

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**CONTAGEM DE COLONIAS DE BACTERIAS UTILIZANDO REDES
NEURAS CONVOLUCIONAIS**

PEDRO HENRIQUE MENDES LIMA E SILVA

GOIÂNIA
2021

PEDRO HENRIQUE MENDES LIMA E SILVA

**CONTAGEM DE COLONIAS DE BACTERIAS UTILIZANDO REDES
NEURAS CONVOLUCIONAIS**

Trabalho de Conclusão de Curso apresentada à
Escola de Ciências Exatas e da Computação, da
Pontifícia Universidade Católica de Goiás, como
requisito parcial para a obtenção do título de
Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Clarimar José Coelho

GOIÂNIA

2021

PEDRO HENRIQUE MENDES LIMA E SILVA

**CONTAGEM DE COLONIAS DE BACTERIAS UTILIZANDO REDES
NEURAS ARTIFICIAIS**

Este Trabalho de Conclusão de Curso julgado adequado para a obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, em: ____/____/____.

Prof. Dr. Clarimar José Coelho
Orientador de TCC

Coorientador: Dr. Arlindo Rodrigues
Galvão Filho

Banca Examinadora:

Engenheiro Diogo Fernandes Costa

Engenheiro Douglas Vieira do Nascimento

GOIÂNIA
2021

AGRADECIMENTOS

Quero agradecer pela oportunidade, pelo esforço para realizar este projeto e todos que dispuseram de seu tempo para me auxiliar a vencer esse desafio. Segundo pela minha família e amigos que sempre estiveram ao meu lado torcendo pelo meu sucesso. Agradeço também ao IPTESP - Instituto de Patologia Tropical e Saúde Pública por disponibilizar as amostras utilizadas no projeto.

*“Não pode o homem ter melhor morte que: Lutando contra o desconhecido.
Pelos cinzas de seus Pais e pelos Templos de seus deuses!”*

Horácio Cocles

RESUMO

A contagem de colônias de bactérias é um trabalho árduo e de suma importância para a área de biomedicina. Tão importante para as pesquisas feitas em laboratório pois com a quantidade de colônias de bactérias é onde é uma das formas de ser definido quanto e qual o antibiótico vai ser utilizado para o tratamento daquela bactéria em específico, também é analisado a poluição existente dentro da amostra. Hoje a contagem é realizada de maneira manual pelos profissionais na área. O intuito deste trabalho é realizar a contagem da colônia de bactérias de forma automatizada em placas de petri para auxiliar na produtividade e performance dos profissionais da área, serão usadas abordagens utilizando imagens digitais em conjunto com técnicas de *Deep learning*.

Palavras-chave: *Deep learning*, placa de petri, Faster R-CNN, Colônia de bactérias, contagem, detecção de objetos.

ABSTRACT

The counting of bacterial colonies is an arduous and extremely important job for biomedicine. It is so important for research carried out in the laboratory because with the number of bacterial colonies in each sample, this is where the definition of how much and which antibiotic will be used, the existing pollution within the sample is also analyzed. Today the counting is performed manually by professionals in the area. The purpose of this job is to carry out the automated counting of the bacterial colony in petri dishes to assist in the productivity and performance of professionals in the area, the approaches that will be used consists in using digital images together with Deep learning techniques.

Keywords: Deep learning, Petri dish, Faster R-CNN, bacterial colony, score, object detection.

LISTA DE FIGURAS

Figura 1 - Demonstração de uma imagem digital bidimensional	13
Figura 2 - Demonstração espaço de cores RGB	13
Figura 3 - exemplo de uma arquitetura de uma CNN	16
Figura 4 - exemplo de utilização do maxpooling	18
Figura 5 - Arquitetura de uma R-CNN.....	19
Figura 6 - Demonstração da arquitetura da Faster R-CNN	20
Figura 7 – exemplo de amostras cultivadas pelo IPTESP	21
Figura 8 - exemplo de figura anotada com <i>bounding boxes e labels</i>	23
Figura 9 - amostras de treino com a inferência do modelo 1	25
Figura 10 - amostras de treino com a inferência do modelo 2	26
Figura 11 - inferência do modelo 1 no <i>dataset</i> de treino	27
Figura 12 - inferência do modelo 1 no <i>dataset</i> de treino	28
Figura 13 - inferência do modelo 1 no <i>dataset</i> de treino	29
Figura 14 - inferência do modelo 1 no <i>dataset</i> de treino	30
Figura 15 - inferência do modelo 1 no <i>dataset</i> de treino	31
Figura 16 - inferência do modelo 2 no <i>dataset</i> de treino	33
Figura 17 - inferência do modelo 2 no <i>dataset</i> de treino	34
Figura 18 - inferência do modelo 2 no <i>dataset</i> de treino	35
Figura 19 - inferência do modelo 2 no <i>dataset</i> de treino	36
Figura 20 - inferência do modelo 2 no <i>dataset</i> de treino	37

LISTA DE TABELAS

Tabela 1 - Divisão de dados de treino e teste	24
Tabela 2 - representação de resultado para amostras do modelo 1 e modelo 2	26
Tabela 3 – Resultado da inferência dos modelos no <i>dataset</i> de treino.....	38

LISTA DE ABREVIATURAS

UFC Unidade formadora de colônia

RGB *Red, Green e Blue*

DL *Deep learning*

ROI *Region of interest*

MS COCO *Microsoft Common Objects in Context*

CNN *Convolutional Neural Network*

ReLU *Rectifier Linear Unity*

R-CNN *Region-based CNNs*

FR-CNN *Faster Region-Based Convolutional Neural Network*

RPN *Region Propostal Network*

SUMÁRIO

1. INTRODUÇÃO	11
2. FUNDAMENTAÇÃO TEÓRICA.....	12
2.1. Imagens digitais.....	12
2.2. Região de interesse	14
2.3. Anotação das imagens	14
2.3.1. Labelme	15
2.4. Redes convolucionais.....	15
2.4.1. Camada Convolucional.....	16
2.4.2. Função de ativação	17
2.4.3. Camada de Subamostragem (Pooling).....	17
2.4.4. Camada de Totalmente Conectada (Full Connected Layer).....	18
2.5. Detectores de objetos baseados em dois estágios.....	18
3. MATERIAIS E MÉTODO.....	21
3.1. Amostras.....	21
3.2. Método proposto	21
3.3. Treinamento.....	24
4. RESULTADOS.....	24
5. CONCLUSÃO	38
6. REFERÊNCIAS	40

1. INTRODUÇÃO

A contagem de colônias de microrganismos, como bactérias é uma parte essencial do trabalho de biomédicos, e amplamente usadas em muitas análises da área biomédica. Existe um baixo rendimento de colônias de bactérias enumeradas, é utilizado muito esforço dos profissionais para realizar essa contagem de colônias de bactérias, pode haver centenas ou milhares de colônias de bactérias em uma área das placas de Petri, o processo de contagem geralmente é manual, executado por técnicos bem treinados (ZHANG, 2008).

Na pesquisa biomédica e no diagnóstico clínico, existe uma grande necessidade de quantificar o número de bactérias em uma amostra. Para analisar os resultados da cultura bacteriana. A contagem de colônias bacterianas é usada para contar o número de bactérias vivas como colônias. Este tipo de determinação é feito despejando uma amostra liquefeita contendo microrganismos em uma placa de Petri e cultivando o número de microrganismos sobreviventes cultivados, para ter esse ambiente de cultura, são necessários nutrientes básicos que alimentem as bactérias, até formar colônias na placa (também chamada de unidade formadora de colônia - UFC). Avaliação é realizada por verificar a taxa de sobrevivência dos microrganismos na amostra. Esses testes também são amplamente utilizados nos campos de biomedicina, como exames, testes de segurança de alimentos e medicamentos, monitoramento ambiental e saúde pública (ZHANG, 2008).

Com o avanço da tecnologia e estudos computacionais hoje em dia, existem métodos alternativos que aprimoram a contagem dessas colônias de bactérias, esses estudos propõem o uso de modelos computacionais que utilizam imagens digitais para identificar as colônias de bactérias e realizar a contagem, um desses métodos foram propostos por Alessandro et al. (2016) é utiliza técnicas de aprendizagem de máquina (*Deep learning*, DL) para realizar a contagem das bactérias.

Existem sistemas para realizar a contagem desses microrganismos, porém alguns desses sistemas utilizam métodos que tiveram problemas quando utilizado amostras com colônias agrupadas (NIYAZI et al., 2007).

Matthew et al. (2010) desenvolveu um sistema com um menor custo que possui uma melhor eficácia que os demais sistemas comparados, foi utilizada uma câmera digital ou scanner de documentos. O software, chamado de “NICE” (*NIST’s Integrated Colony Enumerator*), lê o padrão de formatos de imagem e, portanto, pode ser usado em conjunto com muitos sistemas de imagem.

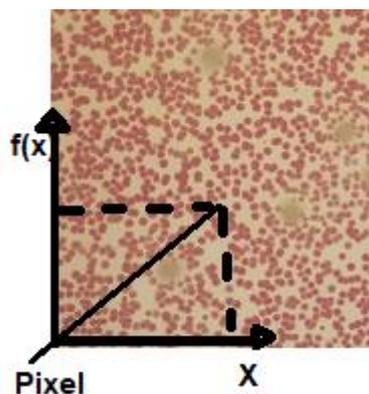
Diante disso, este trabalho tem como objetivo realizar a contagem de bactérias, a partir de imagens digitais em conjunto com técnicas de DL para realizar a segmentação das colônias do fundo da imagem, e dado a segmentação realizada é possível realizar a contagem das colônias de bactérias de uma respectiva imagem digital.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Imagens digitais

A representação da imagem em formato digital é feita através de matrizes, onde, são mapeados os objetos e cores do mundo real para o mundo digital. A matriz é composta por pequenos elementos de imagem, também chamados de pixel (abreviação de *Picture Element*) (CHAGAS, 2021). Os Pixels são considerados o menor componente de uma imagem digital, quanto mais pixels presentes na imagem, a representação da imagem digital ficara mais próxima do objeto real, cada pixel tem um limite de 8 bits, onde possuem valores inteiros e variam entre 0 e 255, o pixel representa um ponto de amostra no(s) eixo(s) $f(x, y)$ e contém o valor que representa a cor nessa coordenada (BAXES, 1994), conforme ilustra a Figura 1

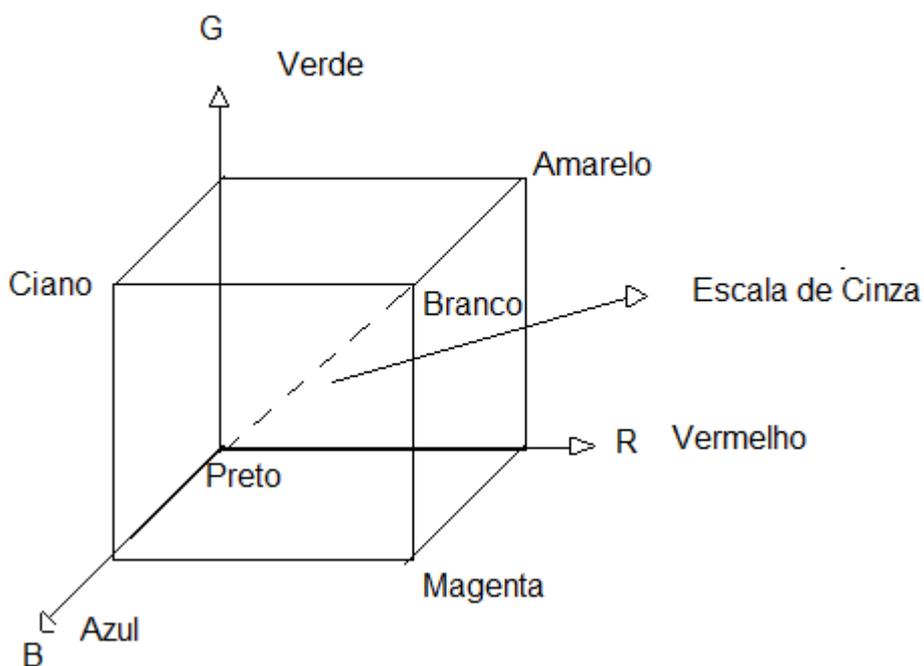
Figura 1 - Demonstração de uma imagem digital bidimensional



Fonte: Autoria própria (2021)

A combinação de três cores, quais são elas RGB (*Red, Green and Blue*) em português traduzimos como vermelho, verde e azul nesta mesma sequência, pode gerar todas as demais cores, essa é um dos modelos para representar cores em imagens digitais, onde cada pixel é composto por 3 canais, nos quais cada cor tem um canal, o primeiro canal é para a cor *Red* (vermelho), o segundo canal para a cor *Green* (verde) e o terceiro canal para a cor *Blue* (azul), esse modelo de cores é baseado em um sistema de coordenadas cartesianas (PEDRINI, 2008), como representado na figura 2.

Figura 2 - Demonstração espaço de cores RGB



Fonte: Autoria própria (2021)

2.2. Região de interesse

A região de interesse (ROI) é a parte da imagem que você deseja filtrar ou manipular. Você pode expressar a ROI como uma imagem de máscara binária. Na imagem da máscara, os pixels pertencentes à ROI são definidos como 1 e os pixels fora da ROI são definidos como 0. Você pode criar uma máscara binária selecionando pixels com base nos valores de intensidade ou posição, ou usando técnicas binárias ou de segmentação, o ROI pode ser representado como formas geométricas e desenhadas à mão (MATLAB, 2021).

2.3. Anotação das imagens

É possível criar ROI a mão livre ou com formas geométricas como, círculos, elipses, polígonos, linhas, polilinhas. Assim o ROI é anotado na imagem realizando assim a separação do ROI e de tudo aquilo que é diferente do ROI. Neste trabalho foi utilizado a ferramenta Labelme, para realizar a anotação da imagem, essa ferramenta é um *open source* encontrado em repositório público, criado por Kentaro Wada e recebe código de várias pessoas que contribuem com o repositório. As anotações das imagens foram feitas com polígonos e depois convertidas em *bounding boxes*.

Bounding boxes tem por sua vez delimitar a área do ROI da sua imagem, especificando a *bounding boxes* em relação ao seu centro e sua largura e altura. Podemos representar nossas *bounding boxes* da seguinte maneira

$$x_c = (x_1 + x_2) / 2 \quad (1)$$

$$y_c = (y_1 + y_2) / 2 \quad (2)$$

$$largura = (x_2 - x_1) \quad (3)$$

$$altura = (y_2 - y_1) \quad (4)$$

Em que:

Xc: é a coordenada x do centro da *bounding boxes*; Yc: é a coordenada y do centro da *bounding boxes*; largura: representa a largura da *bounding boxes*; e Altura: representa a altura da *bounding boxes*.

2.3.1. Labelme

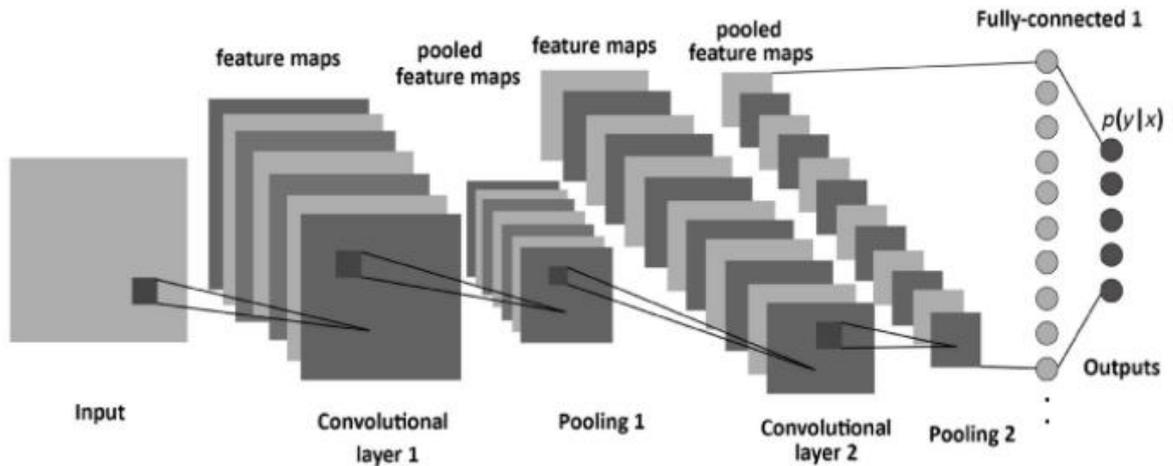
Labelme é uma ferramenta gráfica de anotação de imagens, onde utiliza as formas para polígono, retângulo, círculo, linha e ponto geométricas para anotação de imagens. Utilizado a forma de polígonos para realizar a anotação das imagens deste trabalho e convertidos em *bounding boxes*. O labelme gera um arquivo json com as coordenadas dentro da imagem onde foi realizada a anotação. A anotação realizada no labelme que é gerada uma máscara foram convertidas para *bounding boxes*.

2.4. Redes convolucionais

Uma Rede Neural Convolucional (CNN) é um tipo específico de rede neural normalmente utilizada para classificação, segmentação e processamento de imagens (KHAN, 2019). A estrutura da CNN é baseada em rede de aprendizagem de multicamadas. A fase de aprendizagem consiste em uma combinação das camadas convolucionais e camadas de subamostragens (Pooling). Nas camadas convolucionais, ocorre a operação de convolução, que se aplicação de filtros de peso para extrair características inserida nos pixels da subárea da imagem de entrada, para gerar o diagrama de características da camada. Os diagramas de características são apresentados, a unidade retificadora linear (ReLU). Nas camadas de subamostragem, usando cálculos complexos para reduzir os parâmetros, o diagrama de características é gradualmente reduzido. As camadas de subamostragem podem ser seguida por uma combinação de outras camadas. Em cada camada das CNNs, existem filtros (kernels) sobrepostos em uma determinada área da imagem, considerando a informação da vizinhança do pixel, gerando uma nova matriz de redução (convolução) ou maior (deconvolução) para formar um diagrama de características. Todos esses

recursos obtidos da imagem são propagados da camada de entrada para a camada de saída da rede (KHAN, 2019), como é demonstrado na figura 3.

Figura 3 - exemplo de uma arquitetura de uma CNN



Fonte: Mukherjee (2021)

2.4.1. Camada Convolutiva

A camada convolutiva contém um conjunto de filtros de convolução (kernels) que realizam operações de convolução na imagem. O princípio de funcionamento do kernel de convolução é dividir a imagem em pequenos pedaços, geralmente chamados de campo receptivo. Dividir a imagem em pequenos pedaços ajuda a extrair recursos dela. O kernel utiliza um conjunto específico de pesos para processar imagens, multiplicando seus elementos pelos elementos correspondentes no campo receptivo (COSTA, 2020).

Segundo Costa (2020) A operação de convolução pode ser expressa da seguinte forma:

$$f_l^k(p, q) = \sum_c \sum_{x, y} i_c(x, y) \cdot e_l^k(u, v) \quad (5)$$

Em que:

$ic(x, y)$ é um elemento da imagem de entrada Ic ;

$elk(u, v)$ é o elemento (u, v) do k -ésimo kernel convolucional da l -ésima camada;

u e v representam a u -ésima linha e v -ésima coluna do kernel, respectivamente;

x e y são as coordenadas da imagem;

c é o índice do canal da imagem;

O mapa de característica da k -ésima operação convolucional pode ser expresso como: $Flk = [flk(1,1) \dots, flk(p, q), \dots, flk(P,Q)]$. Em que p é a p -ésima linha e q é a q -ésima coluna de um total de P linhas e Q colunas (COSTA, 2020).

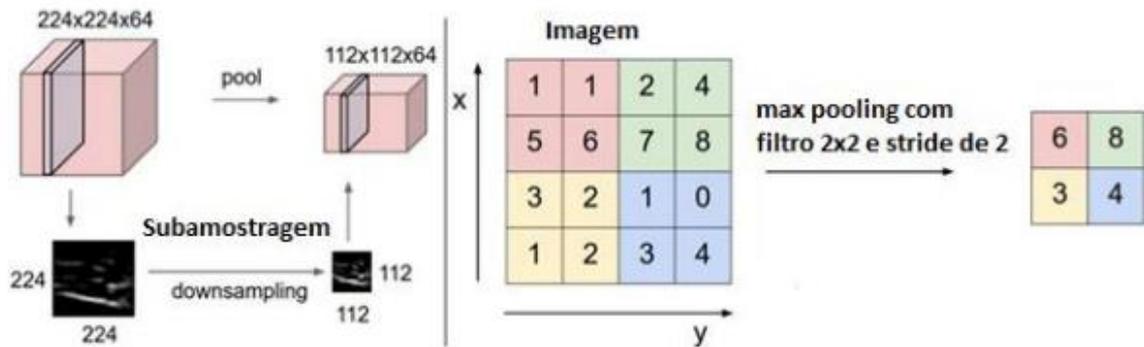
2.4.2. Função de ativação

Uma das formas de acionar a não linearidade as redes neurais é utilizando as funções de ativação não lineares. Hoje em dia, principalmente no contexto de imagens, a mais utilizada é a função *ReLU* (KHAN, 2019). Na literatura, existem diferentes funções de ativação, como *sigmóide*, *tanh*, *maxout*, *SWISH*, *ReLU* e variantes *ReLU*, como *ReLU*, *ELU* e *PReLU* e são usados para incluir combinações de recursos não lineares (KHAN, 2019).

2.4.3. Camada de Subamostragem (Pooling)

Pooling é um meio de redução. Este é uma forma de redução do diagrama de características. Em termos leigos, essa é uma conversão como uma redução na altura e largura da imagem. As operações de subamostragem mais comuns são *pooling* máximo, *pooling* médio e *pooling* de soma, mas o mais comumente usados é o *pooling* máximo (COSTA, 2020). A operação *MaxPooling* (*pooling* máximo) remove o maior elemento de uma determinada área da matriz (considerando o tamanho do *pool* da aplicação). Em seguida, considere o parâmetro de tamanho do passo (semelhante à operação de convolução) para realizar um deslize para aplicar a nova operação (COSTA, 2020), conforme mostrado na Figura 4.

Figura 4 - exemplo de utilização do maxpooling



Fonte: Costa (2020)

2.4.4. Camada de Totalmente Conectada (Full Connected Layer)

Khan (2019) diz que a camada totalmente conectada é usada principalmente no final da rede para classificação, diferente da camada convolucional e da camada de *pooling*, na última camada (totalmente conectada) os neurônios são conectados totalmente aos neurônios da penúltima camada de desconvolução, assim podemos afirmar que a camada totalmente conectada esta conectando com todas as características retiradas da imagem de entrada, e a camada totalmente conectada juntamente a camada de *softmax* são responsáveis pela classificação dos dados.

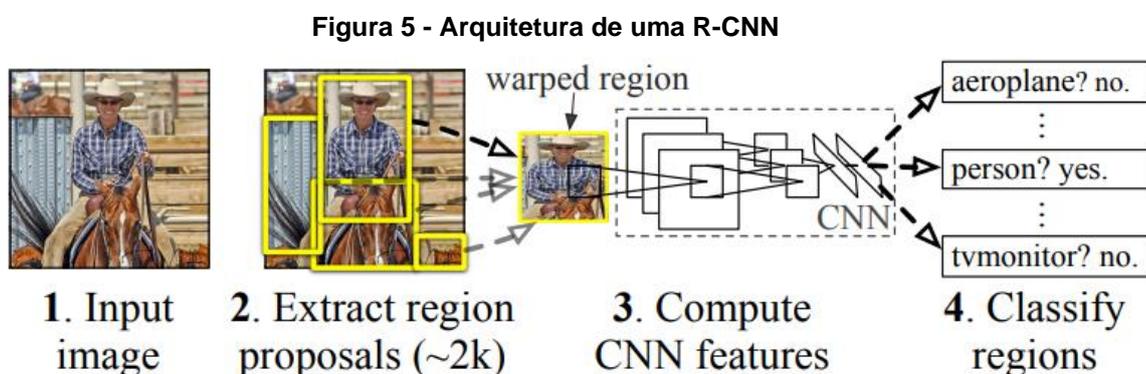
2.5. Detectores de objetos baseados em dois estágios

O nome de detectores de objetos baseados em dois estágios é obtido porque sua arquitetura é dividida em duas etapas, a primeira é encontrar a posição do objeto na imagem é a etapa subsequente e a da classificação.

Um dos métodos mais importantes no âmbito da detecção de objetos é a CNN baseado em região (R-CNN), que é derivado do trabalho de Girshick et al. (2014). Uma das principais características deste tipo de sistema é que não utiliza o método tradicional de camadas deslizantes, podendo extrair regiões de interesse por meio de busca seletiva (*Selective Search*) (RESTANI, 2019).

O princípio de funcionamento do R-CNN é o seguinte: Primeiro, durante a chamada proposta de região, a rede extrai regiões que considera relevantes. Em seguida, os recursos de cada uma dessas regiões são extraídos por meio da CNN

(AlexNet, no caso da R-CNN). Depois, há a classificação dessas áreas (RESTANI, 2019), como é demonstrado na figura 5 abaixo.

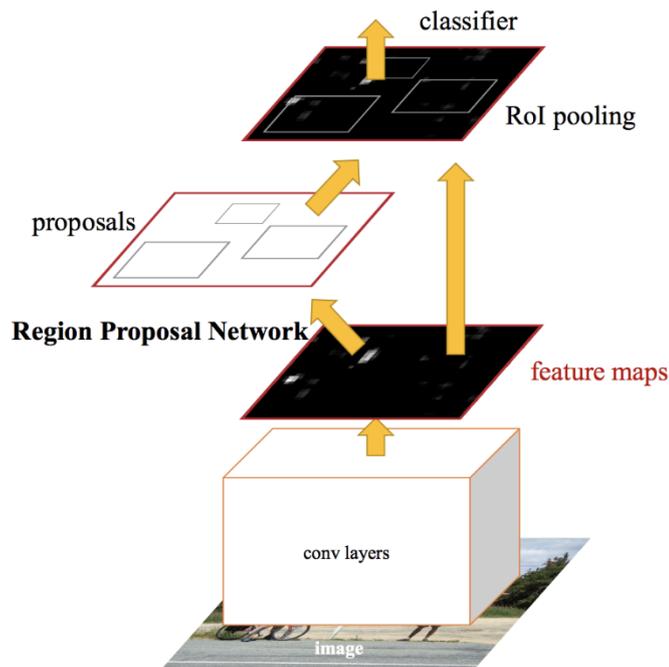


Fonte: Girshick et al. (2014)

A Faster R-CNN foi desenvolvido como um detector de categoria (suporte) em cenários de detecção de objetos de várias categorias (REN; He; GIRSHICK; Sun, 2017). O desenvolvimento do Faster R-CNN (REN; He; GIRSHICK; Sun, 2017) resolve os problemas de desempenho não resolvidos no Fast R-CNN. O Fast R-CNN só passa a imagem original para o modelo CNN uma vez onde esse modelo é pré-treinado, independentemente da proposta de cada região, o algoritmo de Busca Seletiva (Selective Search) é calculado com base na saída do mapa de características pela camada anterior, Selective Search e o algoritmo de interesse de descoberta de ROI(s) (GIRELLI; CORSO, 2020). Uma das principais evoluções é a introdução do algoritmo Rede de Proposta Regional (Region Proposal Network - RPN) (REN; He; GIRSHICK; Sun 2017) em vez da busca seletiva (Selective Search), o que melhora muito a velocidade de detecção. De acordo com Gandhi (2018), a rede proposta de região (RPN) é baseada em uma CNN puramente convolucional para realizar operações de caixa delimitadora (bounding box) e atribuição de pontuação. Essa rede tem duas camadas convolucionais adicionais, uma para criar um mapa de características como em uma rede CNN e a segunda camada serve para criar um mapa de valores de objetos, e seus limites correspondentes, a RPN apresenta o conceito de âncora. Âncora é usada para criar o ponto que fica no centro da janela deslizante com base nas coordenadas da camada de regressão (reg layer). O número de pontos de ancoragem criados pode depender das medidas de altura e da largura (em pixels) da imagem. Em outras palavras, Faster R-CNN envolve associar um RPN

com Fast R-CNN, desta forma, o tempo de busca é otimizado pela região de interesse. O Faster R-CNN possui duas etapas: Na primeira, a rede RPN propõe as possíveis localizações dos objetos a serem detectados. A segunda etapa é o classificador Fast RCNN, que utiliza as regiões propostas pelo RPN para detectar e desenhar a caixa delimitadora e, após obter o mapa de características fornecido pelo RPN, gera a proposta de região onde o possível objeto está localizado. Após esta etapa, o mapa de regiões passa pela camada de ROI pooling e encaminha sua saída para classificação. Todo o sistema é apenas uma rede e unificada com objetivo de detecção de alvos, na qual o módulo RPN informa ao módulo Fast R-CNN onde encontrar o objeto a ser detectado (GIRELLI; CORSO, 2020). A figura 6 demonstra a arquitetura da Faster RCNN.

Figura 6 - Demonstração da arquitetura da Faster R-CNN



Fonte: GIRELLI e CORSO (2020).

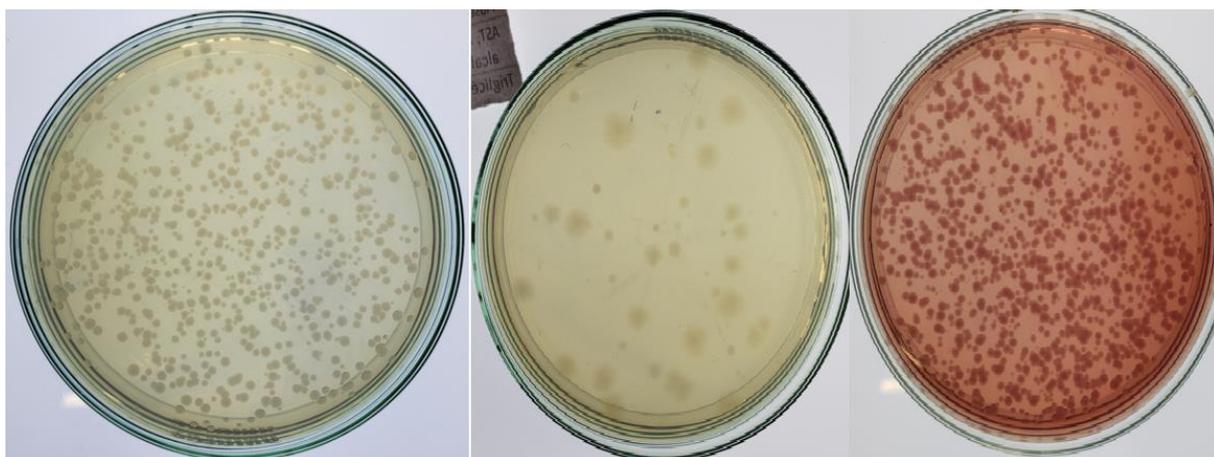
WANG et al. (2018) afirma que as existem vantagens em CNN(s) e suas variantes comparado com métodos tradicionais de aprendizado de máquina, o mesmo cita que, técnica de aprendizado profundo (CNN e suas variantes) leva a uma melhor precisão de classificação para localização e segmentação celular, contagem de colônias bacterianas, detecção de células cancerosas e etc.

3. MATERIAIS E MÉTODO

3.1. Amostras

Para a pesquisa realizada nesta monografia, foram utilizadas imagens de colônias bacterianas cultivadas no LBMic do IPTESP, obtidas por meio de smartphone no Laboratório de Computação Científica da Pontifícia Universidade Católica de Goiás. Foram coletadas um total de 31 imagens, onde algumas das imagens possuem poucas colônias de bactérias e outras imagens que possuem muitas colônias de bactérias, como demonstrado na figura 7 abaixo:

Figura 7 – exemplo de amostras cultivadas pelo IPTESP



Fonte: IPTESP (2020).

3.2. Método proposto

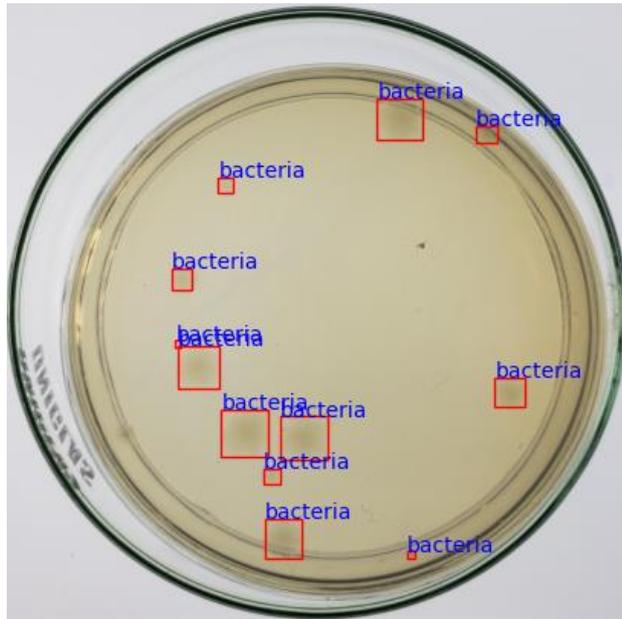
No método proposto, uma abordagem Faster R-CNN foi desenvolvido e usado o pytorch, foi utilizado o *framework* MMDetection. O MMDetection é um *framework* de detecção de objetos de código aberto. É uma parte do projeto OpenMMLab, o *framework* suporta diretamente estruturas de detecção populares e contemporâneas, por exemplo, Faster RCNN, Mask RCNN, RetinaNet, etc (CHEN, Kai et al. 2019). Utilizado conjunto de dados em formato MS-COCO. O conjunto COCO é publicamente disponível, é uma forma de colocar as anotações em um *dataset*, foi realizado as

anotações para o treinamento do modelo para definir a *label*: bactéria. Todas as anotações foram feitas usando o software de código aberto Labelme (WADA, K, et al, 2020). Posterior as *labels* são convertidas para *labels* no formato MS-COCO *dataset* (LIN et al, 2015), pois o modelo utilizado está adaptado para trabalhar com essa anotação.

Para cada imagem antes de ser inserida no modelo é realizado um pré-processamento, é realizado um redimensionamento na imagem, onde a imagem ficara com a resolução de 1333, 704 largura, altura, esse redimensionamento foi realizado pois os recursos computacionais são limitados.

A *Label* bacteria foi anotada para cada imagem dentro do nosso conjunto de dados, na figura 8 podemos ver a representação da imagem anotada com o rótulo proposto e com a *bounding boxes* ao redor do ROI. O RPN classifica as caixas de âncora e propõe os aqueles mais provavelmente contendo objetos que seria o ROI, a saída de um RPN é uma variedade de caixas que serão examinados por um classificador e um regressor eventualmente, verificar a ocorrência de objetos para localização e detecção. As quatro proporções de aspecto (*ratios*): 0.5, 1.0, 0.3 e 0.7 que correspondem à relação largura / altura de uma caixa de âncora, foram usadas, essas métricas de ancoras justamente pois o ROI em algumas das imagens da base tem uma dimensão mínima, então os parâmetros da ancora foram reduzidos para que tenha coerência com o problema. Após RPN, as regiões propostas foram obtidas com mapas de recursos da CNN de diferentes tamanhos.

Figura 8 - exemplo de figura anotada com *bounding boxes* e *labels*



Fonte: IPTESP (2020) adaptada pelo autor.

O pool ROI foi usado para simplificar o problema, reduzindo os mapas de recursos da CNN no mesmo tamanho. Os recursos de saída da camada ROI *Pooling* foram convertidos em uma matriz unidimensional e alimentado as camadas totalmente conectadas (fc) para classificação e regressão operações em que a CNN encontra quais/quantas colônias existem para aquela imagem. O ramo de classificação de *Softmax* camada no modelo fornece probabilidades para cada ROI e a previsão é feita nessa fase. Na *bounding boxes* ramo de regressão gera quatro deslocamentos de regressão de caixa de âncora k onde:

$$i = x, y, w \text{ e } h ; \quad (6)$$

em que:

(x, y) representa o canto superior esquerdo;

w, h denotam a altura e a largura da caixa de âncora.

Para metrificarmos o nosso modelo utilizamos o parâmetro de perda (Loss), A localização da Loss escolhida foi a smooth L1 loss (GIRSHICK,2015) é dada na Equação (7), multitarefa conjunta para cada ROI é dada pela combinação das perdas de classificação e localização como mostrado na Equação (8)

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} smooth_{L1}(t_i^u - v_i) \quad (7)$$

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \quad (8)$$

onde, L_{cls} é a perda de classificação e L_{loc} é a perda de localização, λ é o coeficiente de peso para equilibrar as duas perdas.

O parâmetro de máximo de épocas foi preenchido com 22, de forma de teste, foi observado por meio da *loss* que o modelo convergia a partir da 8 época.

3.3. Treinamento

O nosso conjunto de dados é composto por 30 imagens, onde as 25 primeiras imagens foram separadas para o conjunto de dados de treino e as últimas 5 imagens para o conjunto de dados de teste. Dentro do modelo foi dividido os dados de treino da seguinte forma, para treino 18 imagens são utilizadas e para a validação foram utilizadas 7 imagens, como demonstra na tabela abaixo.

Tabela 1 - Divisão de dados de treino e teste

	Nº imagens	%
Treino	18	70
Teste	7	30

Fonte: Autoria própria (2021)

4. RESULTADOS

Realizado o treinamento do modelo proposto, para a detecção de colônia de bactérias nas imagens da base de dados, as imagens foram classificadas com base na *label* bactéria, que foi anotada durante o processo do trabalho. O resultado demonstrado neste foi avaliado de forma qualitativa, a partir de observações durante a criação do trabalho.

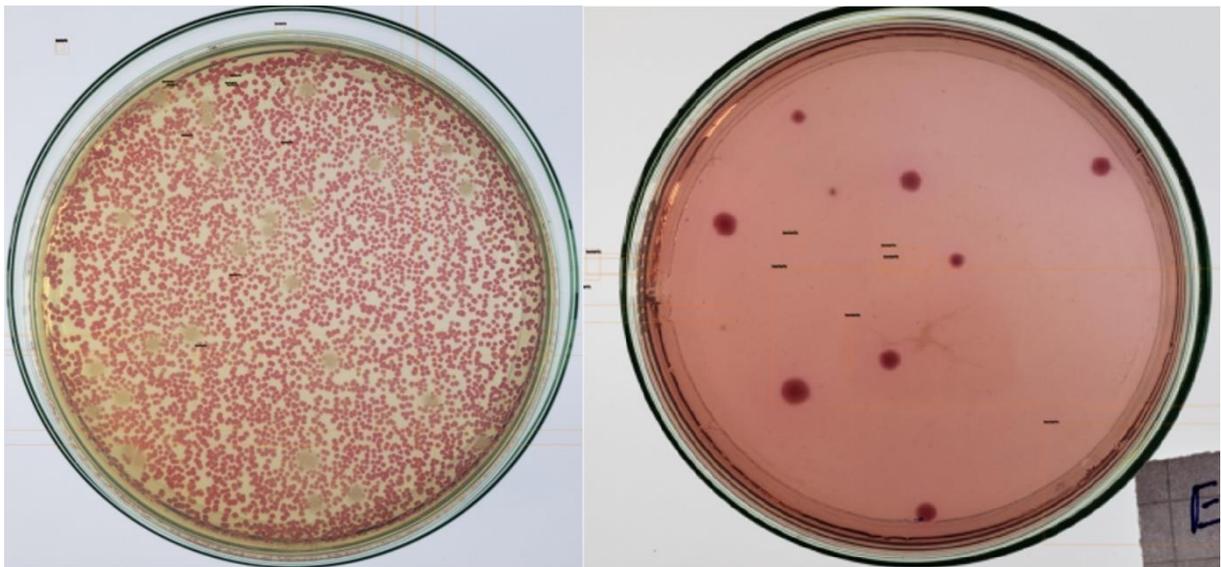
Foi observado que teve uma grande discrepância de resultado real e resultado obtido pelo modelo. Foram realizados 2 modelos alterando os parâmetros de ancoragem da RPN, esses parâmetros foram alterados de acordo com os testes realizados e observações feitas durante o treinamento do nosso modelo.

Modelo 1 foi treinado com o parâmetro de ancora *ratio*: (0.41, 0.7, 1.2), foi utilizado vários *ratios* para validarmos qual seria o melhor para o nosso problema. Já o modelo 2 teve o parâmetro da ancora *ratio*: (0.35, 0.76, 1.0).

Para realizar a contagem das colônias de bactérias o parâmetro retornado do modelo, que é um vetor de coordenadas das *bounding boxes* o tamanho desse vetor indica que o modelo previu essa quantidade de colônias na amostra.

Para os dados de treinamento é possível analisar o resultado observando que não foram identificadas uma quantidade de bactérias considerável nas amostras, dado que algumas amostras possuem uma quantidade de colônias de bactérias na casa de milhoar, então podemos concluir que não foi obtido um resultado satisfatório como demonstrados nas figuras 9 e 10

Figura 9 - amostras de treino com a inferência do modelo 1

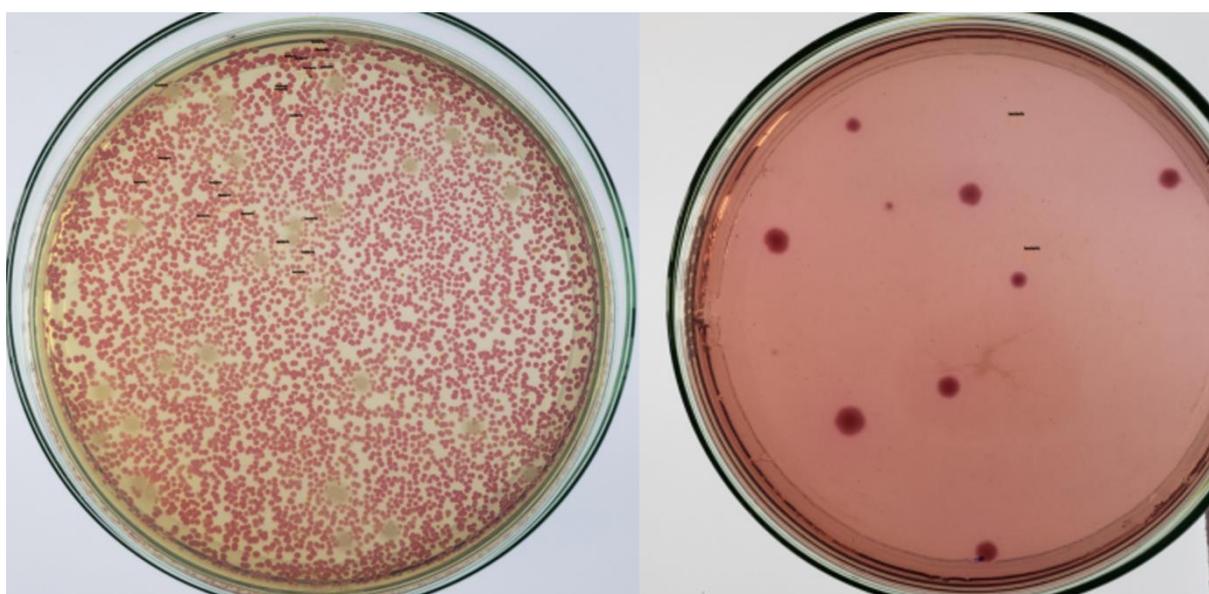


(a)

(b)

Fonte: IPTESP (2020) adaptada pelo autor.

Figura 10 - amostras de treino com a inferência do modelo 2



(a)

(b)

Fonte: IPTESP (2020) adaptada pelo autor.

Abaixo segue a tabela mostrando a quantidade real de colônias da amostra e a quantidade de colônias que o modelo propôs para a amostra.

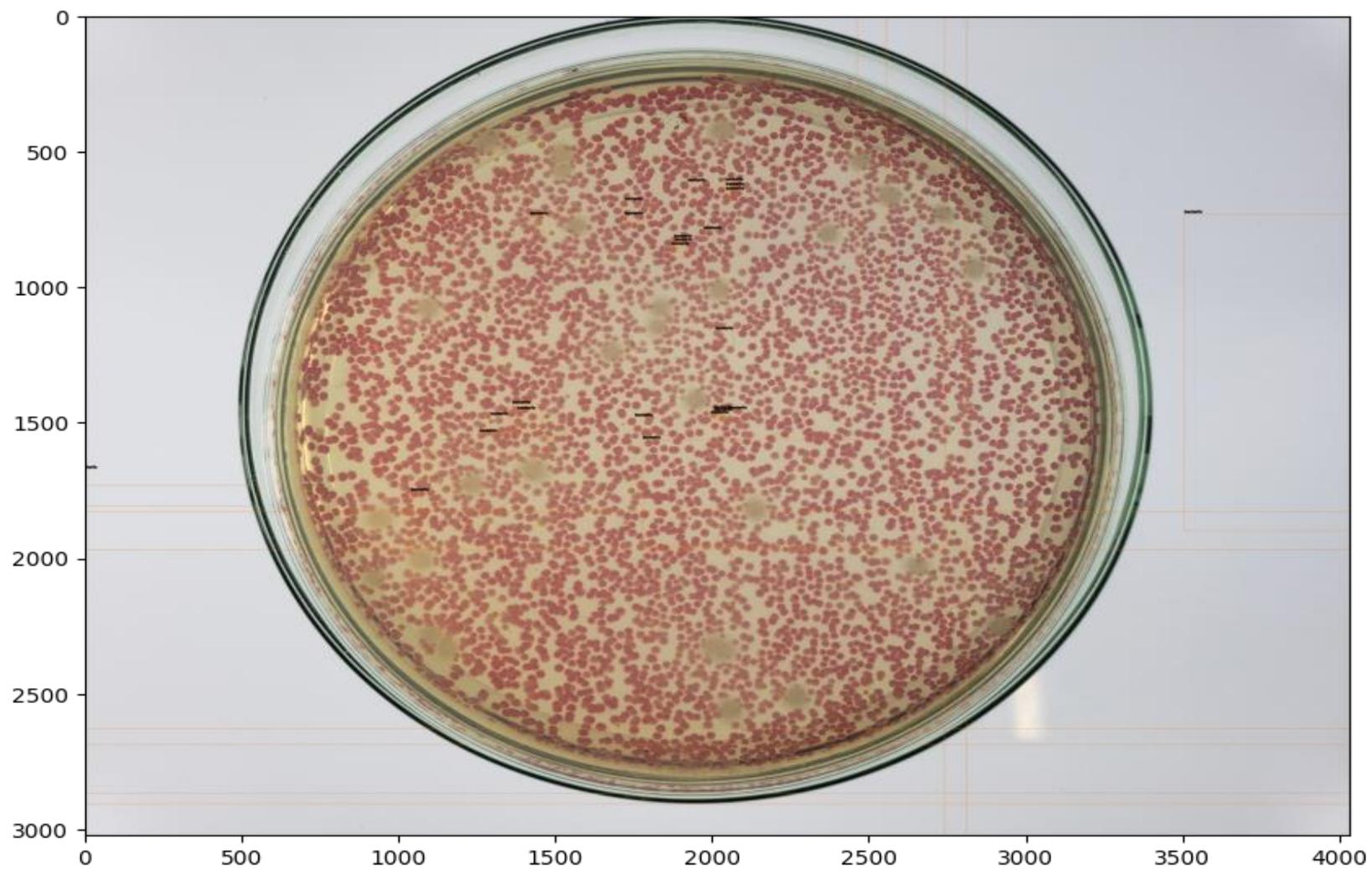
Tabela 2 - representação de resultado para amostras do modelo 1 e modelo 2

Modelo	Amostras	Nº de colônias reais da amostra	Nº de colônias previstas pelo modelo
Modelo 1	9 a	2946	49
Modelo 1	9 b	9	21
Modelo 2	10 a	2946	21
Modelo 2	10 b	9	2

Fonte: Autoria própria (2021).

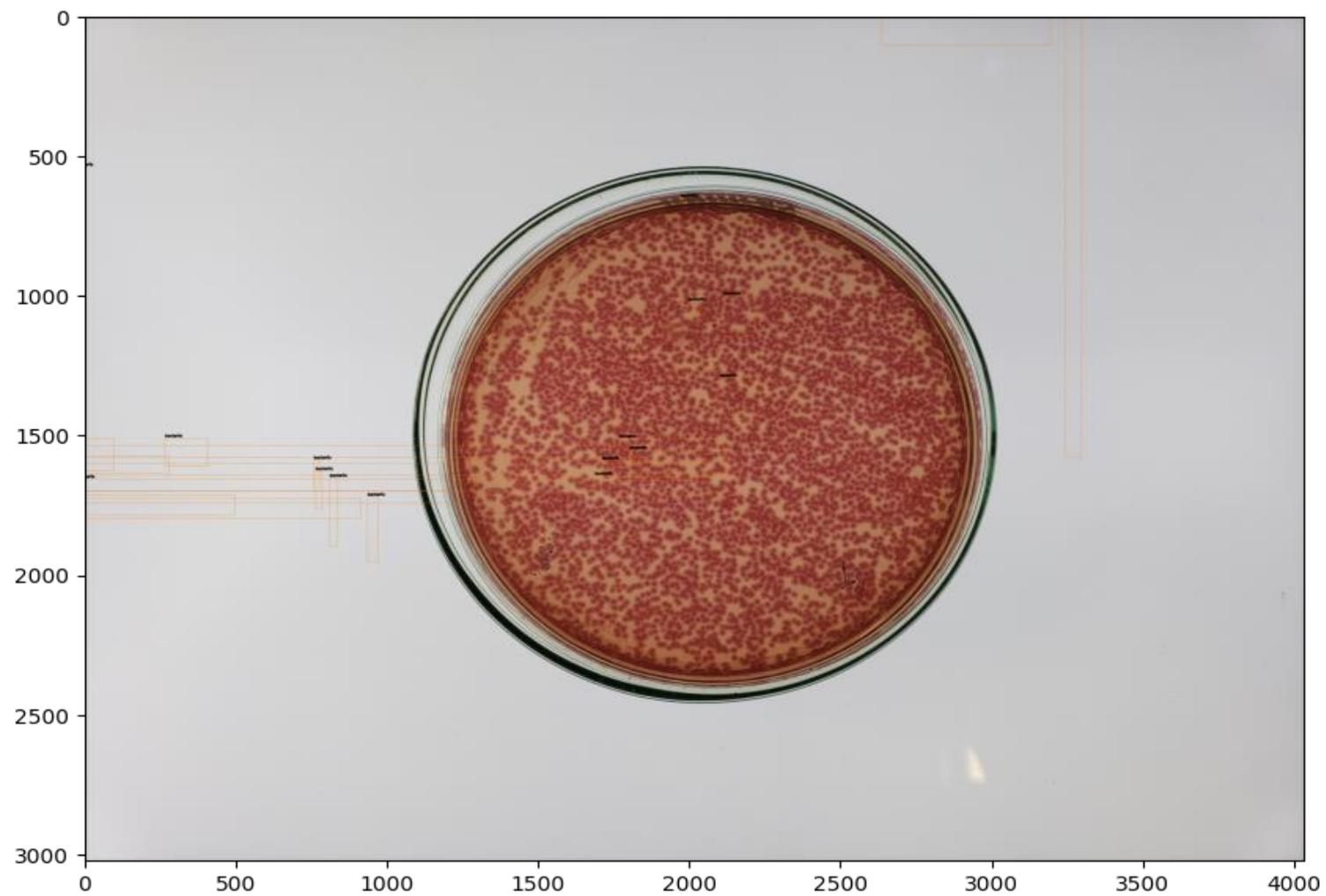
Foi realizada a inferência do modelo para 5 amostras para as quais nunca foi vista pelo modelo, nas figuras 11, 12, 13, 14 e 15 conseguimos observar que não teve um acerto eficiente em relação ao real para as imagens para o modelo 1.

Figura 11 - inferência do modelo 1 no *dataset* de treino



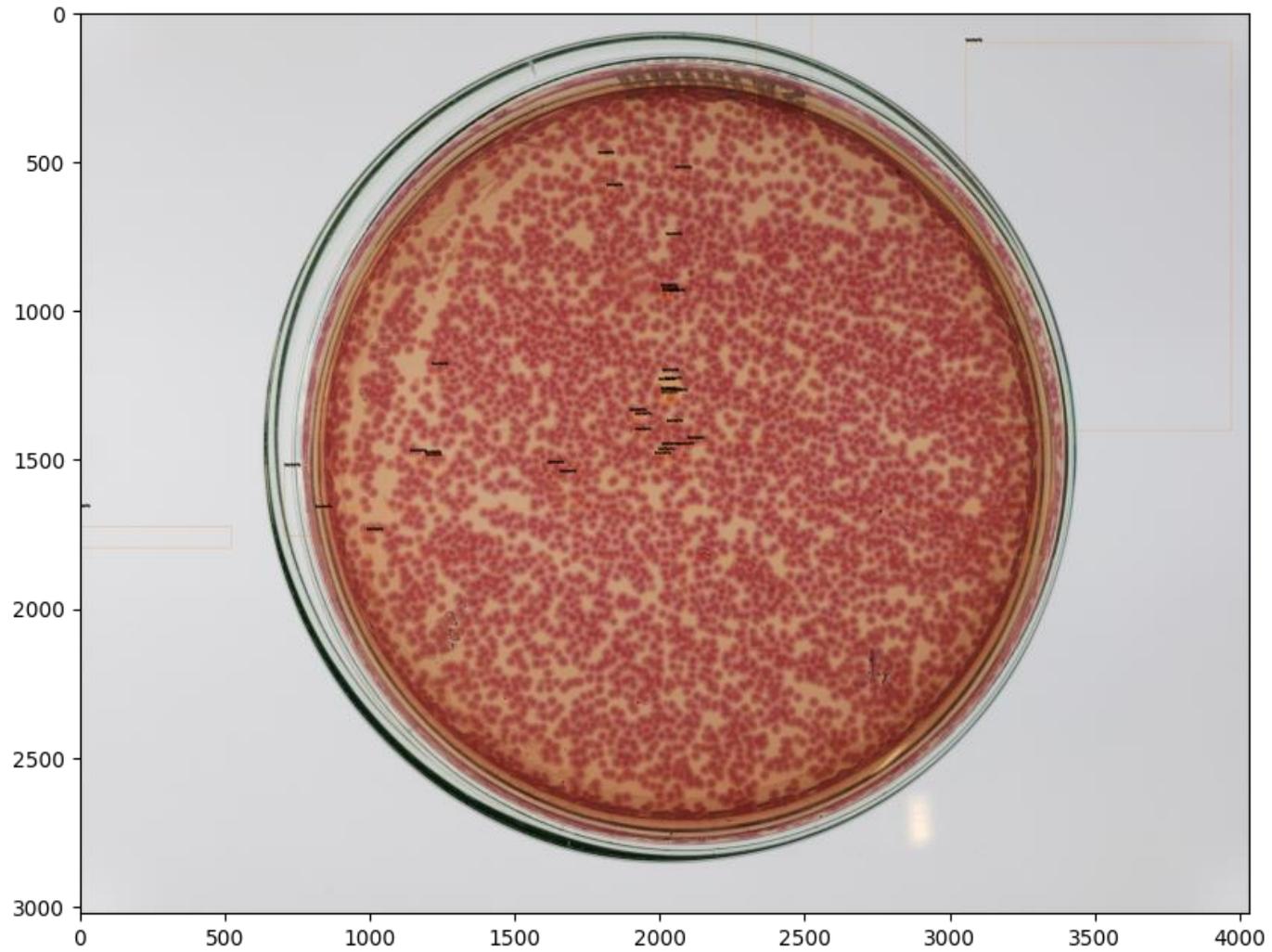
Fonte: IPTESP (2020) adaptada pelo autor.

Figura 12 - inferência do modelo 1 no *dataset* de treino



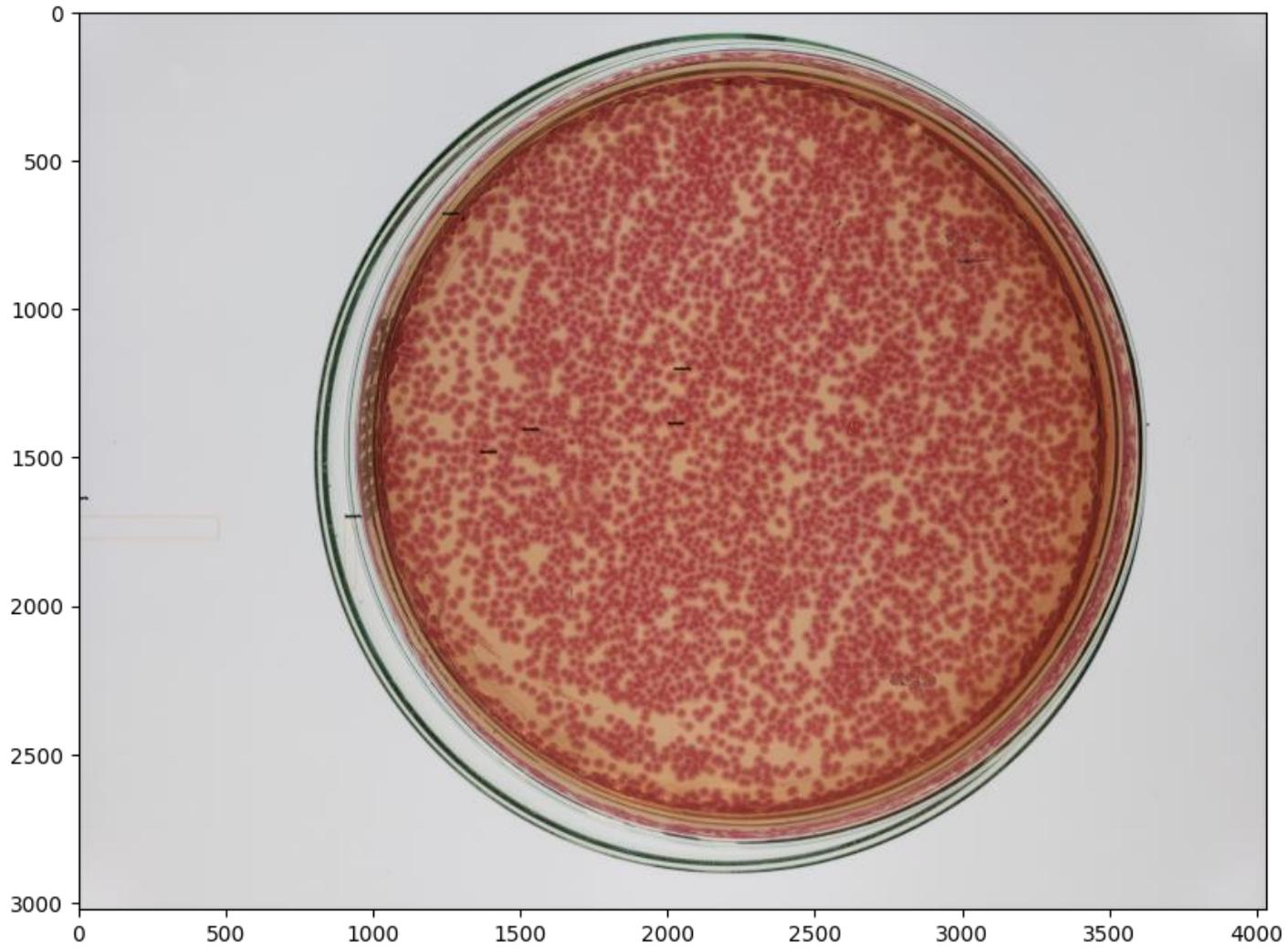
Fonte: IPTESP (2020) adaptada pelo autor.

Figura 13 - inferência do modelo 1 no *dataset* de treino



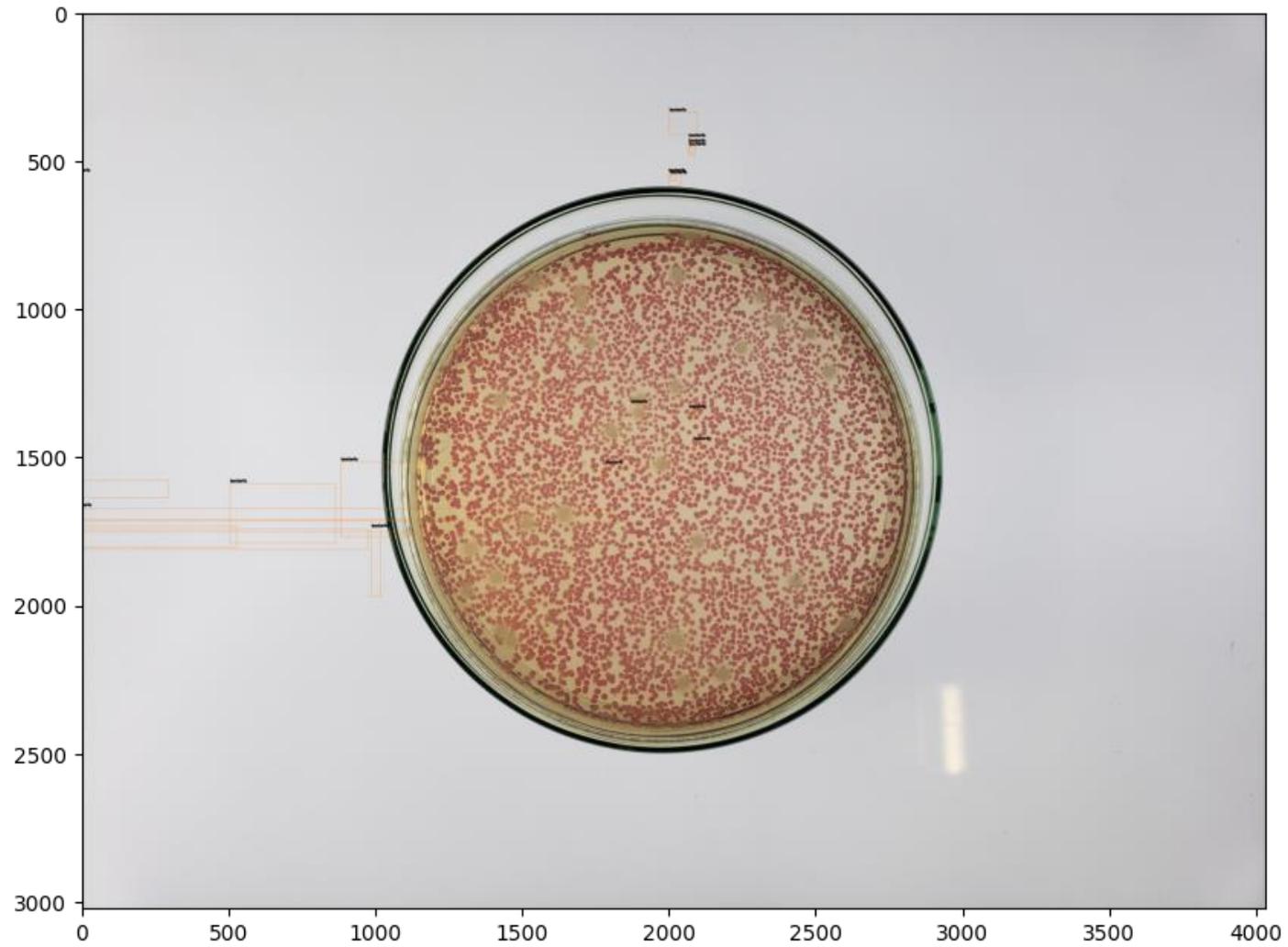
Fonte: IPTESP (2020) adaptada pelo autor.

Figura 14 - inferência do modelo 1 no *dataset* de treino



Fonte: IPTESP (2020) adaptada pelo autor.

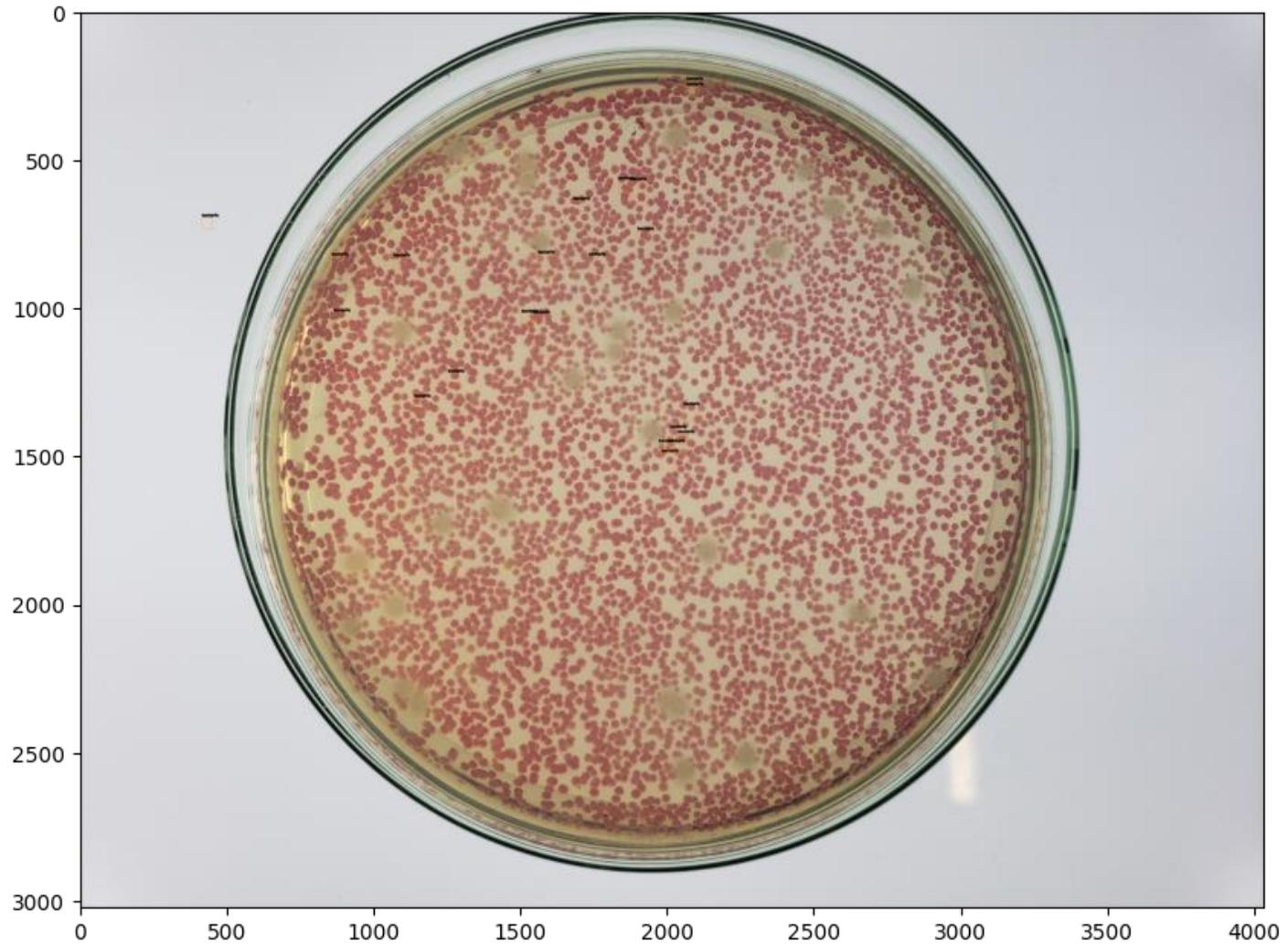
Figura 15 - inferência do modelo 1 no *dataset* de treino



Fonte: IPTESP (2020) adaptada pelo autor.

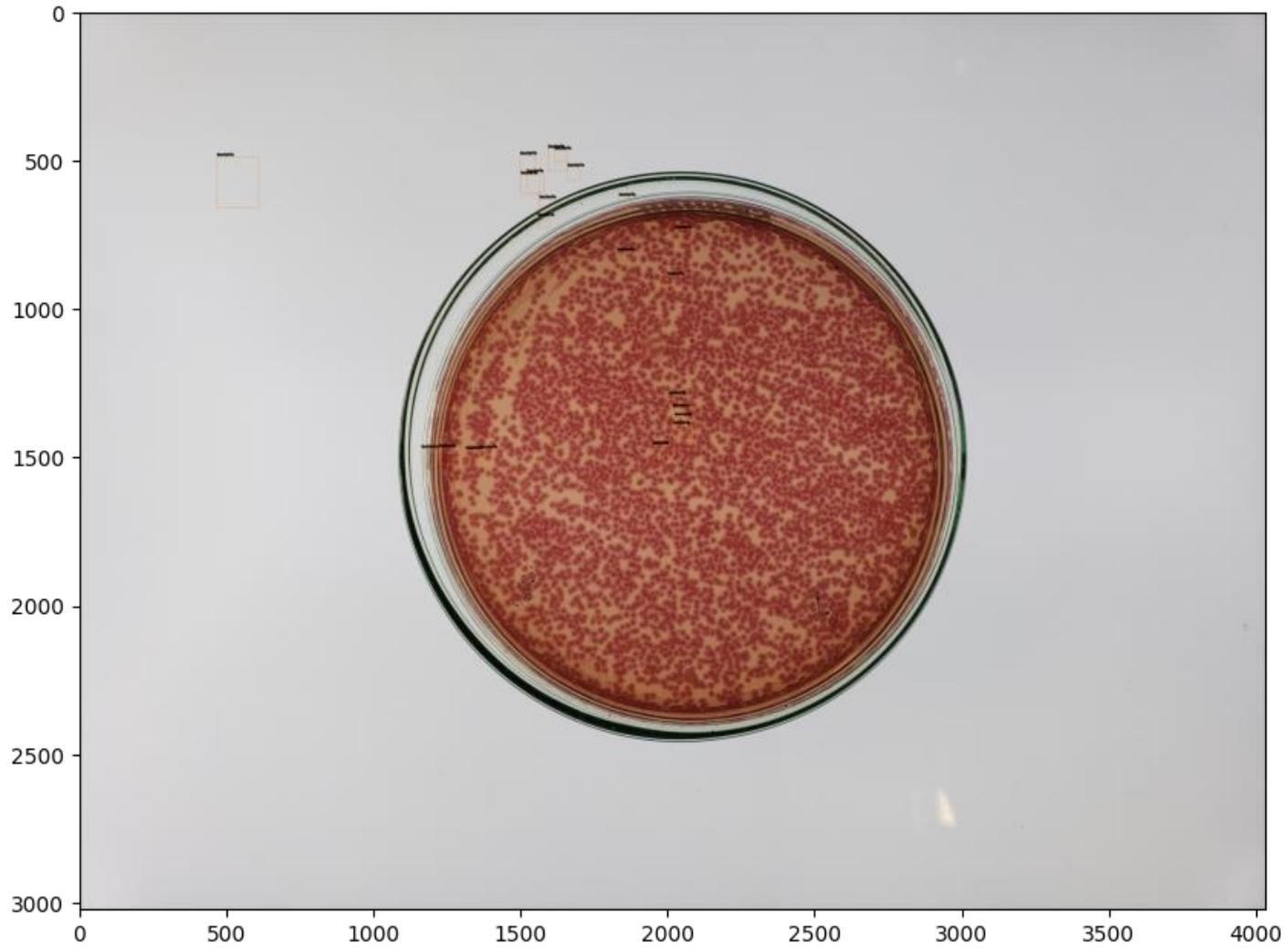
Podemos observar nas figuras 16, 17, 18, 19 e 20 que o modelo 2 também não teve um acerto eficiente em relação ao real para as imagens.

Figura 16 - inferência do modelo 2 no *dataset* de treino



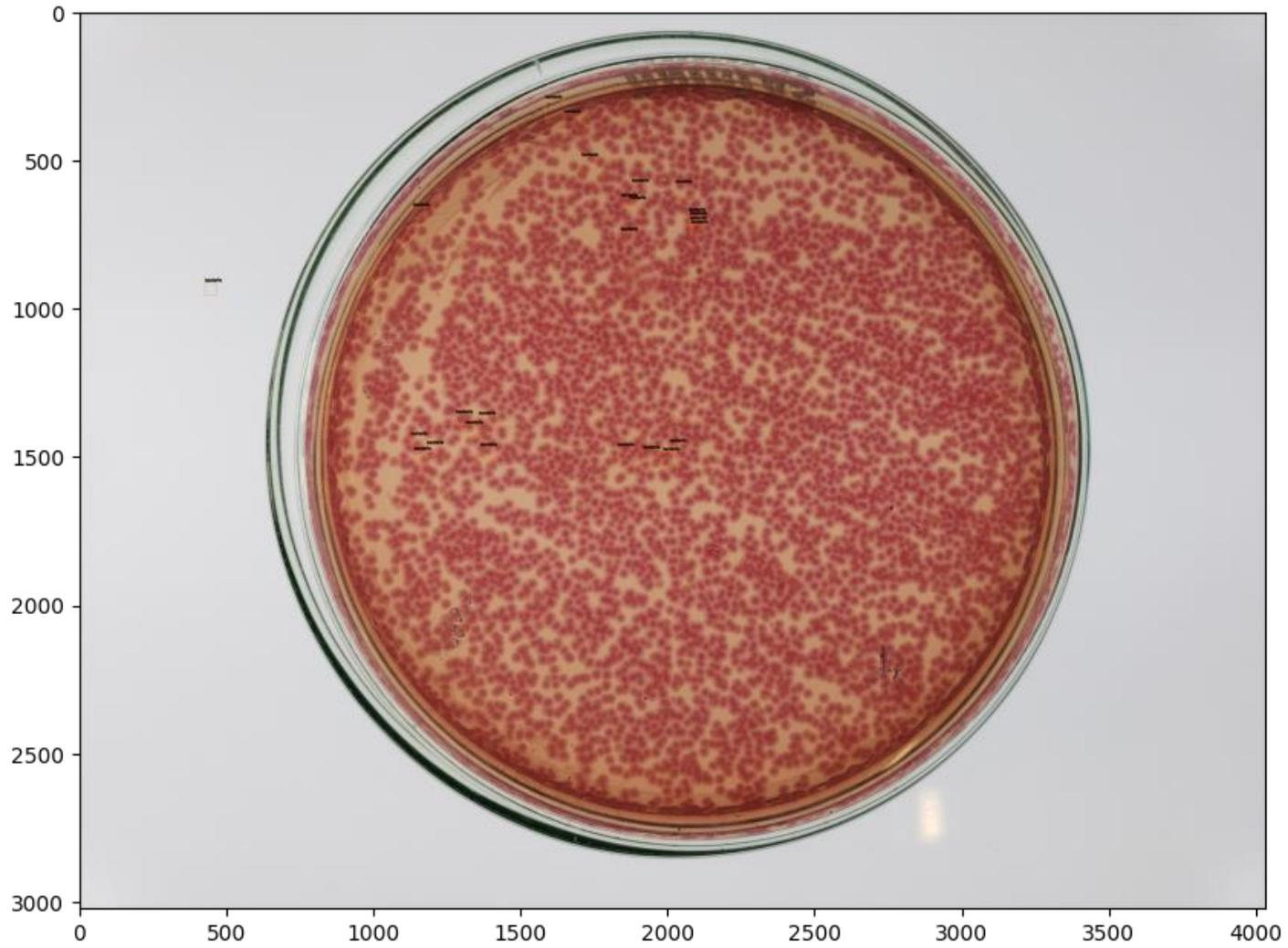
Fonte: IPTESP (2020) adaptada pelo autor.

Figura 17 - inferência do modelo 2 no *dataset* de treino



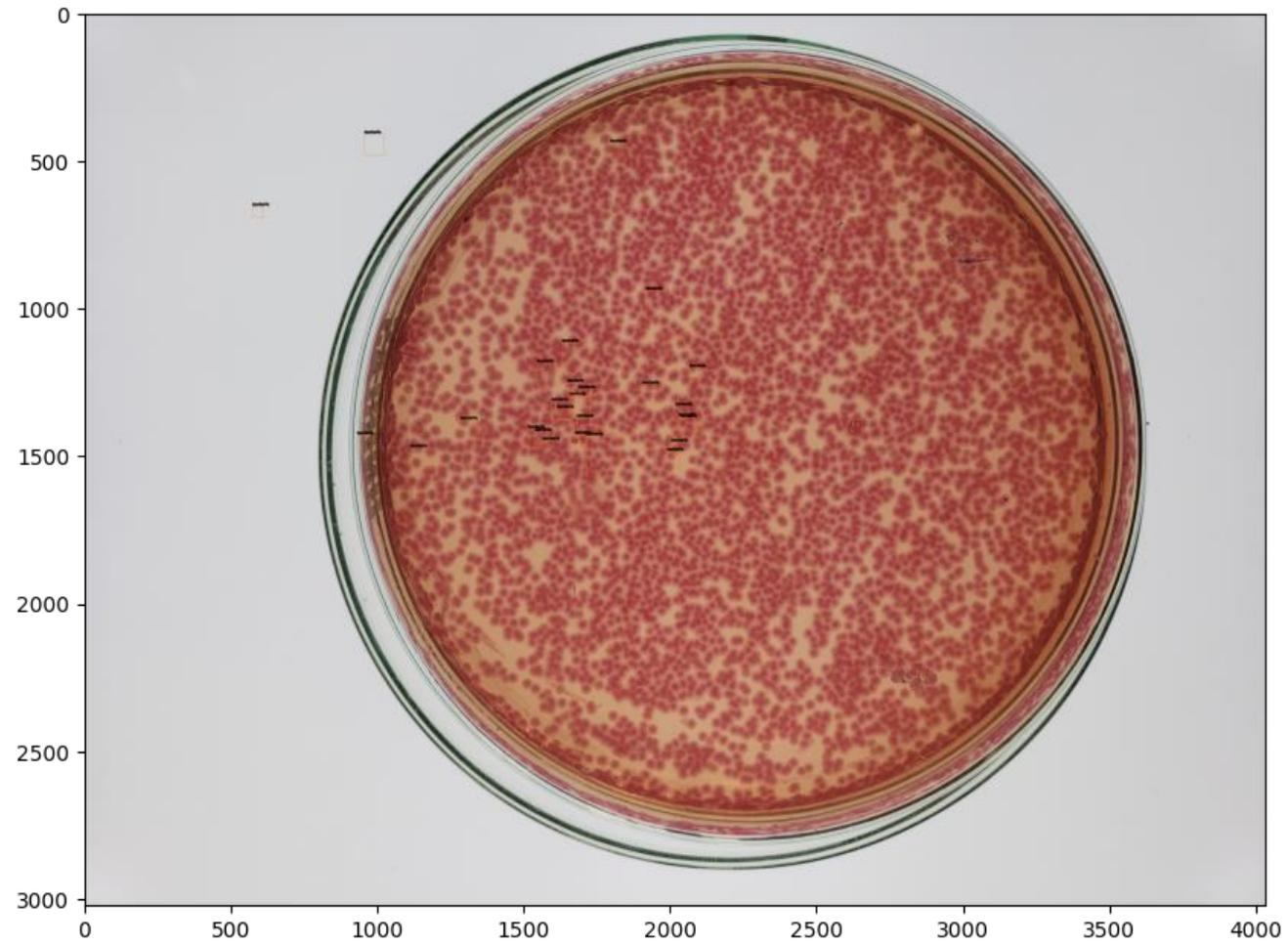
Fonte: IPTESP (2020) adaptada pelo autor.

Figura 18 - inferência do modelo 2 no *dataset* de treino



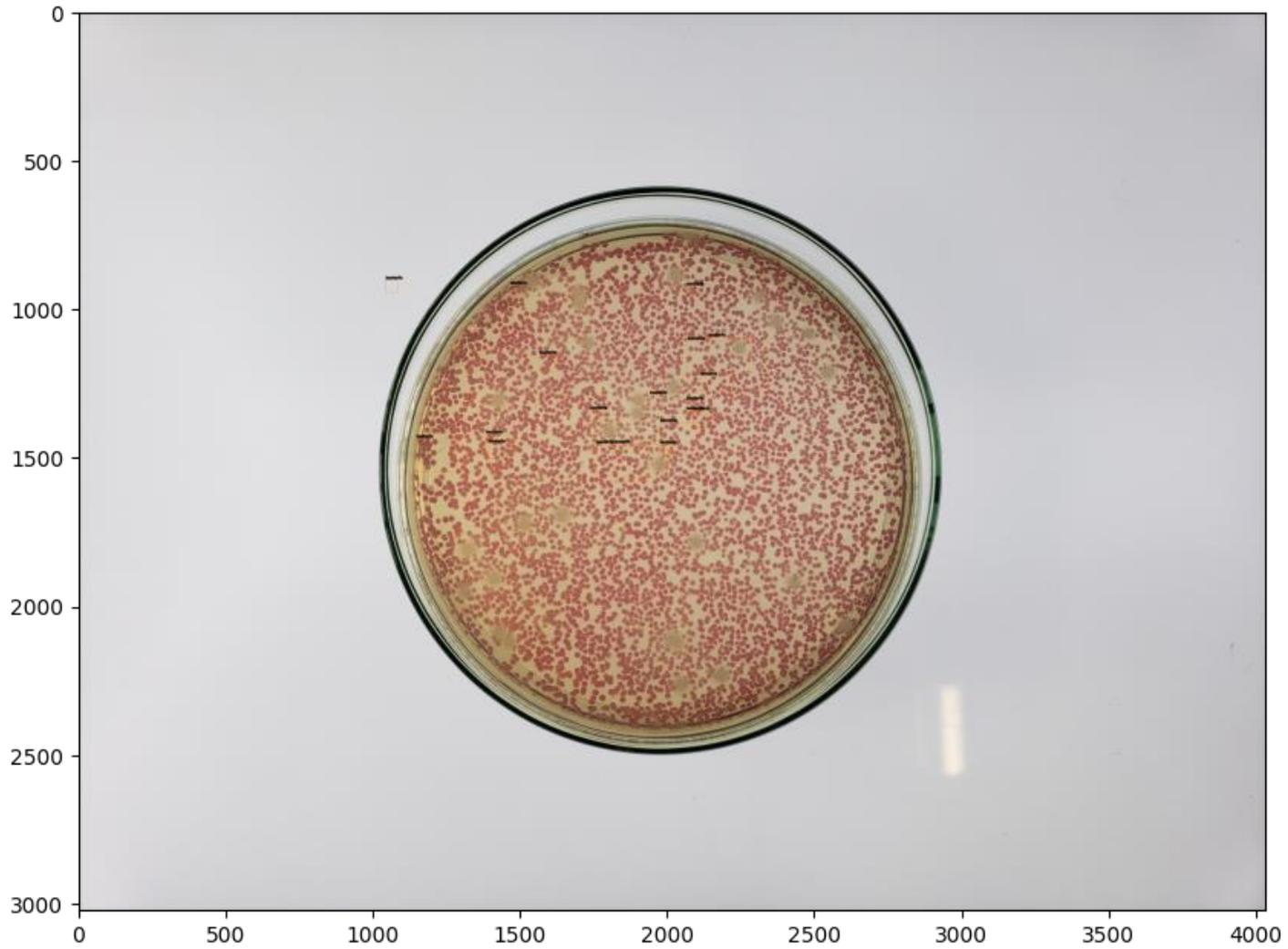
Fonte: IPTESP (2020) adaptada pelo autor.

Figura 19 - inferência do modelo 2 no *dataset* de treino



Fonte: IPTESP (2020) adaptada pelo autor.

Figura 20 - inferência do modelo 2 no *dataset* de treino



Fonte: IPTESP (2020) adaptada pelo autor.

Segue abaixo tabela com o total de colônias encontradas para as amostras acima.

Tabela 3 – Resultado da inferência dos modelos no *dataset* de treino

Modelo	Amostras	Nº de colônias previstas pelo modelo
Modelo 1	Figura 11	60
Modelo 1	Figura 12	91
Modelo 1	Figura 13	74
Modelo 1	Figura 14	43
Modelo 1	Figura 15	72
Modelo 2	Figura 16	22
Modelo 2	Figura 17	22
Modelo 2	Figura 18	25
Modelo 2	Figura 19	27
Modelo 2	Figura 20	19

Fonte: Autoria própria (2021).

Com os resultados listados acima pode-se concluir que os dois modelos tiveram uma eficiência insatisfatória para a detecção de objetos, como pode-se observar nas imagens de treinamento foi proposto pelo modelo várias regiões que não possuía colônias de bactérias, dado isso podemos concluir que o modelo não teve eficiência para nosso problema em questão.

5. CONCLUSÃO

O resultado desse trabalho foi o desenvolvimento de um modelo para detectar colônias de bactérias e realizar suas contagens. Onde os dados de testes para o modelo foram separados em 70% para treino e 30% para teste. O processo de geração de um modelo de detecção de objetos não é uma tarefa simples, requer análise e vários testes, para chegar a um resultado satisfatório. Como visto nesse

trabalho, o modelo gerado não atendeu as expectativas apresentando problemas e se mostrando ineficiente para o resultado esperado. Isso forçou o autor a executar os processos várias vezes a fim de encontrar um modelo que atendesse às necessidades propostas, porém a abordagem proposta não teve melhorias na classificação do modelo. Foi observado que no pré-processamento da imagem para dar entrada para o modelo teve uma perda considerável de informações das amostras, esse pré-processamento foi realizado pois existe uma limitação de custo computacional, e com isso afetando o resultado deste trabalho. Também foi observado que a pequena quantidade de dados que a base possui é um problema, pois assim prejudica o treinamento do modelo proposto pois não possui dados suficientes para o treinamento do modelo. Em estudos futuros, pretende-se usar uma técnica de divisão da base de dados onde 1 imagem irá virar n imagens, assim tendo mais dados e conseqüentemente menos perda de informações para o treinamento do modelo proposto, também pretende-se testar mais modelos baseados em detecção de objetos de 1 e/ou 2 estágios, como R-CNN e/ou Fast R-CNN.

6. REFERÊNCIAS

ZHANG, Chengcui et al. **An Automated Bacterial Colony Counting System**. IEEE Computer Society, [S. l.], p. 233, 14 dez. 2018. DOI DOI 10.1109/SUTC.2008.50. Disponível em: <<<https://cpb-us-w2.wpmucdn.com/sites.uab.edu/dist/3/66/files/2020/01/SNUTS2008.pdf>>>. Acesso em: 14 dez. 2021.

FERRARIA, Alessandro; LOMBARDIA, Stefano; SIGNORONIA, Alberto. **Bacterial Colony Counting with Convolutional Neural Networks in Digital Microbiology Imaging. Pattern Recognition**, [S. l.], v. 61, p. Pages 629-640, 30 jan. 2016. DOI <https://doi.org/10.1016/j.patcog.2016.07.016>. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0031320316301650>>. Acesso em: 17 nov. 2021.

CLARKE, Matthew L. et al. **Low-Cost, High-Throughput, Automated Counting of Bacterial Colonies**. 2010. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2909336/>>. Acesso em: 17 nov. 2021

NIYAZI, Maximilian; NIYAZI, Ismat; BELKA, Claus. **Counting colonies of clonogenic assays by using densitometric software**. 2007. Disponível em: <<https://link.springer.com/article/10.1186/1748-717X-2-4>>. Acesso em: 17 nov. 2021.

BAXES, Gregory. **Digital Image Processing: Principles and Applications**. 1. ed. New York: John Wiley & Sons, 1994.

K. R. Castleman; **Digital Image Processing**. Prentice-Hall, Inc, 1995.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais: Princípios, algoritmos e aplicações**. São Paulo: THOMSON Learning, 2008.

MATLAB. Mathworks. **ROI-Based Processing**. Disponível em: <<https://www.mathworks.com/help/images/roi-based-processing.html>>. Acesso em: 29 nov 2021.

MATLAB. Mathworks. **Create ROI Shapes**. Disponível em: <<https://www.mathworks.com/help/images/roi-creation-overview.html>>. Acesso em: 29 nov 2021.

WADA, K, et al. **Labelme**. 2020. Disponível em: <<https://github.com/wkentaro/labelme>> Acesso em: 29 nov 2021.

LIN, Tsung-Yi et al. **Microsoft COCO: Common Objects in Context**. Cornell University, [S. l.], p. 1-15, 21 fev. 2015. Disponível em: <https://arxiv.org/abs/1405.0312>. Acesso em: 29 nov. 2021.

KHAN, A.; SOHAIL, A.; ZAHOORA, U.; SAEED, A. (2019). **A Survey of the Recent Architectures of Deep Convolutional Neural Networks**.

MUKHERJEE, P. **Convolution Neural Networks vs Fully Connected Neural Networks**. Disponível em: <<https://medium.datadriveninvestor.com/convolution-neural-networks-vs-fully-connected-neural-networks-8171a6e86f15>> Acessado em: 01 de dez 2021.

COSTA, FERNANDA. **SEGMENTAÇÃO SEMÂNTICA DE ÁREAS DESMATADAS UTILIZANDO REDES NEURAS CONVOLUCIONAIS NO SUL DA AMAZÔNIA LEGAL**. MANAUS, 2020. Disponível em: <https://tede.ufam.edu.br/bitstream/tede/8028/13/Disserta%C3%A7%C3%A3o_FernandaCaetano_PPGEE.pdf>. Acesso em: 2 dez. 2021.

Ren S, He K, Girshick R, Sun J. **Faster R-CNN: Towards real-time object detection with region proposal networks**. IEEE Trans Pattern Anal Mach Intell 2017; 39:1137-49.

GIRELLI, Cassio; CORSO, Leandro Luís. **Deteção de falhas em contentores plásticos utilizando Redes Neurais Convolucionais**. 2020. Disponível em: <<http://www.uces.br/etc/revistas/index.php/scientiacumindustria/article/view/9132>>. Acesso em: 3 dez. 2021.

GANDHI, R. **R-CNN, Fast R-CNN, Faster R-CNN, YOLO—Object Detection Algorithms**. 2018. Disponível em: <<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>> Acesso em: 3 dez. 2021.

WANG, Hongkai; SHANG, Shang; LONG, Ling; HU, Ruxue; WU, Yi; CHEN, Na; ZHANG, Shaoxiang; CONG, Fengyu; LIN, Sijie. **Biological image analysis using deep learning-based methods: Literature review**. *Digit Med [serial online]*, [S. l.], p. 157-165, 28 dez. 2018. Disponível em: <https://www.digitmedicine.com/article.asp?issn=2226-8561;year=2018;volume=4;issue=4;spage=157;epage=165;aui=Wang#top>. Acesso em: 3 dez. 2021.

CHEN, Kai; WANG, Jiaqi; PANG, Jiangmiao; CAO, Yuhang; XIONG, Yu; LI, Xiaoxiao; SUN, Shuyang; FENG, Wansen; LIU, Ziwei; XU, Jiarui; ZHANG, Zheng; CHENG, Dazhi; ZHU, Chenchen; CHENG, Tianheng; ZHAO, Qijie; LI, Buyu; LU, Xin; ZHU, Rui; WU, Yue; DAI, Jifeng; WANG, Jingdong; SHI, Jianping; OUYANG, Wanli; LOY, Chen Change; LIN, Dahua. **MMDetection: Open MMLab Detection Toolbox; Benchmark**. ArXiv preprint arXiv:1906.07155, 2019. Disponível em: <<https://github.com/open-mmlab/mmdetection>>. Acesso em: 5 dez. 2021.

YANIK, Hüseyin; KALOĞLU, A. Hilmi; DEĞIRMENC, Evren. **Detection of Escherichia Coli Bacteria in Water Using Deep Learning: A Faster R-CNN**. *Gazeta Técnica*, [S. l.], v. 14. 2020. DOI <https://doi.org/10.31803/tg-20200524225359>. Disponível em: <<https://hrcak.srce.hr/243661>>. Acesso em: 6 dez. 2021.

PYTORCH: CROSSENTROPYLOSS. 2019. Disponível em:
<<https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>>. Acesso em: 6 dez. 2021.

GIRSHICK, Ross. **Fast R-CNN.** 27 set. 2015. Disponível em:
<https://arxiv.org/pdf/1504.08083.pdf>. Acesso em: 6 dez. 2021.