



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
ENGENHARIA ELÉTRICA - ELETRÔNICA

**Detecção de Anormalidades em Imagens de endoscopia do trato inferior
utilizando EfficientNets e Class Activation Mapping (CAM) para
segmentação de pólipos**

Lucas Modesto Torres

MANAUS-AM

2023

Lucas Modesto Torres

Detecção de Anormalidades em Imagens de endoscopia do trato inferior utilizando
EfficientNets e Class Activation Mapping (CAM) para segmentação de pólipos

Monografia apresentada à Coordenação do
Curso de Engenharia Elétrica da Universi-
dade Federal do Amazonas, como parte dos
requisitos necessários à obtenção do título de
Engenheiro Eletricista.

Orientador: Prof. Dr. Cícero Ferreira Fernandes Costa

MANAUS-AM

2023

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

T693d Torres, Lucas Modesto
Detecção de Anormalidades em Imagens de endoscopia do trato inferior utilizando EfficientNets e Class Activation Mapping (CAM) para segmentação de pólipos / Lucas Modesto Torres . 2023
35 f.: il. color; 31 cm.

Orientador: Cícero Ferreira Fernandes Costa
TCC de Graduação (Engenharia Elétrica - Eletrônica) -
Universidade Federal do Amazonas.

1. Endoscopia. 2. Cnn. 3. EfficientNet. 4. Cam. 5. Classificação Multiclasse. I. Costa, Cícero Ferreira Fernandes. II. Universidade Federal do Amazonas III. Título

Lucas Modesto Torres

Detecção de Anormalidades em Imagens de endoscopia do trato inferior utilizando EfficientNets e Class Activation Mapping (CAM) para segmentação de pólipos

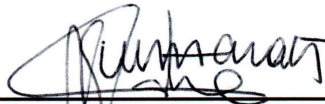
Monografia apresentada à Coordenação do Curso de Engenharia Elétrica da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Engenheiro Eletricista.

Aprovado em 08 de novembro de 2023

Banca Examinadora:



Prof. Dr. Cícero Ferreira Fernandes Costa
Universidade Federal do Amazonas - UFAM



Prof^a. Dr^a. Marly Guimarães Fernandes Costa
Universidade Federal do Amazonas - UFAM



Prof^a. MSc. Anne de Souza Oliveira
Universidade Federal do Amazonas - UFAM

MANAUS-AM
2023

Agradecimentos

Gostaria de expressar minha sincera gratidão a todas as pessoas que me apoiaram ao longo desta jornada acadêmica.

Primeiramente gostaria de agradecer aos meus pais, Alberto e Gilmara, assim como meus avós, Iran e Cisa, por seu apoio e orientação. Agradeço também meus irmãos, Gustavo e Melissa, e meus primos, David e Daniel, pela inspiração e pelos momentos de diversão.

Agradeço aos meus amigos por compartilharmos momentos de alegria e tristeza, Kha-leb, Daniel, Renata e Mayra. Agradeço também meus colegas que ajudaram a mapear os experimentos desse trabalho, Iago, Eneida e Otavio.

Agradeço a todos os professores, em especial prof. Dr. Cícero, prof^a Marly e Anne, por terem compartilhado seu conhecimento e orientações para que conseguisse me desenvolver na área. E agradeço à UFAM por ter sido um ambiente provedor de aprendizados.

*”Nós só podemos ver um pouco do futuro,
mas o suficiente para perceber que há muito
a fazer.”*

(Alan Turing)

Resumo

As doenças relacionadas ao trato gastrointestinal (TG) estão cada vez mais frequentes e afetam a vida de milhões de pessoas, representando uma incidência de 3,5 milhões de casos de acordo com a Organização Mundial da Saúde (OMS) para o ano de 2023. No mundo médico, os exames de endoscopia são amplamente utilizados para diagnóstico das doenças do TG. Com base nisso, este trabalho apresenta uma solução usando a rede EfficientNet-B1 para o diagnóstico automático de imagens do TG (normais, pólipos e úlceras) e a investigação do uso do CAM (mapas de ativação por classe) para segmentação da doença pólipos. Além disso, investiga o uso do aumento de dados e compara com o experimento sem aumento usando métricas e matrizes de confusão.

Key-words: Endoscopia, CNN, EfficientNet, CAM, Classificação Multiclasse.

Abstract

Diseases related to the gastrointestinal tract (GT) are increasingly common and affect the lives of millions of people, representing an incidence of 3.5 million cases according to the World Health Organization (WHO) for the year 2023. In the medical world, endoscopy exams are widely used to diagnose GT diseases. This work presents a solution using the EfficientNet-B1 for automatic diagnosis of GT images (normal, polyps, ulcerative) and the investigation of the use of CAM (Class Activation Mapping) for polyps disease segmentation. Furthermore, test the use of data augmentation and compare with experiments without, using metrics and confusion matrix.

Key-words: Endoscopy, CNN, EfficientNet, CAM, Multiclass classification.

Lista de Figuras

2.1	Funcionamento de um neurônio em uma camada densa.	3
2.2	Funcionamento de uma camada densa.	3
2.3	Funcionamento da função ReLU. Fonte: (Liu, 2017)	4
2.4	Funcionamento da convolução. Fonte: (Rangel, 2020)	5
2.5	Acurácia x número de parâmetros das EfficientNets em relação aos modelos padrões de CNNs. Fonte:(Tan and Le, 2020)	8
2.6	Arquitetura da EfficientNet-B1. Fonte:(Agarwal, 2020)	9
2.7	Representação visual da IoU. Fonte:(Rosebrock, 2016)	12
2.8	Imagem (a): original. Imagem (b): cachorro. Imagem (c): gato. Fonte:(Selvaraju et al., 2019)	12
3.1	Diagrama dos experimentos	13
3.2	Imagem (a): original. Superior (b): rotação em 30°. Superior (c): rotação em 125°. Meio (d): reflexão eixo y. Meio (e): cisalhamento	15
3.3	Imagem (a): imagem a ser segmentada. Imagem (b): máscara anotada disponível no banco de dados (Ground Truth)	17
3.4	Sumário do modelo Keras da EfficientNet-B1	18
3.5	Diagrama detalhando processo do CAM	19
3.6	Diagrama detalhando o processo de segmentação usando saída do CAM	20
4.1	Gráficos de treino experimento 1. Superior: acurácia. Inferior: perda.	22
4.2	Matriz confusão referente ao experimento de classificação sem aumento de dados	23
4.3	Exemplos da obtenção do heatmap no experimento 1. Esquerda: amostra. Meio: região da doença. Direita: heatmap.	24

4.4	Resultados de segmentação no experimento 1. Esquerda: amostra. Meio: Ground Truth. Direita: Inferência.	25
4.5	Gráfico ilustrando a distribuição dos valores para métrica IoU no experimento 1.	26
4.6	Gráficos de treino experimento 2. Superior: acurácia. Inferior: perda. . . .	27
4.7	Matriz confusão referente ao experimento 2	28
4.8	Exemplos da obtenção do heatmap no experimento sem aumento de dados. Esquerda: amostra. Meio: região da doença. Direita: heatmap.	29
4.9	Resultados de segmentação no experimento sem aumento de dados. Esquerda: amostra. Meio: Ground Truth. Direita: Inferência.	30
4.10	Gráfico ilustrando a distribuição dos valores para métrica IoU no experimento 2	31

Lista de Tabelas

3.1 Tabela referente ao banco de dados HyperKvasir com suas classes e número de amostras	16
--	----

Sumário

1	Introdução	1
1.1	Objetivo Geral	1
1.2	Objetivos Específicos	1
2	Fundamentos Teóricos	2
2.1	Conceitos Básicos de Aprendizado de Máquina e Redes Neurais	2
2.1.1	Camadas densas	2
2.1.2	Funções de ativação	3
2.1.2.1	ReLU	4
2.1.2.2	Softmax	4
2.1.3	Convoluções	4
2.1.3.1	Agrupamento médio e máximo (max and average pooling)	6
2.1.3.2	Normalização	6
2.1.3.3	Linearização (Flatten)	6
2.1.4	Função de perda	6
2.1.4.1	Entropia cruzada (Cross Entropy)	7
2.1.5	Otimizadores	7
2.2	EfficientNets	7
2.2.1	EfficientNet-B1	8
2.3	Métricas	9
2.3.1	Matriz confusão	9
2.3.2	Acurácia	10
2.3.3	Precisão	10
2.3.4	Sensibilidade (Recall)	11
2.3.5	F1-score	11

2.3.6	Interseção sobre a união (Intersection Over Union - IoU)	12
2.4	Grad-CAM (Class Activation Mapping baseado em Gradientes)	12
3	Metodologia	13
3.1	HyperKvasir para classificação (a)	14
3.2	HyperKvasir para segmentação de pólipos (b)	16
3.3	EfficientNet-B1 (c)	17
3.4	Treinamento (d)	18
3.5	Métricas (e e g)	18
3.6	CAM para segmentação (f)	19
4	Resultados	21
4.1	Experimento 1 (sem aumento de dados)	21
4.1.1	Classificação	21
4.1.2	Segmentação	23
4.2	Experimento 2 (com aumento de dados)	26
4.2.1	Classificação	26
4.2.2	Segmentação	28
5	Conclusão	32
	Referências Bibliográficas	33
	Códigos Utilizados	35
.1	CAM	35

Capítulo 1

Introdução

As principais causas de mortalidade relacionadas ao câncer estão localizadas no trato gastrointestinal (TG)([Mohapatra et al., 2023](#)), que correspondem a um quarto dos casos de câncer e um terço das mortalidades TG([Torres et al., 2023](#)). Como o diagnóstico antecipado dessas doenças impacta de forma positiva no tratamento e recuperação do paciente e muitas vezes não há profissionais suficientes para atender a alta demanda, um processo automático para ajudar no diagnóstico contribuiria para a redução na taxa de mortalidade dessa doença. Portanto, o atual trabalho entra com o propósito de apresentar uma solução para detecção automática de câncer do TG usando técnicas e arquiteturas de aprendizado profundo.

1.1 Objetivo Geral

Classificar as doenças do trato inferior usando aprendizado profundo e investigar o uso de uma técnica CAM (mapas de ativação por classe) para tarefas de segmentação.

1.2 Objetivos Específicos

- Investigar o uso da rede selecionada para as tarefas de classificação do TG;
- Verificar os mapas de calor usando CAM para validar os modelos treinados;
- Testar a tarefa de segmentação usando o mapa de calor gerado pelo método CAM;

Capítulo 2

Fundamentos Teóricos

2.1 Conceitos Básicos de Aprendizado de Máquina e Redes Neurais

Aprendizado de máquina é um subcampo de inteligência artificial (IA) focado na criação de modelos capazes de aprender uma determinada tarefa por meio de um banco de dados. Seu desempenho está completamente interligado às amostras, função de custo, otimzadores e arquitetura adotada. Os subtópicos a seguir ilustrarão cada um desses pontos destacados para contextualizar o atual trabalho.

2.1.1 Camadas densas

As camadas densas são compostas por neurônios e, para a solução proposta, é composto pelo número de classes adotada pela problemática. Isso se deve ao fato de a camada densa ter sido usada apenas no final da arquitetura adotada (EfficientNet-B1) para retornar a probabilidade para cada uma das classes usando a função de ativação Softmax (detalhada nos subtópicos adiante). A imagem [2.1](#) ilustra o funcionamento de um neurônio e a imagem [2.2](#) ilustra como é usada no final da rede EfficientNet-B1 para classificação de Normal, Pólipos e Úlceras.

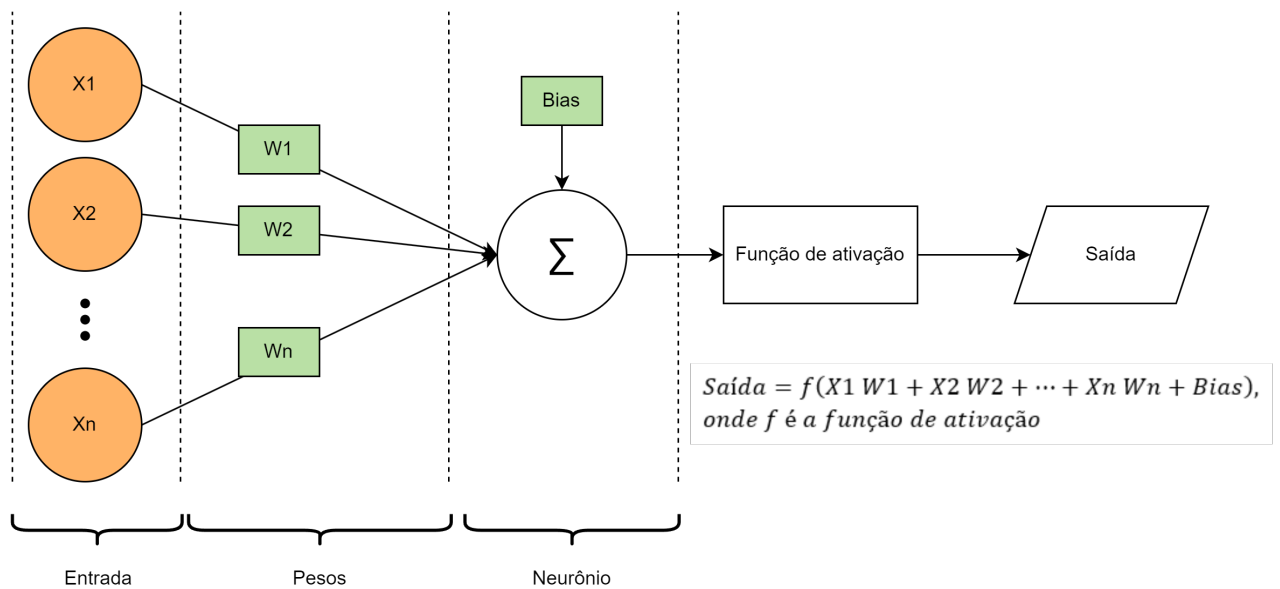


Figura 2.1: Funcionamento de um neurônio em uma camada densa.

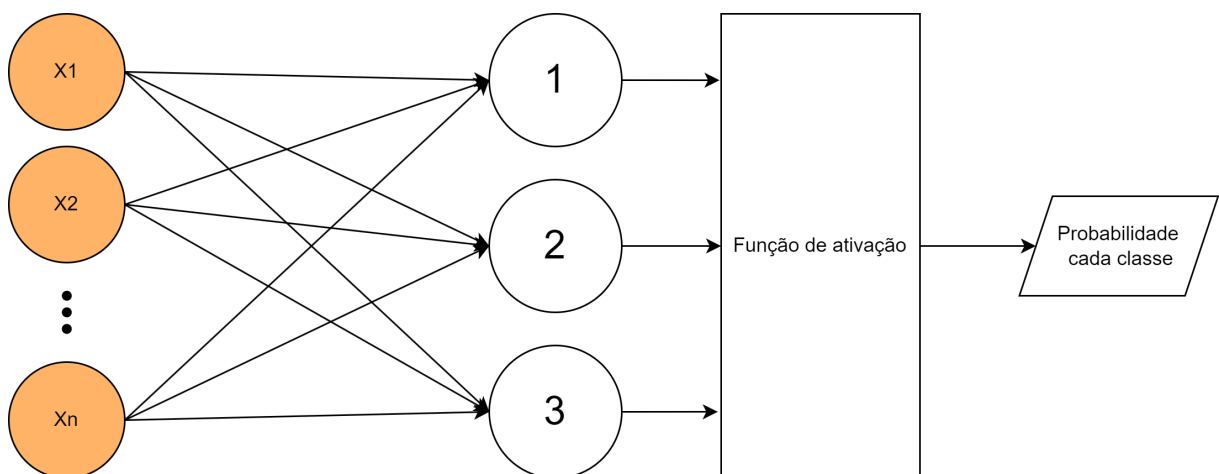


Figura 2.2: Funcionamento de uma camada densa.

2.1.2 Funções de ativação

As funções de ativação irão atuar diretamente sobre a saída de cada uma das camadas da arquitetura adotada. As mais utilizadas para classificação usando CNNs (Convolution Neural Network) são a ReLU e Softmax. O trabalho apresentado por (Lederer, 2021) resume as funções existentes e o (Ramachandran et al., 2017) propõe um método para gerar novas.

2.1.2.1 ReLU

O gráfico disposto na figura 2.3 ilustra o funcionamento da função de ativação ReLU. Como é possível observar, a vantagem dessa com relação a outras funções de ativação é sua facilidade em derivar o que aceleraria o processo de back propagation na etapa de treinamento da rede.

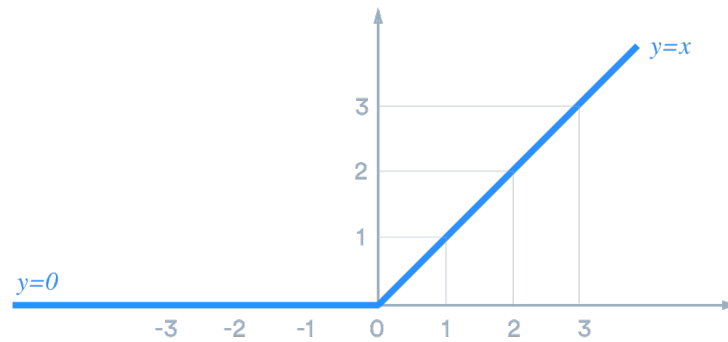


Figura 2.3: Funcionamento da função ReLU. Fonte: (Liu, 2017)

2.1.2.2 Softmax

A função softmax é usada para se obter a probabilidade para cada uma das classes, que no caso desse trabalho são: normal, pólipo e úlcera. A equação 2.1 é usada para se obter a probabilidade de cada uma das classes.

$$S(y_i) = \frac{e^{y_i}}{\sum_{n=1}^{n_classes} e^{y_n}} \quad (2.1)$$

onde y_i é o valor logits do neurônio sendo calculado e y_n é o valor logits do neurônio n .

O trabalho de (Madhusudhan, 2020) apresenta um bom resumo sobre a função Softmax.

2.1.3 Convoluções

Os processos de convolução, no âmbito de processamento de imagens e redes neurais, entrou para acelerar os treinamentos, reduzir o número de parâmetros e extrair características notáveis nas imagens. Abaixo estão listadas as vantagens em utilizá-la:

- Com um filtro predefinido é possível detectar bordas, texturas, formas e aplicar degradadores;

- Redução de parâmetros, uma vez que os pesos são compartilhados entre os pixels e dependem do tamanho do kernel configurado;
- Eficiência computacional, pois a forma que as convoluções funcionam permitem a paralelização que é vital para aceleração em GPU.

A imagem 2.4 demonstra como funciona um processo de convolução, onde: na esquerda tem a imagem sendo processada, em amarelo é a janela do kernel com seus pesos em vermelho, e na direita o resultado da operação de convolução. É possível observar na imagem que os pesos do kernel são multiplicados pelo seu respectivo pixel e então tudo é somado para se obter o valor para aquele quadrante.

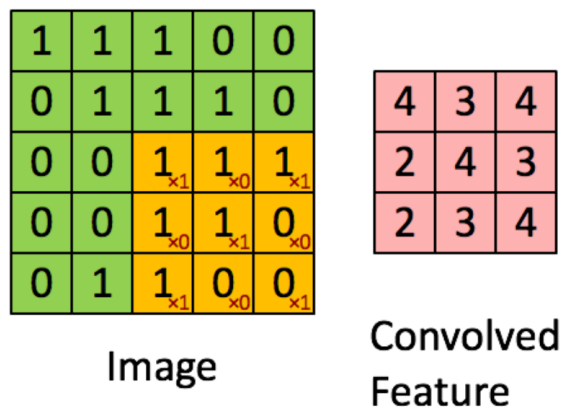


Figura 2.4: Funcionamento da convolução. Fonte: (Rangel, 2020)

Uma das características dos processos de convolução é a redução da resolução da imagem após cada processo, no qual quanto maior o tamanho do kernel, maior será a redução. Caso se queira evitar isso, basta usar uma técnica acrescentando linhas e colunas nas extremidades da imagem (chamado de padding) e geralmente os valores usados para esse preenchimento são zeros.

O trabalho apresentado por (Li et al., 2022) apresenta um resumo das redes CNNs existentes até seu ano de publicação e é um bom guia para a escolha de um modelo, por apresentar as vantagens e desvantagens de cada um.

Ainda com relação às CNNs, algumas operações são usadas em conjunto, como é o caso do agrupamento (pooling), normalização e linearização (flatten).

2.1.3.1 Agrupamento médio e máximo (max and average pooling)

As técnicas de pooling funcionam de forma muito similar ao que foi explicado anteriormente sobre convoluções. No caso do agrupamento médio, os valores dentro da janela do kernel serão usados para se tirar uma média que representará o resultado desse agrupamento no final da operação. Já no caso do agrupamento máximo, o valor máximo da janela do kernel é usado como resultado para o agrupamento.

2.1.3.2 Normalização

Quando se está trabalhando com redes convolucionais com muitas camadas é comum utilizar camadas de normalização entre cada operação de convolução. Isso é recomendado por dois principais fatores:

- Evitar o desvanecimento ou explosão dos valores de ativação: pode ocorrer de, na etapa de treinamento, após consecutivas operações de convolução os valores irem ficando cada vez menores ou muito grandes o que impacta de forma muito negativa, logo a normalização mantém esses valores dentro de um intervalo estável.
- Com os valores normalizados as redes tendem a convergir de forma mais acelerada.

2.1.3.3 Linearização (Flatten)

Como o próprio nome diz, é um processo de linearização, muito utilizado entre a última camada de convolução de uma rede CNN e sua camada densa para classificação. O procedimento é simples, a saída da última camada de convolução passa seus valores por um achatamento para ser representado por uma matriz unidimensional que pode servir de entrada para uma camada densa.

2.1.4 Função de perda

As funções de perda são métricas matemáticas que quantificam quão bem um modelo de aprendizado de máquina está realizando uma tarefa específica. Essas funções são essenciais para o treinamento de modelos, uma vez que permitem avaliar o quão perto ou longe as previsões do modelo estão dos valores reais ou dos rótulos do conjunto de treinamento. O objetivo durante o treinamento é minimizar o valor da função de perda, o que significa que o modelo está se tornando cada vez mais preciso na tarefa em questão.

2.1.4.1 Entropia cruzada (Cross Entropy)

A função de perda usada nesse trabalho é a Entropia Cruzada que é descrita pela equação 2.2.

$$CE = - \sum_i^c t_i \log(s_i) \quad (2.2)$$

onde t_i é o valor rotulado para a classe i e s_i é o valor predito pelo modelo para a classe i .

Nota-se pela equação 2.2 que somente o valor alvo em i irá manter seu valor na loss, pois os valores de t_i estão no formato one-hot, ou seja, todos os s_i que não estão relacionados à classe alvo serão multiplicados por zero.

2.1.5 Otimizadores

Em aprendizado de máquina, otimizadores são algoritmos que desempenham um papel fundamental no treinamento de modelos de machine learning. A principal função de um otimizador é ajustar os parâmetros do modelo de maneira iterativa, a fim de minimizar a função de perda (ou função de custo), que quantifica o quão bem o modelo está realizando uma tarefa específica.

O otimizador Adam é um dos algoritmos de otimização mais utilizados em aprendizado de máquina. É uma combinação de duas técnicas de gradiente descendente: média móvel exponencial (momento) e média móvel exponencial do quadrado (RMSprop).

O Adam foi apresentado primeiramente pelo trabalho de (Kingma and Ba, 2017) em 2014 e desde então é amplamente utilizado na comunidade de aprendizado de máquina por conta da sua velocidade de convergência no treinamento de modelos.

2.2 EfficientNets

As redes EfficientNets são um grupo de arquiteturas inicialmente apresentadas por (Tan and Le, 2020) com o objetivo de trazer uma relação eficiente entre resolução, profundidade e largura. Com base nesses três parâmetros, existem oito versões de EfficientNets variando de B0 (mais simples) até B7 (mais profunda).

Adicionalmente, o trabalho (Tan and Le, 2020) também compara o desempenho dessas redes com outras baseadas em CNNs amplamente utilizadas, como é o caso da ResNet.

A figura 2.5 demonstra o resultado obtido pelos autores de (Tan and Le, 2020) nessas comparações.

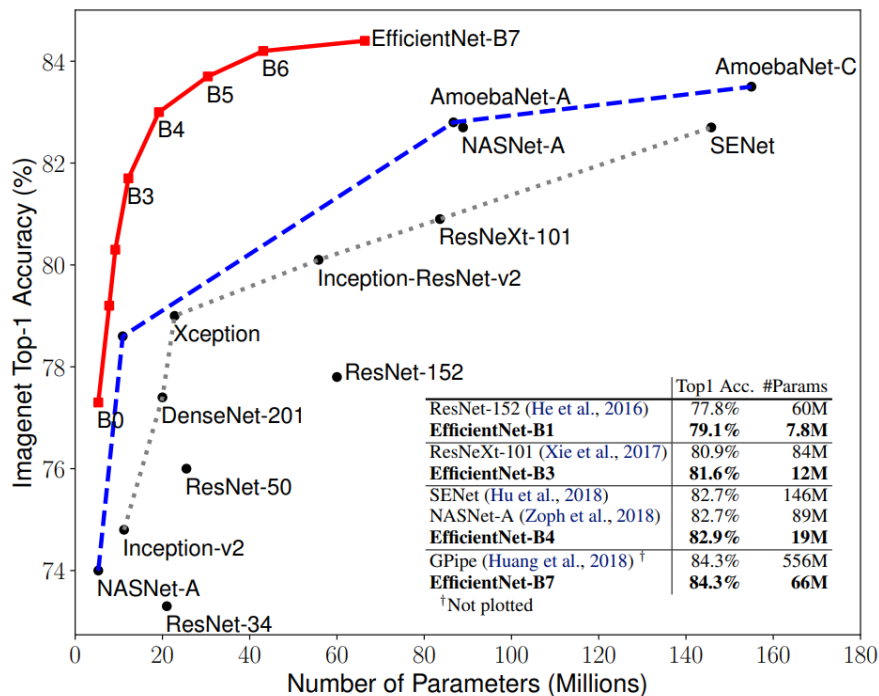


Figura 2.5: Acurácia x número de parâmetros das EfficientNets em relação aos modelos padrões de CNNs. Fonte:(Tan and Le, 2020)

2.2.1 EfficientNet-B1

O modelo usado para os experimentos foi o EfficientNet-B1. O motivo de sua escolha será detalhado nos tópicos de metodologia e resultados.

A figura 2.6 representa a estrutura do modelo EfficientNet-B1 usado nos experimentos deste trabalho.

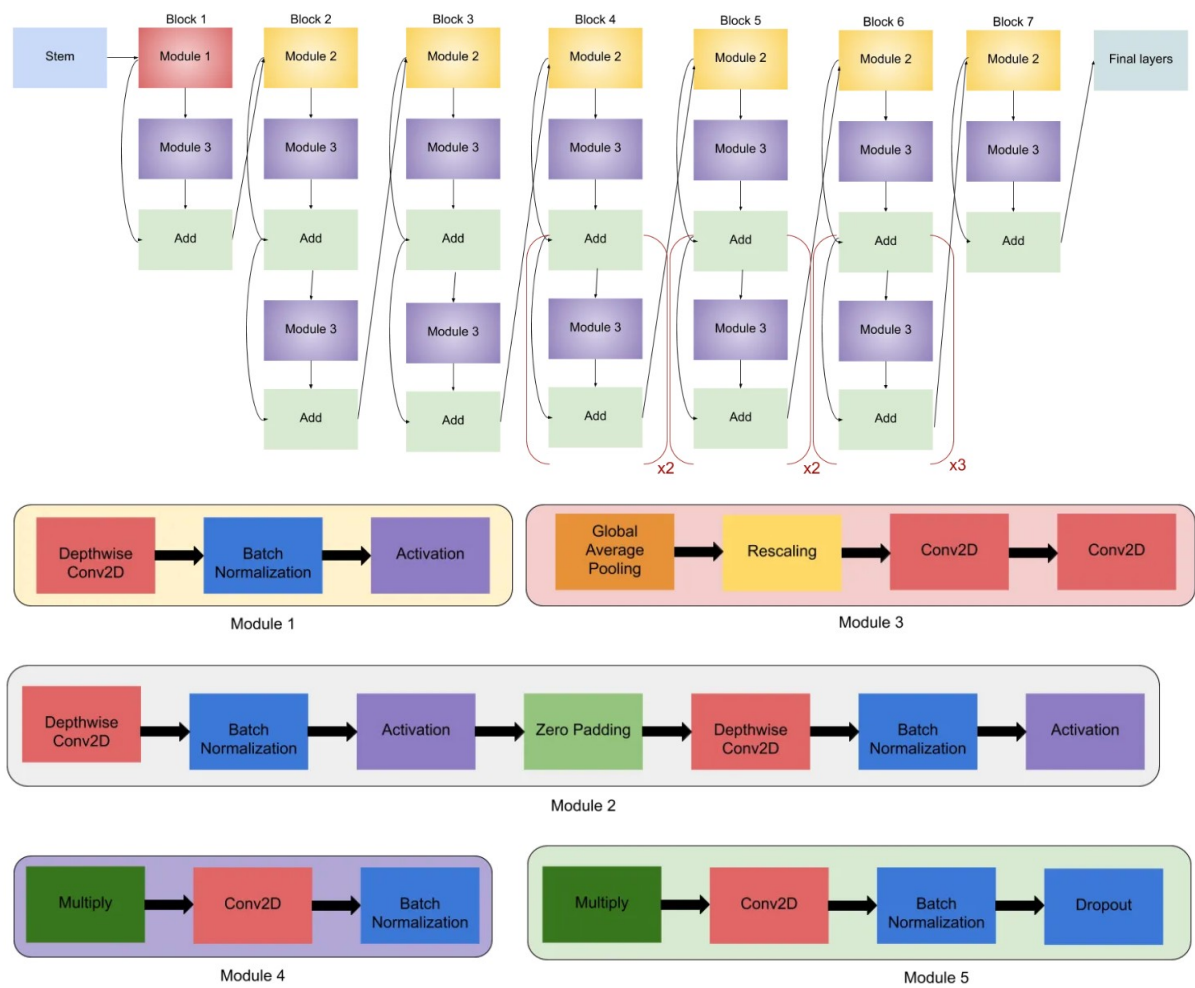


Figura 2.6: Arquitetura da EfficientNet-B1. Fonte: (Agarwal, 2020)

2.3 Métricas

As métricas usadas nos experimentos de classificação são: acurácia, precisão, sensibilidade (recall) e f1-score. Já no experimento de segmentação foi a IoU (Intersection Over Union). Em todas essas métricas, seus valores variam de 0 até 1.

A matriz confusão irá ser usada para exemplificar as métricas de classificação. Logo, será necessário fazer um breve resumo do que são matrizes confusão.

2.3.1 Matriz confusão

A matriz confusão é uma forma de ilustrar de forma prática como a rede está performando em uma parcela de dados na tarefa de classificação supervisionado. De forma resumida, em um dos eixos da matriz tem os valores reais (ground truth) e no outro os

valores preditos pelo modelo. O ideal para esses casos é que a maior parte das amostras estejam localizados na diagonal principal da matriz quadrada.

A figura 4.2 é um exemplo de matriz confusão para três classes (normal, pólipó e úlcera).

Na matriz confusão é possível separar o número de amostras em quatro subgrupos por classe:

- Verdadeiros positivos (True Positives - TP): Número de amostras corretamente classificadas como positiva para a classe;
- Falsos positivos (False Positives - FP): Número de amostras incorretamente classificadas como positivas para a classe;
- Verdadeiro negativo (True Negative - TN): Número de amostras corretamente classificadas como negativas para a classe;
- Falso negativo (False Negative - FN): Número de amostras incorretamente classificadas como negativas para a classe.

2.3.2 Acurácia

De forma breve, a acurácia é dada pela equação 2.3.

$$Acc = \frac{Classificações_corretas}{Total_amostras} \quad (2.3)$$

2.3.3 Precisão

A precisão pode ser dada de três formas: por classes, macro e micro.

Para calcular a precisão por classe basta usar a equação 2.4.

$$Precisao_{classe.a} = \frac{TP_{classe.a}}{TP_{classe.a} + FP_{classe.a}} \quad (2.4)$$

Para calcular a precisão macro deve-se usar a equação 2.5

$$Precisao_{macro} = \frac{precisao_{classe.a} + precisao_{classe.b} + \dots + precisao_{classe.n}}{n} \quad (2.5)$$

onde n é o número de classes.

Já no caso da precisão micro, deve-se usar a equação 2.6.

$$Precisao_{micro} = \frac{TP_a + TP_b + \dots + TP_n}{TP_a + FP_a + TP_b + FP_b + \dots + TP_n + FP_n} \quad (2.6)$$

onde n é o número de classes.

Se o modelo obtém bons valores para precisão, então ele performa bem em evitar os falsos positivos.

2.3.4 Sensibilidade (Recall)

A sensibilidade pode ser dada de três formas: por classes, macro e micro.

Para calcular a sensibilidade por classe basta usar a equação 2.7.

$$Recall_{classe.a} = \frac{TP_{classe.a}}{TP_{classe.a} + FN_{classe.a}} \quad (2.7)$$

Para calcular a sensibilidade macro deve-se usar a equação 2.8

$$Recall_{macro} = \frac{Recall_{classe.a} + Recall_{classe.b} + \dots + Recall_{classe.n}}{n} \quad (2.8)$$

onde n é o número de classes.

Já no caso da sensibilidade micro, deve-se usar a equação 2.9.

$$Recall_{micro} = \frac{TP_a + TP_b + \dots + TP_n}{TP_a + FN_a + TP_b + FN_b + \dots + TP_n + FN_n} \quad (2.9)$$

onde n é o número de classes.

Se o modelo obtém bons valores para sensibilidade, então ele performa bem em evitar os falsos negativos.

2.3.5 F1-score

A métrica F1-score traz um valor que é uma combinação da precisão e sensibilidade e é obtido pela equação 2.10.

$$F1_score = \frac{2 * precisao * recall}{precisao + recall} \quad (2.10)$$

2.3.6 Interseção sobre a união (Intersection Over Union - IoU)

Como o próprio nome diz, a métrica IoU segue a equação 2.11 (0 à 1 onde quanto maior melhor) e é ilustrada pela imagem 2.7.

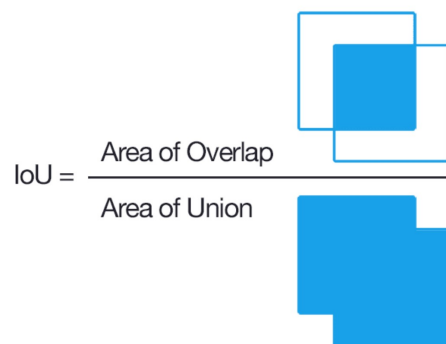
$$IoU = \frac{\textit{intersecao}}{\textit{uniaio}} \quad (2.11)$$


Figura 2.7: Representação visual da IoU. Fonte:([Rosebrock, 2016](#))

2.4 Grad-CAM (Class Activation Mapping baseado em Gradientes)

O Grad-CAM foi inicialmente apresentado por ([Selvaraju et al., 2019](#)) com o objetivo de ilustrar visualmente as decisões de redes neurais convolucionais, destacando as regiões de uma imagem que mais influenciaram a classificação ou previsão feita pelo modelo.

A imagem 2.8 demonstra o uso do Grad-CAM para visualização em uma classificação de cães e gatos numa imagem.

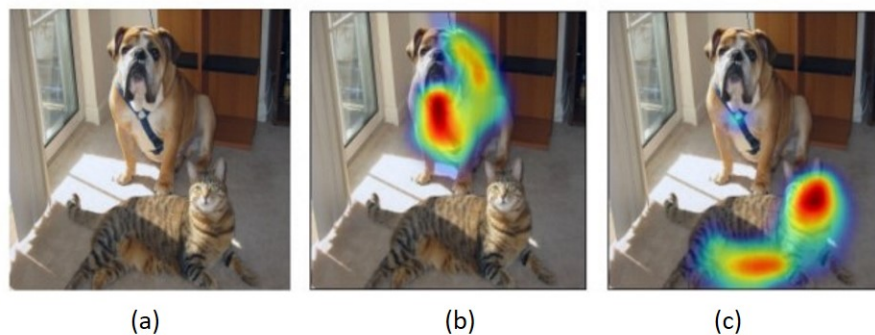


Figura 2.8: Imagem (a): original. Imagem (b): cachorro. Imagem (c): gato. Fonte:([Selvaraju et al., 2019](#))

Capítulo 3

Metodologia

O atual trabalho tem como objetivo apresentar um modelo EfficientNet para classificação multiclasse de imagens do TG inferior e segmentação de pólipos usando CAM. Sendo assim, dois experimentos foram realizados: sem aumento de dados (experimento 1) e com aumento de dados (experimento 2).

O diagrama da figura 3.1 ilustra como foram feitos os experimentos.

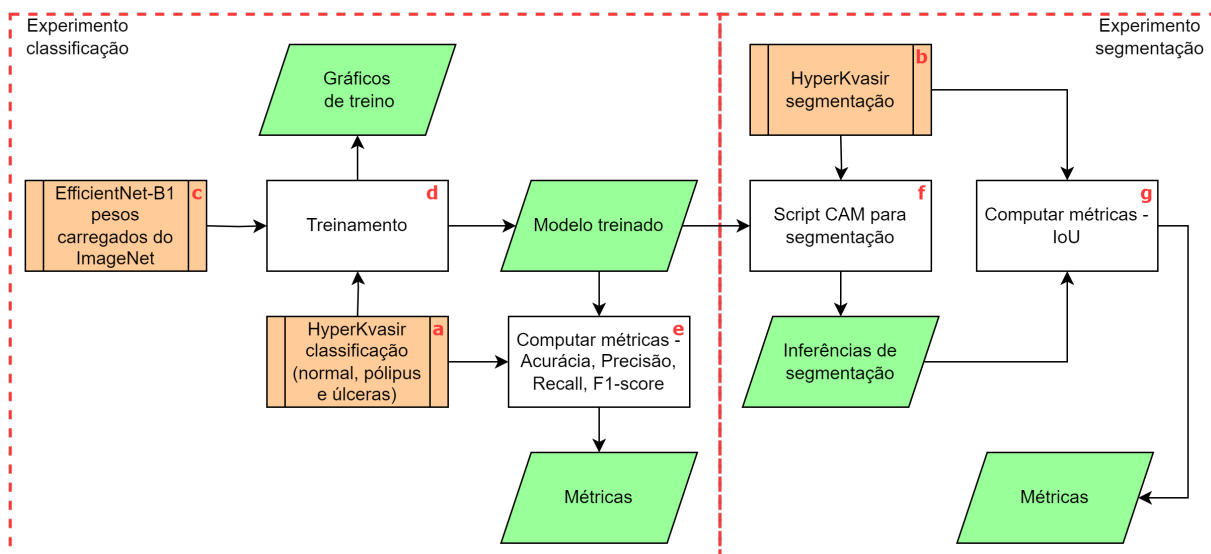


Figura 3.1: Diagrama dos experimentos

Na esquerda da figura estão os blocos referentes a classificação, e os da direita referentes a segmentação. Cada processo do diagrama está sendo indicado por uma letra de (a) à (g), que são detalhados nos seguintes subtópicos do atual capítulo.

3.1 HyperKvasir para classificação (a)

O banco de dados utilizado foi o HyperKvasir inicialmente apresentado por (Borgli et al., 2020), que consiste em 110.079 imagens do TG, no qual 10.662 imagens estão rotuladas e divididas em 23 classes. A tabela 3.1 ilustra a quantidade de amostras por classe:

Como o objetivo é classificar amostras do trato inferior entre normais, pólipos e úlceras apenas as classes marcadas por "*" na tabela 3.1 serão utilizadas, totalizando em 3.288 imagens. Dentre essas imagens: 1.409 são normais, 1028 são pólipos e 851 são úlceras.

Para o experimento, a classe hemorroida foi excluída por apresentar um número muito pequeno de amostras. A resolução das imagens do banco de dados não é a mesma para todas as amostras, então todas foram redimensionadas para 224x224 pixels, o mesmo formato de entrada para o modelo treinado. Nenhuma das imagens do banco de dados apresenta uma resolução maior que a definida, portanto não foi necessário usar técnicas para aumento de dimensão como interpolação linear.

Para o experimento 2 o procedimento adotado foi o mesmo apresentado por (Torres et al., 2023) e foi executado apenas na parcela de teste do banco de dados: rotação em 30°, rotação em 125°, cisalhamento e reflexão no eixo y. Dessa forma, o banco de dados aumentado resultou em 13.796 amostras.

A figura 3.2 representa uma das amostras com suas imagens derivadas do aumento de dados.

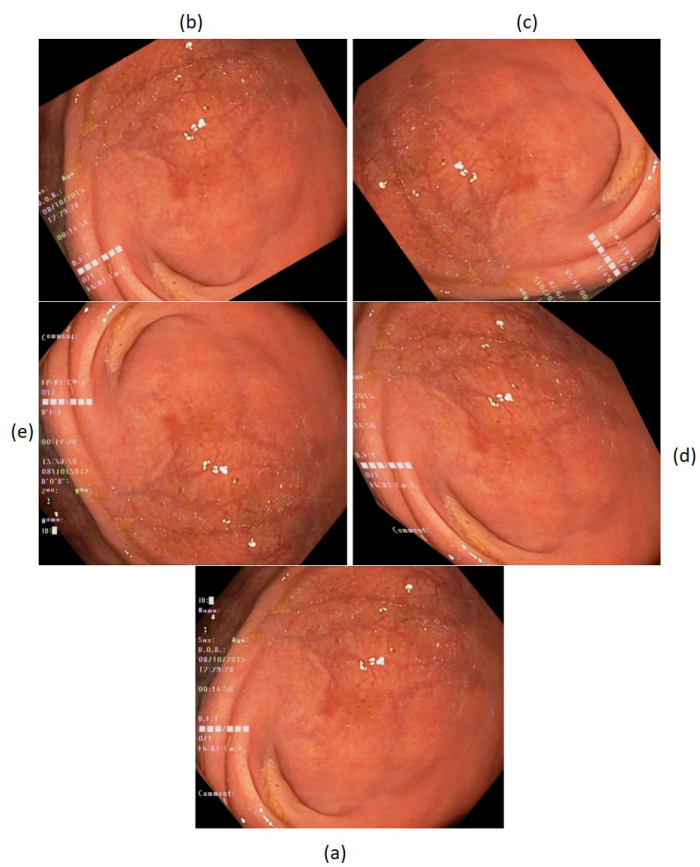


Figura 3.2: Imagem (a): original. Superior (b): rotação em 30° . Superior (c): rotação em 125° . Meio (d): reflexão eixo y. Meio (e): cisalhamento

Trato	Grupo	Classe	N amostras		
Trato Inferior	marcas anatômicas	Cecum *	1009		
		Ileum *	9		
		retroflex-rectum *	391		
	achados patológicos		Hemorrhoids	6	
			Polyps *	1028	
			ulcerative-colitis-grade-0-1 *	35	
			ulcerative-colitis-grade-1 *	201	
			ulcerative-colitis-grade-1-2 *	11	
			ulcerative-colitis-grade-2 *	443	
			ulcerative-colitis-grade-2-3 *	28	
			ulcerative-colitis-grade-3 *	133	
			qualidades das vistas da mucosa	bbps-0-1	646
				bbps-2-3	1148
		impacted-stool	131		
	intervenção terapêuticas	dyed-lifted-polyps	1002		
	dyed-resection-margins	989			
Trato Superior	marcas anatômicas	pylorus	999		
		retroflex-stomach	764		
		z-line	932		
	achados patológicos		barretts	41	
			barretts-short-segment	53	
			esophagitis-a	403	
			esophagitis-b-d	260	
Total de imagens			10662		

Tabela 3.1: Tabela referente ao banco de dados HyperKvasir com suas classes e número de amostras

3.2 HyperKvasir para segmentação de pólipos (b)

No HyperKvasir apenas a doença pólipo apresenta rotulos de segmentação, que serão usados para verificar a capacidade do algoritmo CAM para executar essa tarefa. O trata-

mento das imagens segue o mesmo processo feito para o banco de dados de classificação, porém as anotações de segmentação são máscaras de segmentação (preto e branco), onde o branco indica a região do pólipo (como mostra a figura 3.3). Ao todo, o banco de dados disponibiliza 1.000 amostras que foram usadas para computar a métrica IoU média.

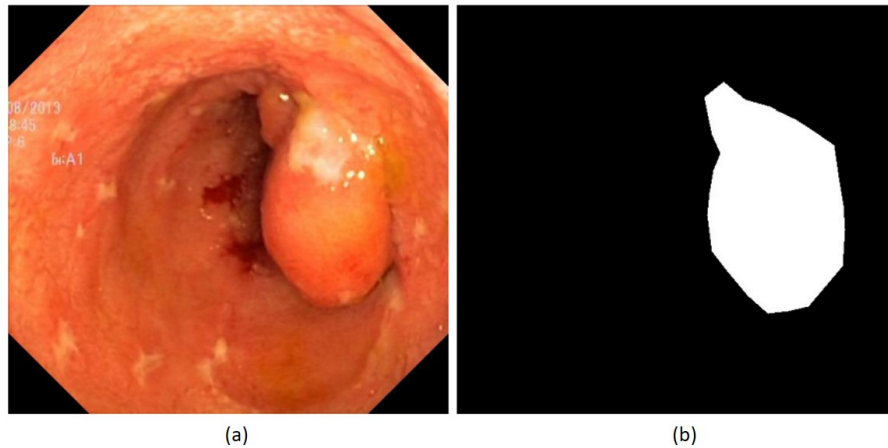


Figura 3.3: Imagem (a): imagem a ser segmentada. Imagem (b): máscara anotada disponível no banco de dados (Ground Truth)

3.3 EfficientNet-B1 (c)

A arquitetura utilizada neste trabalho foi a EfficientNet-B1, pois usou-se como base o estudo realizado por [Torres et al. \(2023\)](#), que realizou experimentos no mesmo banco de dados para todas as versões de EfficientNets e concluiu que, para essa quantidade de amostras, o modelo com melhores resultados foi a versão atualmente adotada (B1).

No final da arquitetura apresentada na figura 2.6, foi acrescentado uma camada de amostragem média (average pooling), uma camada de linearização (flatten) e uma camada densa com função de ativação softmax com três saídas. Dessa forma, a figura 3.4 representa o sumário da rede usada.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
efficientnetb1 (Functional) (None, 1280)                6575239
-----
flatten (Flatten)          (None, 1280)                0
-----
dense (Dense)              (None, 3)                   3843
-----
Total params: 6,579,082
Trainable params: 6,517,027
Non-trainable params: 62,055
-----

```

Figura 3.4: Sumário do modelo Keras da EfficientNet-B1

3.4 Treinamento (d)

Para treinamento, a proporção do banco de dados utilizada para ambos os experimentos 1 e 2 foi de 70, 10 e 20 por cento para treino, validação e teste, respectivamente. A função de perda usada foi a entropia cruzada (categorical cross entropy), otimizador Adam (Adaptive Moment Estimation), tamanho do batch 16, número máximo de épocas 200.

O hardware usado para o treinamento está descrito abaixo:

- Intel Core i5-12400
- Nvidia Geforce RTX 3060 (12GB VRAM)
- 16 GB de RAM
- Python 3.8.0
- Tensorflow-gpu 2.4.1

3.5 Métricas (e e g)

As métricas usadas para classificação foram:

- Acurácia
- Precisão (micro)

- Recall (micro)
- F1-score (micro)

Para cada um dos dois experimentos de classificação, uma matriz confusão foi gerada, assim como também os gráficos de convergência para acurácia e perda.

Já no caso da segmentação foi utilizado a métrica IoU.

3.6 CAM para segmentação (f)

A saída da última camada de convolução foi adotada como entrada para o algoritmo de CAM utilizado para segmentação de pólipos. Logo, imagens de 7x7 em 1.280 canais são usados para extrair o mapa de calor do CAM e, com base em um valor de limiar de 0.6 (threshold) é possível segmentar a classe identificada pelo modelo. A imagem 3.5 ilustra como é feito o processo para se extrair o heatmap usando CAM e a imagem 3.6 mostra como é extraído a máscara de segmentação por meio do heatmap.

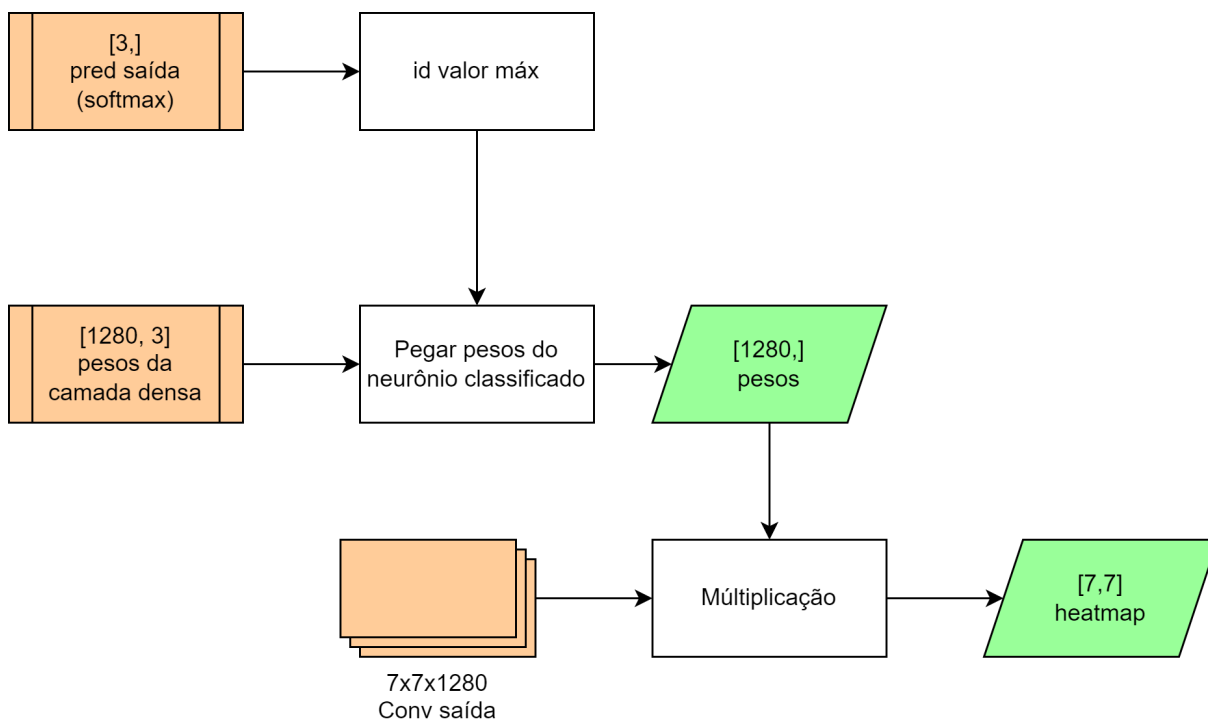


Figura 3.5: Diagrama detalhando processo do CAM

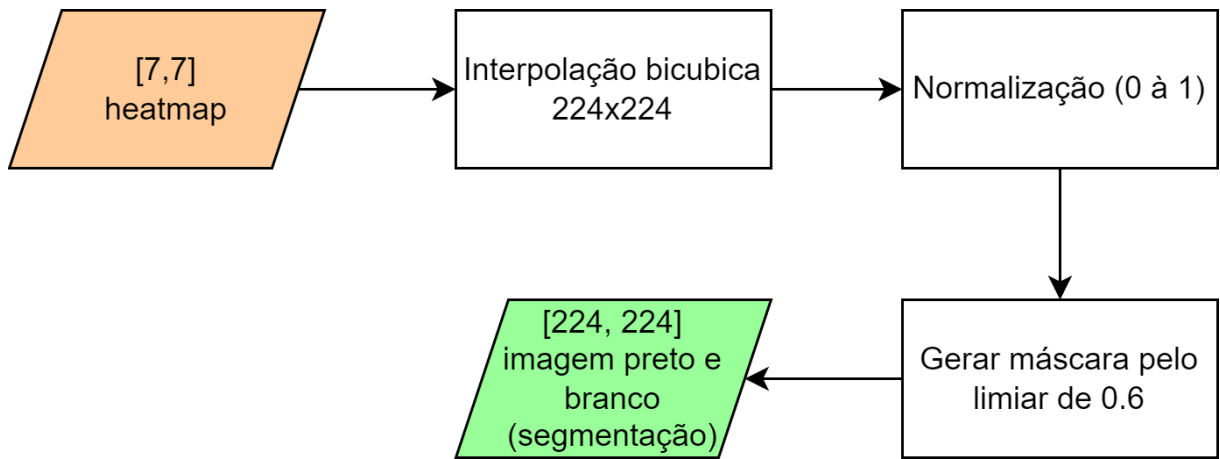


Figura 3.6: Diagrama detalhando o processo de segmentação usando saída do CAM

Capítulo 4

Resultados

4.1 Experimento 1 (sem aumento de dados)

4.1.1 Classificação

O treinamento do modelo EfficientNet-B1 levou 51 minutos para finalizar (99 épocas). O checkpoint com melhor valor de perda na parcela de validação foi salvo e usado para gerar as métricas na parcela de teste. No caso, o modelo que apresentou o melhor resultado foi obtido na época 24 com os valores abaixo:

- Perda treino: 0,0010
- Perda validação: 0,1069
- Acurácia de treino: 1,0000
- Acurácia validação: 0,9781

A figura [4.1](#) é referente aos gráficos de treino, para acurácia e perda, e indicam uma instabilidade característica de uma alta taxa de aprendizado.

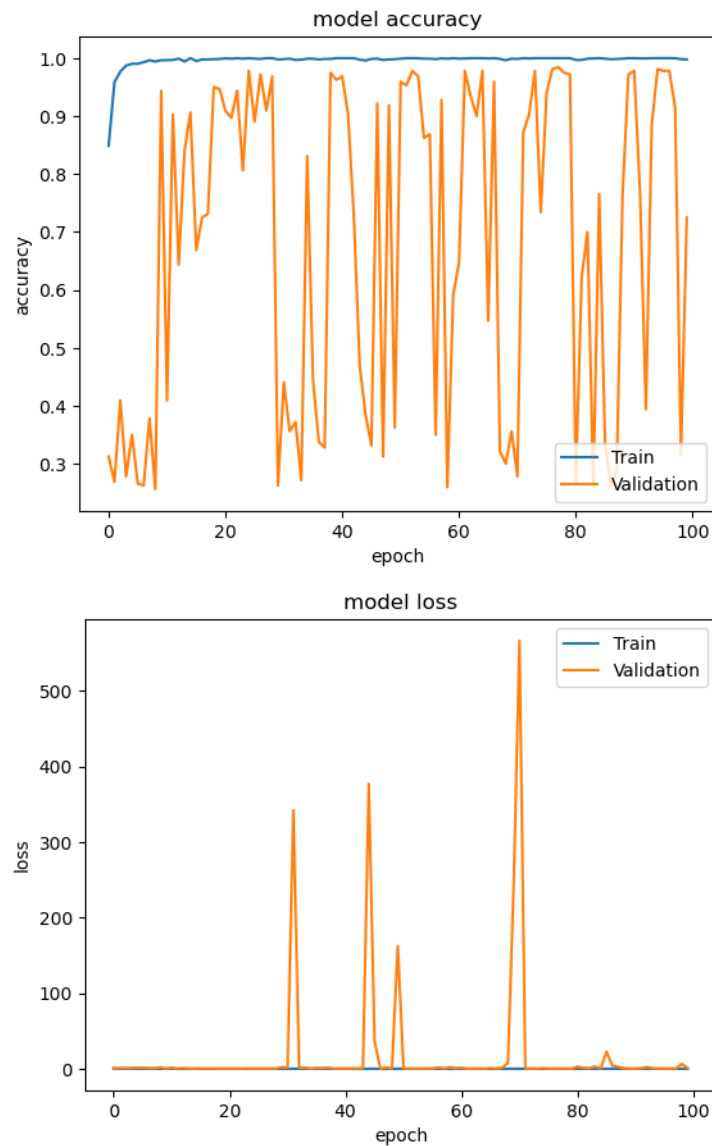


Figura 4.1: Gráficos de treino experimento 1. Superior: acurácia. Inferior: perda.

A figura 4.2 representa a matriz confusão gerada na parcela de teste e é possível observar que o modelo performou bem na classificação. Abaixo estão as métricas para a parcela de teste:

- Acurácia: 0,9712
- Precisão (micro): 0,9713
- Recall (micro): 0,9713
- F1-score (micro): 0,9713
- Precisão (classe pólipos): 0,9717

- Recall (classe pólipos): 0,9613

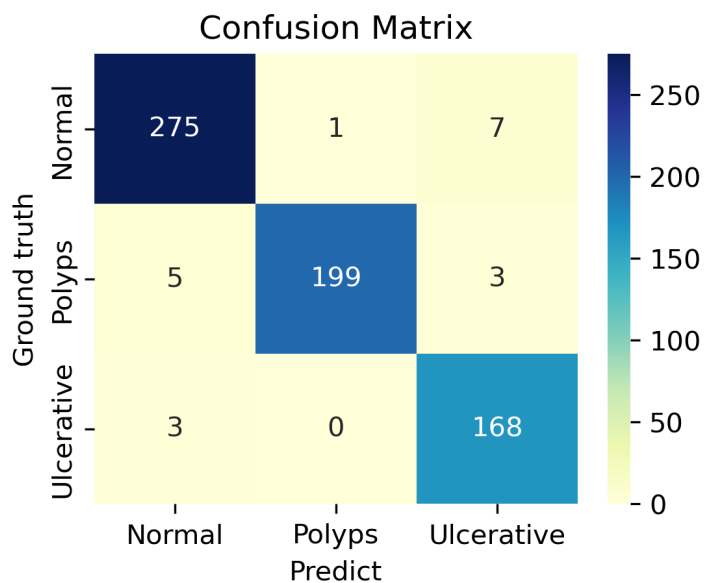


Figura 4.2: Matriz confusão referente ao experimento de classificação sem aumento de dados

4.1.2 Segmentação

Usando a saída da última camada de convolução, seus pesos e o vetor de classificação, o heatmap pode ser obtido. A imagem 4.3 representa quatro exemplos feitos nas amostras:

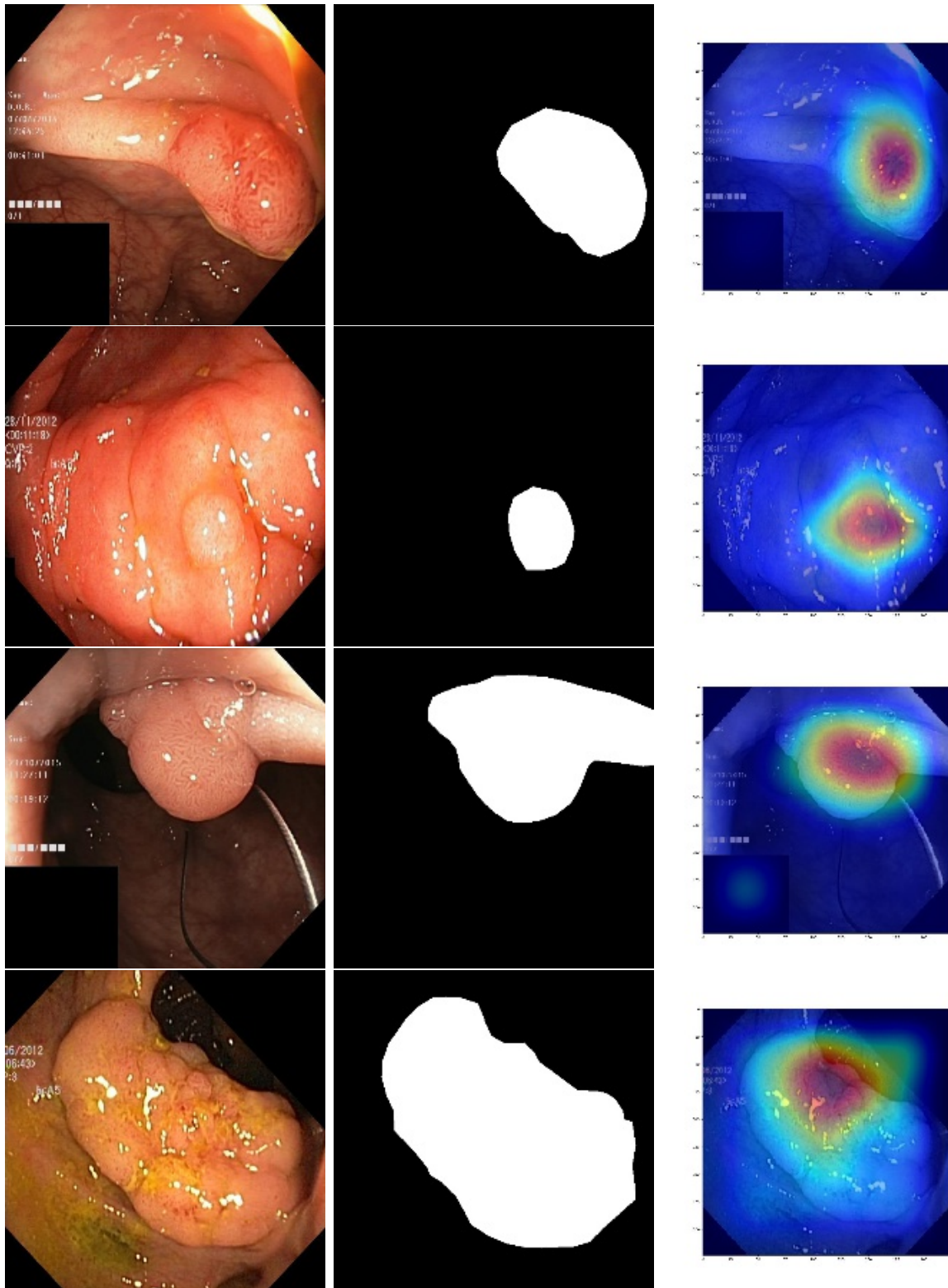


Figura 4.3: Exemplos da obtenção do heatmap no experimento 1. Esquerda: amostra. Meio: região da doença. Direita: heatmap.

Ainda referente à mesma amostra, a figura 4.4 ilustra como está sendo segmentado a doença pólipos.

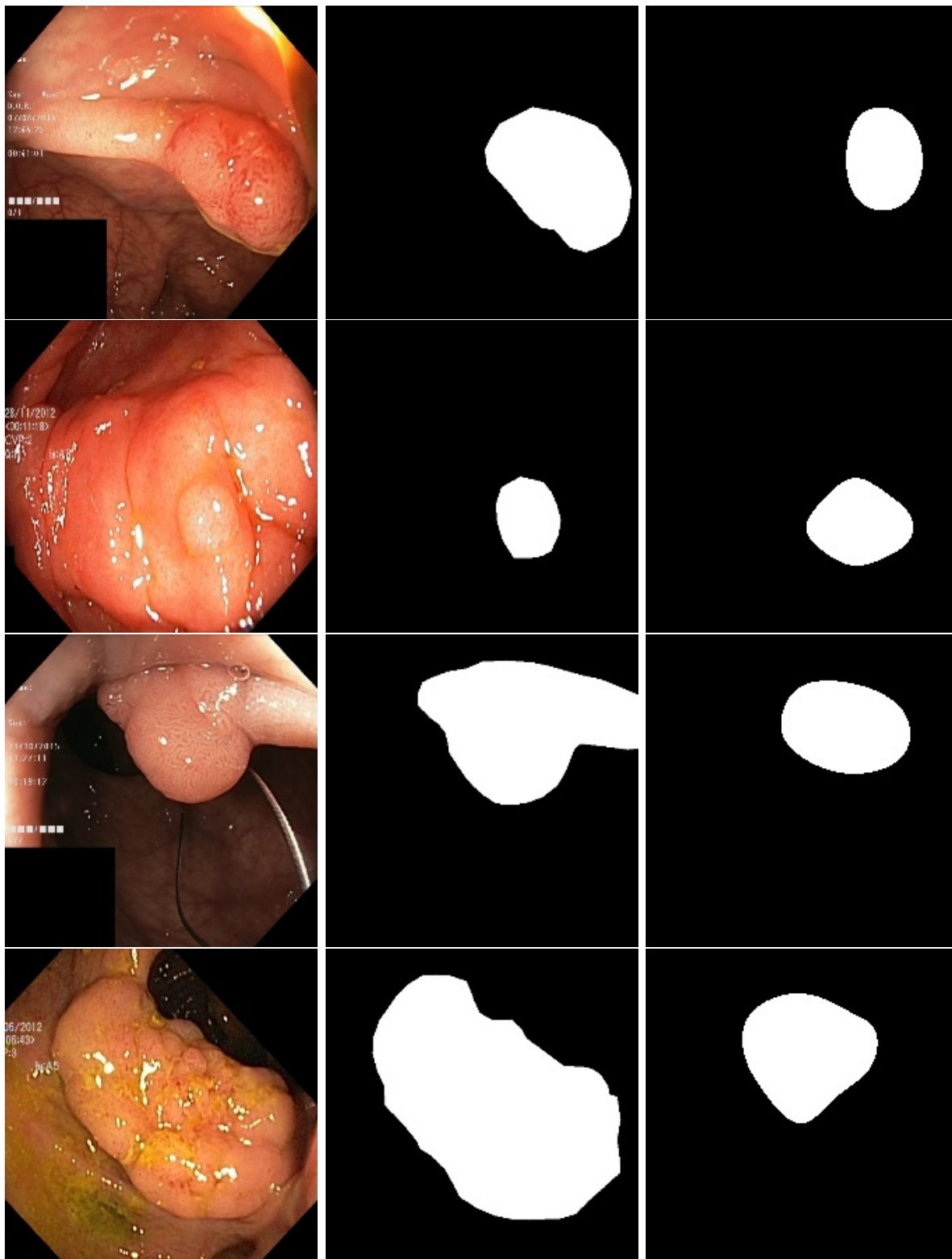


Figura 4.4: Resultados de segmentação no experimento 1. Esquerda: amostra. Meio: Ground Truth. Direita: Inferência.

Os valores abaixo são referentes à métrica IoU:

- Média: 0.3189

- Desvio padrão: 0,2107
- Mínimo: 0.0
- Máximo: 0,8078

O gráfico representado pela figura 4.5 demonstra a distribuição dos valores da métrica IoU.

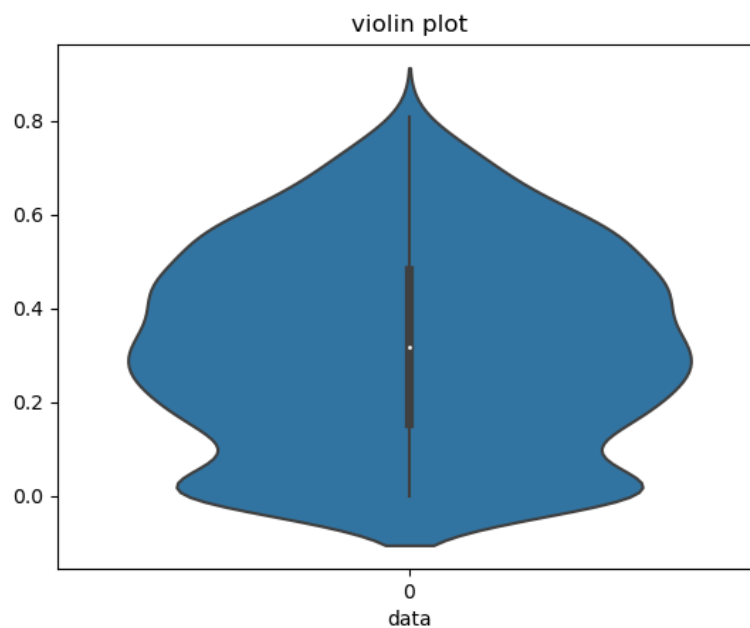


Figura 4.5: Gráfico ilustrando a distribuição dos valores para métrica IoU no experimento 1.

Vale ressaltar que a imagem sendo usada para segmentação tem tamanho de 7x7 pixels, logo é de se esperar que a solução não consiga realizar uma delimitação bem detalhada na segmentação, porém é possível observar que ela indica corretamente a posição da doença para uma boa parte das imagens.

4.2 Experimento 2 (com aumento de dados)

4.2.1 Classificação

O treinamento do modelo EfficientNet-B1 levou 9 horas e 5 minutos para finalizar (123 épocas). O checkpoint com melhor valor de perda na parcela de validação foi salvo

para gerar as métricas na parcela de teste. No caso, o modelo que apresentou o melhor resultado foi obtido na época 61 com os valores abaixo:

- Perda treino: 0,0020
- Perda validação: 0,1803
- Acurácia de treino: 0,9993
- Acurácia validação: 0,9608

A figura 4.6 é referente aos gráficos de treino, para perda e acurácia, e indicam uma instabilidade característica de uma alta taxa de aprendizado.

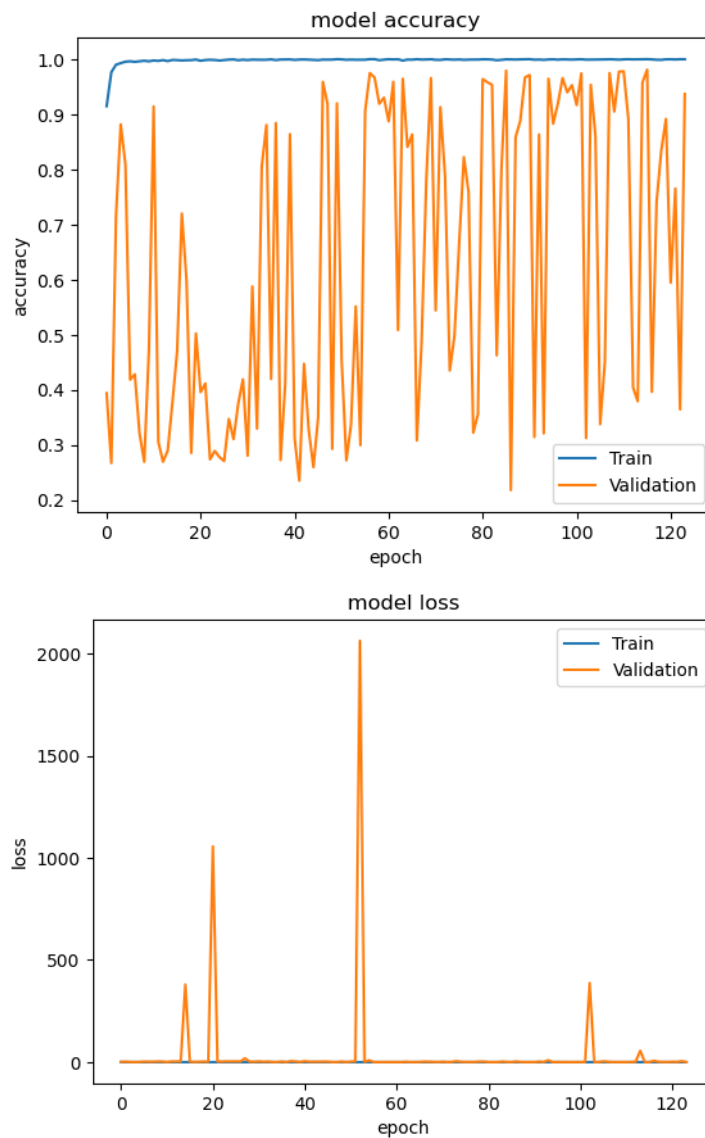


Figura 4.6: Gráficos de treino experimento 2. Superior: acurácia. Inferior: perda.

A figura 4.7 representa a matriz confusão gerada na parcela de teste e é possível observar que o modelo performou bem na classificação. Abaixo estão as métricas para a parcela de teste:

- Acurácia: 0,9637
- Precisão (micro): 0,9637
- Recall (micro): 0,9637
- F1-score (micro): 0,9637
- Precisão (classe pólipos): 0,9717
- Recall (classe pólipos): 0,9550

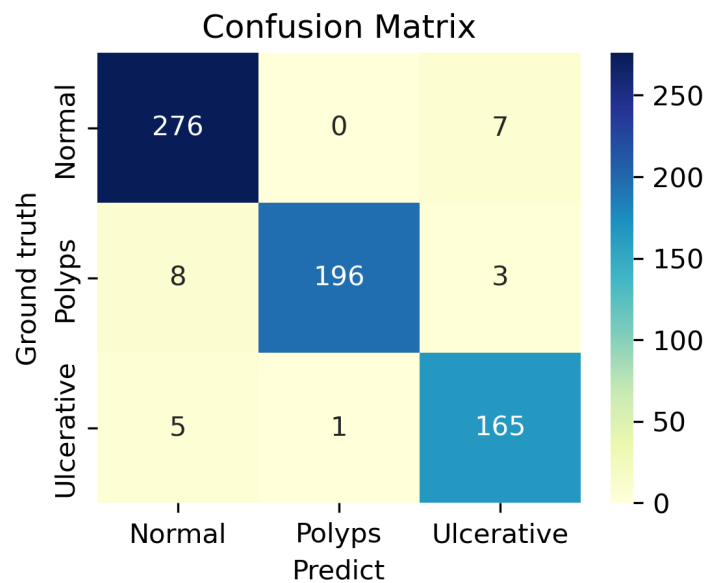


Figura 4.7: Matriz confusão referente ao experimento 2

4.2.2 Segmentação

Seguindo o procedimento descrito anteriormente, usando a saída da última camada de convolução, seus pesos e o vetor de classificação o heatmap pode ser obtido. A imagem 4.8 representa quatro exemplos feitos nas amostras:

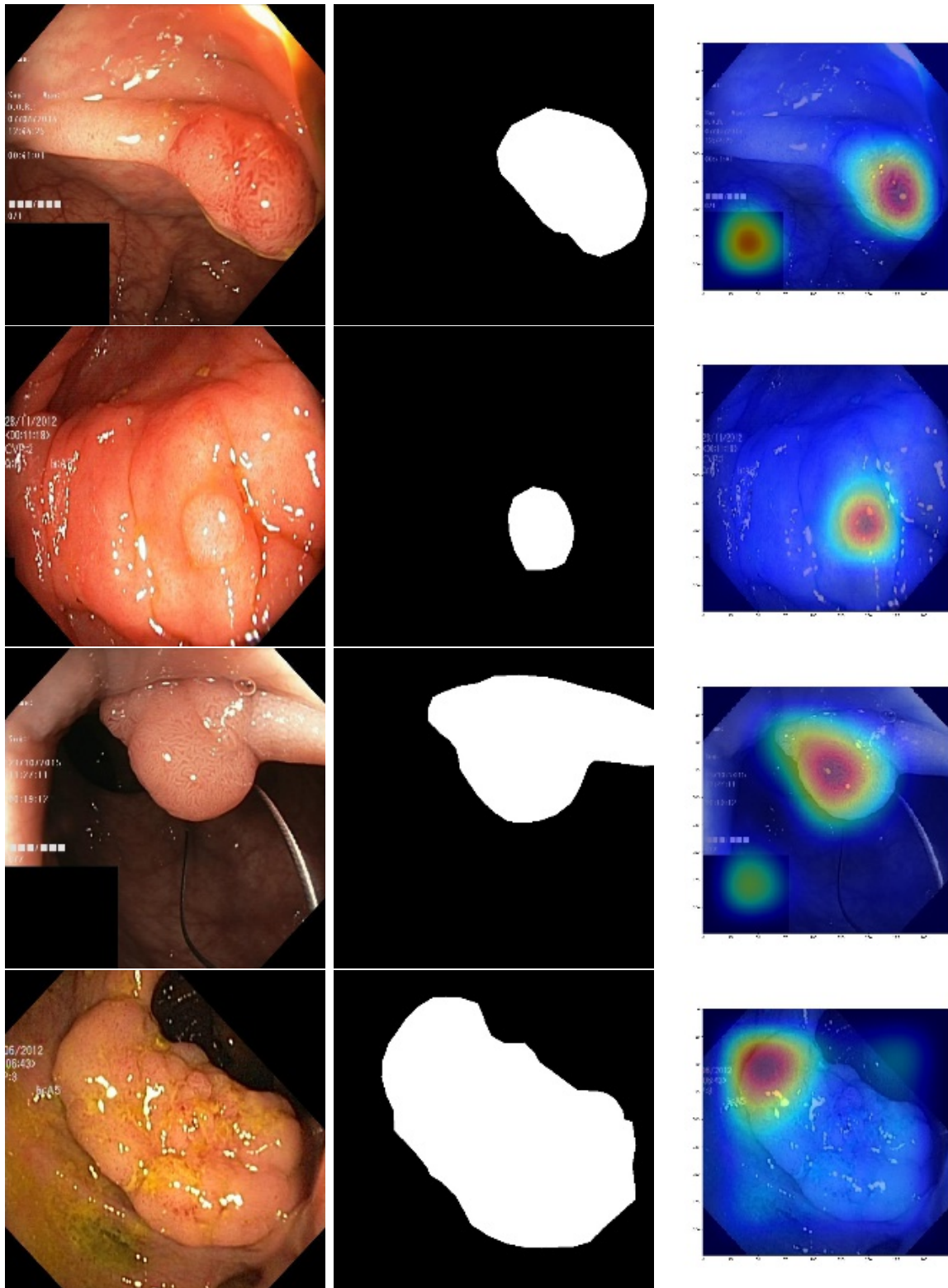


Figura 4.8: Exemplos da obtenção do heatmap no experimento sem aumento de dados. Esquerda: amostra. Meio: região da doença. Direita: heatmap.

Ainda referente à mesma amostra, a figura 4.9 ilustra como está sendo segmentado a doença pólipos.

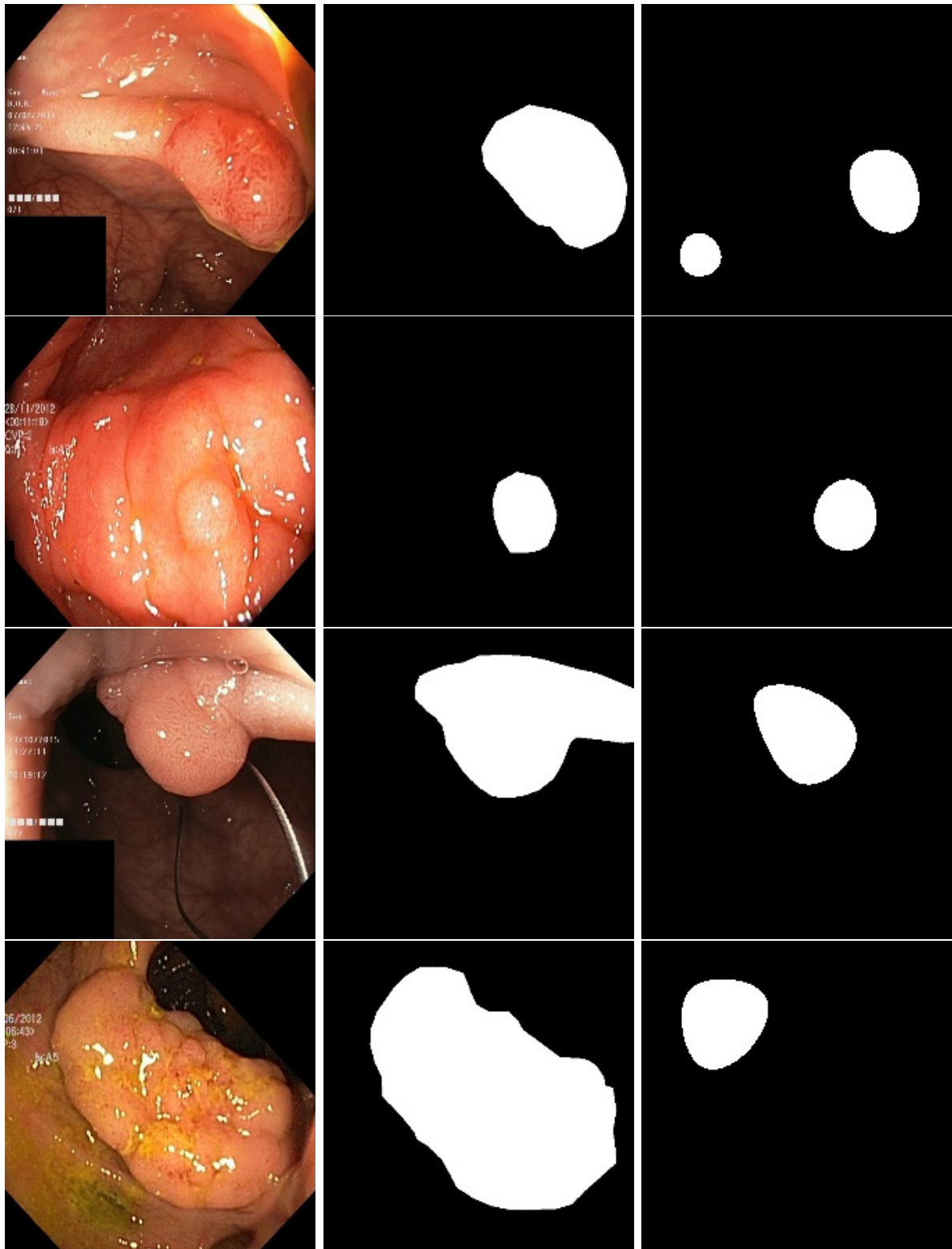


Figura 4.9: Resultados de segmentação no experimento sem aumento de dados. Esquerda: amostra. Meio: Ground Truth. Direita: Inferência.

Os valores abaixo são referentes à métrica IoU:

- Média: 0.3085

- Desvio padrão: 0,2159
- Mínimo: 0.0
- Máximo: 0,8368

O gráfico representado pela figura 4.10 demonstra a distribuição dos valores da métrica IoU.

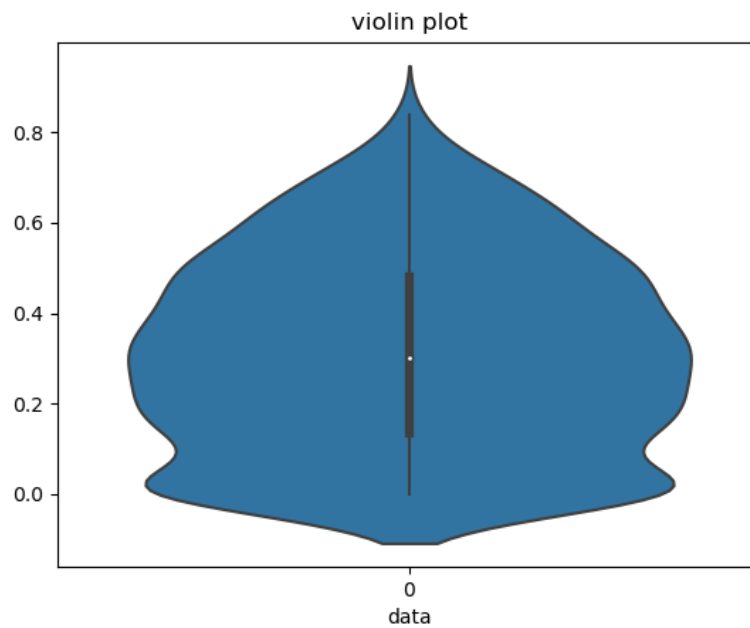


Figura 4.10: Gráfico ilustrando a distribuição dos valores para métrica IoU no experimento 2

Capítulo 5

Conclusão

Com base no que foi exposto, o experimento sem aumento de dados apresentou resultados levemente superiores aos com aumento (resultado contrário ao que foi apresentado por (Mohapatra et al., 2022)), tanto na classificação, quanto na segmentação. Isso pode estar relacionado ao fato da rede selecionada (EfficientNet-B1) ter desempenho melhor quando usado para uma parcela pequena de amostras, como é o caso deste trabalho. Possivelmente, o uso de uma rede mais complexa das EfficientNet (como a B6 ou B7) resultaria em resultados melhores quando usado o aumento de dados.

Vale ressaltar que o treino de uma rede mais complexa das EfficientNet no banco de dados aumentado levaria bastante tempo, visto que só na B1 o tempo aumentou de 51 minutos para 9 horas quando foi mudado para aumento de dados. E isso é outra vantagem da versão testada por esse trabalho (rápido tempo de treinamento e inferência).

Com relação à qualidade da segmentação, em ambos os experimentos o valor médio da métrica IoU não foi tão elevado, mas é capaz de indicar a posição do pólipo. Para reduzir o número de falhas na segmentação (IoU igual a zero) para trabalhos futuros, poderia ser acrescentado a IoU Loss na função de perda em algum momento no treino. Dessa forma, no treinamento a segmentação também seria levada em consideração para calcular a perda, aumentando bastante o desempenho para gerar a máscara na posição correta.

Para se obter uma máscara de segmentação mais detalhada, seria necessário realizar um aumento de resolução das imagens da convolução de 7x7 usando, possivelmente, convolução transposta (como na maioria das soluções usando modelos U-Net para essa problemática).

Referências Bibliográficas

- D. Liu, “A practical guide to relu,” <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>, 2017.
- R. F. Rangel, “Visão computacional — o que é convolução?” <https://medium.com/turing-talks/vis%C3%A3o-computacional-o-que-%C3%A9-convolu%C3%A7%C3%A3o-ad709f7bd6b0>, 2020.
- M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” 2020.
- V. Agarwal, “Complete architectural details of all efficientnet models,” <https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142>, 2020.
- A. Rosebrock, “Intersection over union (iou) for object detection,” <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>, 2016.
- R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2019. [Online]. Available: <https://doi.org/10.1007%2Fs11263-019-01228-7>
- S. Mohapatra, G. Kumar Pati, M. Mishra, and T. Swarnkar, “Gastrointestinal abnormality detection and classification using empirical wavelet transform and deep convolutional neural network from endoscopic images,” *Ain Shams Engineering Journal*, vol. 14, no. 4, p. 101942, 2023.
- L. Torres, O. Pessoa, I. Correa, E. Soares, A. Oliveira, M. Costa, and C. Costa Filho,

- “Identification of digestive tract abnormalities on endoscopic images,” in *19th International Symposium on Medical Information Processing and Analysis (SIPAIM)*, 2023.
- J. Lederer, “Activation functions in artificial neural networks: A systematic overview,” 2021.
- P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” 2017.
- S. Madhusudhan, “The softmax activation function,” <https://medium.com/@shashank.1842000/the-softmax-activation-function-4a184310b3b4>, 2020.
- Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: Analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.
- D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- H. Borgli, V. Thambawita, P. Smedsrud, S. Hicks, D. Jha, S. Eskeland, K. Randel, K. Pogorelov, M. Lux, D. T. Dang Nguyen, D. Johansen, C. Griwodz, H. Stensland, E. Garcia Ceja, P. Schmidt, H. Hammer, M. Riegler, P. Halvorsen, and T. de Lange, “Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy,” *Scientific Data*, vol. 7, 08 2020.
- S. Mohapatra, G. K. Pati, and T. Swarnkar, “Efficiency of transfer learning for abnormality detection using colonoscopy images: A critical analysis,” in *2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 2022, pp. 1–6.
- Seachaos, “Get heatmap from cnn (convolution neural network), aka cam,” <https://tree.rocks/get-heatmap-from-cnn-convolution-neural-network-aka-grad-cam-222e08f57a34>, 2021.

Códigos Utilizados

.1 CAM

O código abaixo foi feito seguindo a referência [Seachaos \(2021\)](#).

```
1 # Código para obter o heatmap pelo CAM
2 target = np.argmax(pred, axis=1).squeeze()
3 w, b = nested_layers.get_layer("dense").weights
4 weights = w[:, target].numpy()
5 heatmap = new_conv.squeeze() @ weights
```

A variável 'pred' é um vetor com as probabilidades de cada classe (0 à 1) e 'w' são os pesos da camada densa.