, a criação da base de dados, todos os procedimentos necessários para tal criação foram descritos nas Seções 2.1, 2.2 e 2.3 e utilizando apenas uma fração desses dados para os testes, como descrito no Capítulo 4. Objetivo foi alcançado, ainda podendo melhorar a quantidade de imagens de animais realmente doentes para equilibrar com as imagens dos animais saudáveis;
2. Para o segundo objetivo de segmentação da mucosa alcançou resultados significativos, como descrito na Subseção 4.1.4. Com relação a presença do dedo do tratador com luva ou sem, as mudanças de cenários, diferentes pelagens dos animais, posição da mucosa em relação a imagem, as tonalidades diferentes de brilho nas imagens, dentre outros fatores, o modelo conseguiu identificar informações não relevantes e desconsiderá-las na imagem de saída. Ainda foi possível aplicar o modelo treinado em outra base de imagens, que também alcançou resultados satisfatório apesar da base de dados ser um pouco distinta da que foi treinado o modelo, como descrito na Subseção 4.1.2. Também desenvolvido

uma aplicação *mobile* para analisar a segmentação da imagem, que obteve um tempo de resposta adequado à complexidade dos modelos, como descrito na Subseção 4.1.3;

3. Para terceiro objetivo, o modelo utilizado com função de classificação alcançou resultados preliminares significativos, isso com as poucas imagens e com a busca do melhor modelo treinado, como descrito na Subseção 4.2.4. Foi possível testar os modelos treinados em outra base de dados de imagens, que alcançou resultados razoáveis, como descrito na Subseção 4.2.2. Na análise uma aplicação *mobile* para a classificação da imagem, obteve tempo razoável para uma espera do resultado, apesar da complexidade de alguns modelos, como descrito na Subseção 4.2.3.

5.2 Perspectivas futuras

Para pesquisas futuras, com intuito fundamentar as aplicações, sugerimos a continuidade do processo de coleta, a fim de obter mais imagens animais com grau de anemia mais elevado. Acredita-se que isso venha a melhorar a precisão do algoritmo de classificação, também a generalização do algoritmo de segmentação. Outro ponto importante é o desenvolvimento de uma aplicação *mobile* através do sistema operacional *Android*, que utilize os melhores modelos treinados de segmentação e classificação.

Suporte do Projeto e Informações Complementares

O projeto foi aprovado na Comissão de Ética no Uso de Animais (CEUA) da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) - Caprinos e Ovinos, com o cadastrado de protocolo 007/2017. Sua base de dados está ligada diretamente a esta sede da EMBRAPA, localizada na região de Sobral - CE, que ao dado momento não estão disponibilizando a sua distribuição pública. Mas os autores do projeto podem ser contactados para mais esclarecimentos. O projeto está dentro de uma colaboração com os pesquisadores Marcel Teixeira e Francisco Selmo Fernandes Alves da referida instituição.

REFERÊNCIAS

- ALEXANDRE, M. **ERFNet (PyTorch version)**. 2017. Disponível em: <<https://github.com/milesial/Pytorch-UNet/>>. Acesso em: 25 dez. 2020.
- AZEVEDO, M. de. **Hematologia Básica: Fisiopatologia e Diagnóstico Laboratorial**. Thieme Revinter, 2019. ISBN 9788554651381. Disponível em: <<https://books.google.com.br/books?id=4vWCDwAAQBAJ>>.
- BATH, G. F.; MALAN, F. S.; WYK, J. A. V. The "FAMACHA®" ovine anaemia guide to assist with the control of haemonchosis. p. 5, jun 1996.
- BATTULA, K. Research of machine learning algorithms using k-fold cross validation. **International Journal of Engineering and Advanced Technology**, v. 8, p. 215–218, 03 2021.
- BROWNLEE, J. **How to use Learning Curves to Diagnose Machine Learning Model Performance**. 2019. Disponível em: <<https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>>. Acesso em: 06 jan. 2021.
- CARLÔTO, P. Y. **Criação de caprinos e ovinos é destaque no sertão do Ceará**. 2018. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/19573-criacao-de-caprinos-e-ovinos-e-destaque-no-sertao-do-ceara>>. Acesso em: 26 nov. 2021.
- CHAGAS, A.; OLIVEIRA, M.; CARVALHO, C. O.; MOLENTO, M. Método FAMACHA®: Um recurso para o controle da verminose em ovinos. **Circular Técnica**, v. 52, p. 1–8, 01 2007.
- CHEN, L.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Rethinking atrous convolution for semantic image segmentation. **CoRR**, abs/1706.05587, 2017. Disponível em: <<http://arxiv.org/abs/1706.05587>>.
- CHRISTOPHER, M.; BELGHITH, A.; BOWD, C.; PROUDFOOT, J. A.; GOLDBAUM, M. H.; WEINREB, R. N.; GIRKIN, C. A.; LIEBMANN, J. M.; ZANGWILL, L. M. Performance of deep learning architectures and transfer learning for detecting glaucomatous optic neuropathy in fundus photographs. **Scientific Reports**, Springer Science and Business Media LLC, v. 8, n. 1, nov. 2018. Disponível em: <<https://doi.org/10.1038/s41598-018-35044-9>>.
- CLIMENI, B. S. O.; MONTEIRO, M. V.; CICOTI, C. A.; NEVES, M. F. Hemoncose ovina. **Revista Científica Eletrônica De Medicina Veterinária**, n. 11, jun 2008. ISSN 1679-7353.
- DAYYAL. **Macro Method (Wintrobe Method) For Estimation Of Packed Cell Volume (Pcv) Or Hematocrit**. 2017. Disponível em: <www.bioscience.com.pk/topics/hematology/item/781-macro-method-wintrobe-method-for-estimation-of-packed-cell-volume-pcv-or-hematocrit>. Acesso em: 23 dez. 2019.
- DEMOLINER, G.; ALVES, R. J. F. Anemímetro: app móvel para implementação do método famacha. **Unoesc & Ciência - ACET**, v. 8, n. 1, p. 25–32, jun. 2017. Disponível em: <<https://portalperiodicos.unoesc.edu.br/acet/article/view/12770>>.
- DIAS, M. V. **Utilização do cartão FAMACHA**. 2017. Disponível em: <<https://www.embrapa.br/busca-de-imagens/-/midia/3931001/utilizacao-do-cartao-famacha>>. Acesso em: 25 dez. 2019.

EROMERA. **ERFNet (PyTorch version)**. 2019. Disponível em: <https://github.com/Eromera/erfnet_pytorch>. Acesso em: 25 dez. 2020.

GONZALEZ, R.; WOODS, R. **Processamento Digital de Imagens**. 3. ed. São Paulo: Pearson Prentice Hall, 2010.

Gu, Y.; Lai, Y.; Xie, P.; Wei, J.; Lu, Y. Multi-scale prediction network for lung segmentation. In: **2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)**. [S.l.: s.n.], 2019. p. 438–442.

HE, K.; ZHANG, X.; S.REN; SUN, J. Deep residual learning for image recognition. p. 770–778, 2016.

HO, Y.; WOOKEY, S. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. **IEEE Access**, v. 8, p. 4806–4813, 2020.

IRUM, A.; AKRAM, M. U.; AYUB, S. W. S.; KHAN, M. Anemia detection using image processing. In: **Proceedings of International Conference on Digital Information Processing, Electronics, and Wireless Communications (DIPEWC2016)**. [S.l.: s.n.], 2016. p. 31–36.

JONES, M. T. **Arquiteturas de aprendizado profundo**. 2017. Disponível em: <<https://developer.ibm.com/br/technologies/deep-learning/articles/cc-machine-learning-deep-learning-architectures/>>. Acesso em: 02 fev. 2021.

KANG, P.; SINGH, A. K. **Data Augmentation with Keras ImageDataGenerator**. 2019. Disponível em: <<https://theailearner.com/2019/07/06/data-augmentation-with-keras-imagedatagenerator/>>. Acesso em: 07 fev. 2021.

KAPLAN, R.; BURKE, J.; TERRILL, T.; MILLER, J.; GETZ, W.; MOBINI, S.; VALENCIA, E.; WILLIAMS, M.; WILLIAMSON, L.; LARSEN, M.; VATTA, A. Validation of the famacha® eye color chart for detecting clinical anemia on sheep and goat farms in the southern united states. **Veterinary parasitology**, v. 123, p. 105–20, 09 2004.

KASVI. **Hematócrito: Como é realizado o exame e o que ele nos diz sobre nossa saúde**. 2019. Disponível em: <www.kasvi.com.br/hematocrito-exame-nossa-saude/>. Acesso em: 23 dez. 2019.

KHAN, A.; SOHAIL, A.; ZAHOORA, U.; QURESHI, A. S. A survey of the recent architectures of deep convolutional neural networks. **Artificial Intelligence Review**, Springer Science and Business Media LLC, v. 53, n. 8, p. 5455–5516, abr. 2020. Disponível em: <<https://doi.org/10.1007/s10462-020-09825-6>>.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2017.

KIZRAK, A. **Deep Learning for Image Segmentation: U-Net Architecture**. 2019. Disponível em: <<https://heartbeat.fritz.ai/deep-learning-for-image-segmentation-u-net-architecture-ff17f6e4c1cf>>. Acesso em: 13 fev. 2021.

KRIZHEVSKY, A. One weird trick for parallelizing convolutional neural networks. **CoRR**, abs/1404.5997, 2014. Disponível em: <<http://arxiv.org/abs/1404.5997>>.

LACVET. **HEMATÓCRITO**. 2018. Universidade Federal do Rio Grande do Sul: Laboratório de Análises Clínicas Veterinárias (LACVET). Disponível em: <www.ufrgs.br/lacvet/hematocrito.htm>. Acesso em: 23 dez. 2019.

LIU, S. Leave-p-out cross-validation test for uncertain verhulst-pearl model with imprecise observations. **IEEE Access**, PP, p. 1–1, 09 2019.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. 2019. Disponível em: <<https://openreview.net/forum?id=Bkg6RiCqY7>>.

MACEDO, F.; LOURENÇO, F.; SANTELLO, G.; NUNES, M.; MORA, N.; MEXIA, A. Accuracy of the famacha© method in ewes fed different levels of crude protein. **Acta Scientiarum - Animal Sciences**, v. 36, p. 209–214, 05 2014.

MATHWORKS. **What Deep Learning?** 2021. Disponível em: <<https://www.mathworks.com/discovery/deep-learning.html>>. Acesso em: 02 fev. 2021.

MCGLINCHY, J.; B. JOHNSON; MULLER, B.; JOSEPH, M.; DIAZ, J. Application of unet fully convolutional neural network to impervious surface segmentation in urban environment from high resolution satellite imagery. In: **IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium**. [S.l.: s.n.], 2019. p. 3915–3918.

MEDEIROS, T. **Criação de caprinos e ovinos é destaque no sertão do Ceará**. 2020. Disponível em: <<https://www.caprilvirtual.com.br/noticias.php?recordID=8773>>. Acesso em: 04 mar. 2021.

MOORS, E.; GAULY, M. Is the FAMACHA© chart suitable for every breed? correlations between FAMACHA© scores and different traits of mucosa colour in naturally parasite infected sheep breeds. **Veterinary Parasitology**, v. 166, n. 1, p. 108 – 111, 2009. ISSN 0304-4017. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304401709004646>>.

NATHANIEL, J. **Understanding PyTorch Loss Functions: The Maths and Algorithms (Part 1)**. 2021. Disponível em: <<https://towardsdatascience.com/understanding-pytorch-loss-functions-the-maths-and-algorithms-part-1-6e439b27117e>>. Acesso em: 13 fev. 2021.

Nguyen, P. X.; Lu, Z.; Huang, W.; Huang, S.; Katsuki, A.; Lin, Z. Medical image segmentation with stochastic aggregated loss in a unified u-net. In: **2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)**. [S.l.: s.n.], 2019. p. 1–4.

PETERSSON, K.; KASS, E. **Online FAMACHA© Certification**. 2016. URI Small Ruminant Parasite Control Program. Disponível em: <<https://web.uri.edu/sheepngoat/famacha/>>. Acesso em: 17 out. 2019.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.ALEXNET**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/alexnet.html#alexnet>. Acesso em: 25 dez. 2020.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.MOBILENET**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/mobilenet.html#mobilenet_v2>. Acesso em: 28 dez. 2020.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.RESNET**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet18>. Acesso em: 28 dez. 2020.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.RESNET**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet152>. Acesso em: 28 dez. 2020.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.RESNET**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnext101_32x8d>. Acesso em: 01 jan. 2021.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.SEGMENTATION.SEGMENTATION**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/segmentation/segmentation.html#fcn_resnet50>. Acesso em: 25 dez. 2020.

PYTORCH. **SOURCE CODE FOR TORCHVISION.MODELS.SEGMENTATION.SEGMENTATION**. 2019. Disponível em: <https://pytorch.org/docs/stable/_modules/torchvision/models/segmentation/segmentation.html#deeplabv3_resnet50>. Acesso em: 25 dez. 2020.

PYTORCH. **TORCHVISION.MODELS**. 2019. Disponível em: <<https://pytorch.org/docs/stable/torchvision/models.html>>. Acesso em: 23 jan. 2021.

ROMERA, E.; ÁLVAREZ, J. M.; BERGASA, L. M.; ARROYO, R. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 1, p. 263–272, 2018.

RONNEBERGER, O.; FISCHER, P.; BROX, T. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. Springer International Publishing, 2015. 234–241 p. Disponível em: <https://doi.org/10.1007/978-3-319-24574-4_28>.

ROSEBROCK, A. **Keras ImageDataGenerator and Data Augmentation**. 2019. Disponível em: <<https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/>>. Acesso em: 07 fev. 2021.

ROSEBROCK, A. **Intersection over Union (IoU) for object detection**. 2020. Disponível em: <<https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>>. Acesso em: 15 set 2020.

RUSSELL, S. J.; NORVIG, P.; DAVIS, E. **Artificial Intelligence: A Modern Approach**. Prentice Hall, 2010. (Prentice Hall series in artificial intelligence). ISBN 9780136042594. Disponível em: <<https://books.google.com.br/books?id=8jZBksh-bUMC>>.

SACHAN, A. **Quick complete Tensorflow tutorial to understand and run Alexnet, VGG, Inceptionv3, Resnet and squeezeNet networks**. 2017. Disponível em: <<https://cv-tricks.com/tensorflow-tutorial/understanding-alexnet-resnet-squeezenetand-running-on-tensorflow/>>. Acesso em: 06 fev. 2021.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L. Mobilenetv2: Inverted residuals and linear bottlenecks. p. 4510–4520, 2018.

- SHELHAMER, E.; LONG, J.; DARRELL, T. Fully convolutional networks for semantic segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, p. 1–1, 05 2016.
- SHORTEN, C.; KHOSHGOFTAAR, T. A survey on image data augmentation for deep learning. **Journal of Big Data**, v. 6, 07 2019.
- STEVENS, E.; ANTIGA, L.; VIEHMANN, T. **Deep Learning with PyTorch**. Manning Publications, 2020. ISBN 9781617295263. Disponível em: <<https://books.google.com.br/books?id=89BlwwEACAAJ>>.
- SULTANA, F.; SUFIAN, A.; DUTTA, P. Evolution of image segmentation using deep convolutional neural network: A survey. **Knowledge-Based Systems**, v. 201-202, p. 106062, 2020. ISSN 0950-7051. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705120303464>>.
- SUTSKEVER, I.; MARTENS, J.; DAHL, G. E.; HINTON, G. E. On the importance of initialization and momentum in deep learning. In: . JMLR.org, 2013. v. 28, p. 1139–1147. Disponível em: <<http://dblp.uni-trier.de/db/conf/icml/icml2013.html#SutskeverMDH13>>.
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. [S.l.]: Springer London, 2010. (Texts in Computer Science). ISBN 9781848829350.
- TAMIR, A.; JAHAN, C. S.; SAIF, M. S.; ZAMAN, S. U.; ISLAM, M. M.; KHAN, A. I.; FATTAH, S. A.; SHAHNAZ, C. Detection of anemia from image of the anterior conjunctiva of the eye by image processing and thresholding. In: **2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)**. [S.l.: s.n.], 2017. p. 697–701.
- VAN WYK, J. A.; BATH, G. F. The FAMACHA© system for managing haemonchosis in sheep and goats by clinically identifying individual animals for treatment. **Veterinary research**, v. 33, p. 509–529, 09 2002.
- VAN WYK, J. A.; MALAN, F. S.; BATH, G. F. Rampant anthelmintic resistance in sheep in south africa - what are the options? In: **MANAGING ANTHELMINTIC RESISTANCE IN ENDOPARASITES**. [S.l.]: Conference of the World Association for the Advancement of Veterinary Parasitology (WAAVP), 1997. p. 51–63.
- VOULODIMOS, A.; PROTOPAPADAKIS, E.; KATSAMENIS, I.; DOULAMIS, A.; DOULAMIS, N. **Deep learning models for COVID-19 infected area segmentation in CT images**. 2020.
- WEISS, D. J.; WARDROP, K. J. **Schalm's Veterinary Hematology**. Wiley, 2011. ISBN 9780470961834. Disponível em: <<https://books.google.com.br/books?id=lQXtQn593F4C>>.
- XIE, S.; GIRSHICK, R.; DOLLAR, P.; TU, Z.; HE, K. Aggregated residual transformations for deep neural networks. p. 5987–5995, 07 2017.
- ZHANG, M.; WANG, P.; ZHANG, X. **Vehicle Color Recognition Using Deep Convolutional Neural Networks**. ACM, 2019. Disponível em: <<https://doi.org/10.1145/3349341.3349408>>.
- ZHOU, Z.; SIDDIQUEE, M. M. R.; TAJBAKSH, N.; LIANG, J. Unet++: A nested u-net architecture for medical image segmentation. In: . [S.l.: s.n.], 2018.