UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Segmentação de cristais de clínquer em imagens microscópicas via redes neurais convolucionais

Renan Vinicius Rodrigues

Dissertação de Mestrado do Programa Interinstitucional de Pós-Graduação em Estatística (PIPGEs)

ICMC USP SÃO CARLOS

Renan Vinicius Rodrigues

Segmentação de cristais de clínquer em imagens microscópicas via redes neurais convolucionais

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP e ao Departamento de Estatística – DEs-UFSCar, como parte dos requisitos para obtenção do título de Mestre em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística. *VERSÃO REVISADA*

Área de Concentração: Estatística

Orientadora: Profa. Dra. Daiane Aparecida Zuanetti
Coorientador: Profa. Dra. Rosineide Fernando da Paz

USP – São Carlos
Dezembro de 2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R696s

Rodrigues, Renan Vinicius
Segmentação de cristais de clínquer em imagens microscópicas via redes neurais convolucionais / Renan Vinicius Rodrigues; orientadora Daiane Aparecida Zuanetti; coorientadora Rosineide Fernando da Paz. -- São Carlos, 2024.
90 p.

Dissertação (Mestrado - Programa Interinstitucional de Pós-graduação em Estatística) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Aprendizado por transferência. 2. Automação de processo. 3. Desempenho preditivo, . 4. Filtros de pré-processamento.. I. Zuanetti, Daiane Aparecida, orient. II. Paz, Rosineide Fernando da, coorient. III. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Renan Vinicius Rodrigues

Segmentation of clinker crystals in microscopic images by convolutional neural networks

Dissertation submitted to the Institute of Mathematics and Computer Science – ICMC-USP and to the Department of Statistics – DEs-UFSCar – in accordance with the requirements of the Statistics Interagency Graduate Program, for the degree of Master in Statistics. *FINAL VERSION*

Concentration Area: Statistics

Advisor: Profa. Dra. Daiane Aparecida Zuanetti
Co-advisor: Profa. Dra. Rosineide Fernando da Paz

USP – São Carlos
December 2024

“Dedico esse Trabalho para minha mãe Diva De Jesus dos Santos Barbosa (in memorian) que será minha eterna "Bonititica".

AGRADECIMENTOS

Eu sempre esperei pelo momento que estaria escrevendo esses agradecimentos, porque é o momento que estou finalizando o ciclo do meu mestrado. A vida por outro lado, gosta de mostrar que nem sempre as coisas serão como imaginamos, escrevo esses agradecimentos sem aquela que sempre foi o motivo de toda minha gratidão, aquela que me adotou com apenas 15 meses de vida, aquela que durante 26 anos sempre me deu os melhores abraços e sempre cuidou de mim, aquela que eu tive a imensa sorte de chamar de mãe. Espero que onde quer que esteja você saiba que não finalizaria esse trabalho sem você e sempre serei grato.

Apesar disso, existem outras pessoas que também foram muito importantes nesse processo. Agradeço minha tia Zilda e meus tios Reinaldo e Sebastião, que acreditaram no meu sonho de estudar fora da cidade. Agradeço a minha irmã Maria que sempre me ajudou, a meus primos Ruãn, Leonardo e Lígia por estarem sempre presentes (nem que seja para nos irritarmos), e a todos da família pelos "domingos de bolinho".

Sou imensamente grato aos meus amigos, que estão sempre ao meu lado, me mostrando o quanto sou sortudo por tê-los na minha vida: Claudio, Malu, Edvaldo, Luri, Juan, Lais, Thais, Murilo, Ana, Nati, Edu e tantos outros que, mesmo nos momentos mais difíceis, conseguiram arrancar de mim um sorriso sincero.

Por fim, mas não menos importante, sou grato a minha orientadora, Profª Drª Daiane Aparecida Zuanetti. Você foi uma luz na minha vida, nunca vou me esquecer de tudo que você fez por mim nesses anos. Muito obrigado por tudo!

RESUMO

RODRIGUES, R. V. **Segmentação de cristais de clínquer em imagens microscópicas via redes neurais convolucionais**. 2024. 90 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

O processo de análise de qualidade do cimento Portland, atualmente, é feito através de profissionais capacitados que analisam os cristais presentes na microestrutura do clínquer (insumo produzido no processo de fabricação do cimento e que confere a este suas principais características). Dentre esses cristais o que mais interfere no produto final é a Alita (C_3S).

Devido a isso, criar um processo automático para segmentação e classificação da C_3S em imagens microscópicas do clínquer pode trazer economia e eficiência na fabricação do cimento. Este trabalho, portanto, busca, através de redes neurais convolucionais e de filtros de pré-processamento de imagens, realizar essa segmentação para que seja viável a automatização do processo aumentando a qualidade do produto. Uma descrição de redes neurais e suas extensões é realizada, assim como uma breve revisão dos filtros de pré-processamento de imagens mais comuns. Posteriormente, vários modelos de redes neurais são ajustados e comparados na análise de imagens de clínquer.

Palavras-chave: Aprendizado por transferência, automação de processo, desempenho preditivo, filtros de pré-processamento.

ABSTRACT

RODRIGUES, R. V. **Segmentation of clinker crystals in microscopic images by convolutional neural networks**. 2024. 90 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

The quality analysis process of Portland cement is currently carried out by trained professionals who analyze the crystals present in the microstructure of the clinker (an input produced in the cement manufacturing process and which gives it its main characteristics). Among these crystals, the one that most affects the final product is Alita (C_3S).

Because of this, building an automatic process for segmenting and classifying C_3S in microscopic images of clinker can bring savings and efficiency in cement manufacturing. This work, therefore, seeks, through convolutional neural networks and image pre-processing filters, to carry out this segmentation so that the automation of the process is viable, increasing the quality of the product. A description of neural networks and their extensions is provided, as well as a brief review of the most common image preprocessing filters. Subsequently, several neural network models are fitted and compared in the analysis of clinker images.

Keywords: Transfer learning, process automation, predictive performance, preprocessing filters.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

O cérebro humano, ao longo do tempo, foi aperfeiçoando diversas habilidades que puderam proporcionar uma vantagem intelectual dos seres humanos em relação às demais espécies. Seja por meio da fala, escrita ou expressões artísticas é indiscutível que os *homo sapiens* mantiveram um constante processo de modernização que permitiu a criação da sociedade atual.

Com esses avanços, surgiram novas tecnologias que conseguem armazenar grandes quantidades de informação e, para processá-las com a finalidade de extrair conhecimento ou detectar padrões, houve a necessidade da criação de ferramentas estatísticas e computacionais de análises cada vez mais robustas.

Com isso, houve um aumento no interesse sobre a área de reconhecimento de padrões, que busca por meio de modelos estatísticos extrair conhecimento sobre conjuntos de dados cada vez maiores, como por exemplo em imagens, para as quais é possível através da análise de seus pixels, entender ou detectar seus padrões. Esse campo de estudo é denominado classificação e segmentação de imagens (HE *et al.*, 2016; HE *et al.*, 2017a; RONNEBERGER; FISCHER; BROX, 2015; HE *et al.*, 2017a; CHEN *et al.*, 2017; KRIZHEVSKY; SUTSKEVER; HINTON, 2012; IZBICKI; SANTOS, 2020).

A classificação e segmentação de imagens podem ser úteis em diversas áreas, como por exemplo no auxílio do diagnóstico de doenças quando aplicada em tomografias computadorizadas (RESMINI *et al.*, 2012); em análises geológicas quando utilizada em imagens aéreas de algum terreno (AMARAL *et al.*, 2009) e até mesmo na astrofísica quando desejamos analisar fotos tiradas por satélites ou telescópios (PEREIRA, 2018).

Sendo assim, é necessário se atentar a oportunidade de melhorias de processos já existentes utilizando classificação e segmentação de imagens. Um exemplo de processo que poderia ser mais eficiente com a utilização dessa tecnologia é o de produção de cimento.

O cimento é um dos produtos mais utilizado na área da construção civil e sua qualidade pode afetar as características das alvenarias que o utilizam. Em seu processo de fabricação é necessário que seja utilizado um insumo que corresponde a 90% do produto final. Esse insumo é denominado clínquer, que é obtido por meio da queima de algumas matérias-prima em fornos com temperaturas controladas (LEA, 1998; FLATT; SCHERER; BULLARD, 2007; BERETKA; BROWN; TAYLOR, 1993; GARCÍA-VIDAL; HIDALGO; PUERTAS, 2007; DÉJA; MROZ; KURDOWSKI, 2002; HILLS; ROBERTS, 1979). O controle do aquecimento destes fornos é importante para garantir a estrutura molecular do clínquer de modo que o mesmo proporcione qualidade às argamassas utilizadas nas construções de alvenarias. Para garantir que as temperaturas desses fornos gerem clínqueres capazes de fornecer cimentos com as características desejadas, é necessário que o produto seja avaliado periodicamente por profissionais altamente qualificados utilizando métodos de controle de qualidade. Entre os métodos utilizados está a microscopia, que consiste na obtenção de fotos microscópicas do clínquer já produzido, a fim de observar visualmente a presença de determinados elementos em sua estrutura. A forma, tamanho e distribuição desses elementos trazem informações sobre a temperatura do forno, além de outras informações como qualidade da matéria prima e estabilidade do processo produtivo como um todo.

Dentre os elementos investigados via microscopia está um cristal que pode ser representado de forma resumida por C_3S (também conhecido como alita), sendo este o cristal que mais influencia nas propriedades do produto final (o cimento Portland). Deste modo, realizar estudos com foco em identificar de forma automática o cristal de C_3S em imagens microscópicas de clínquer e classificá-los como sendo de boa ou ruim qualidade é de suma importância, tendo em vista que tal detecção (ou segmentação) e classificação irá proporcionar a sua análise de forma automatizada.

A automatização da análise do C_3S irá proporcionar maior agilidade na obtenção de informações sobre o processo produtivo, possibilitando intervenções mais ágeis para o controle do processo de produção como um todo. Além disso, uma vez que o controle da temperatura do forno pode evitar aquecimentos acima do necessário, esta automação também pode evitar o excesso de emissão de gás carbônico (CO_2) a ser lançado na atmosfera e reduzir o consumo energético.

Logo, com a segmentação e classificação da alita em imagens da estrutura molecular de amostras de clínquer retiradas dos fornos, será possível detectar anomalias, observar padrões e, através disso, determinar com o auxílio de ferramentas estatísticas se a temperatura utilizada deve permanecer ou ser alterada, economizando tempo e dinheiro na fabricação do material.

Neste trabalho, serão analisadas imagens realizadas por microscópio em clínqueres de uma fábrica do Ceará com a finalidade de trazer mais eficiência e qualidade na produção desse produto através de redes neurais convolucionais (LECUN *et al.*, 1998; JURASZEK, 2014; VARGAS; PAES; VASCONCELOS, 2016; ROSA, 2018; VOGADO *et al.*, 2019). As redes

neurais convolucionais (CNN) têm se destacado na análise e modelagem de imagens porque é uma metodologia que capta uma imagem de entrada, atribui importância (pesos que podem ser estimados) a vários aspectos e objetos da imagem e é capaz de diferenciar uns dos outros. A metodologia, portanto, detecta os padrões e segmenta os pixels da imagem e, consequentemente, extrai informações das figuras estudadas (SKANSI, 2018).

Apesar de estudos que analisam imagens via aprendizado de máquina serem frequentes na literatura (SHIBA *et al.*, 2005; SOUSA *et al.*, 2010; RESMINI *et al.*, 2012; SILVA; BREVE, 2013; NEGRI; SANT'ANNA; DUTRA, 2013; FERREIRA; MARCACINI, 2017; BRAGANÇA, 2018; SANTOS *et al.*, 2019; LEITE; MORAES; LOPES, 2020; IZBICKI; SANTOS, 2020) e de existirem inúmeros trabalhos sobre a microscopia do cimento (RIBEIRO, 2003; GOBBO, 2003; FREITAS, 2022; ZAKI *et al.*, 2023) e até mesmo de classificação de imagens microscópicas de cimento via redes neurais (PADILHA, 2022), não encontramos com facilidade estudos que se aprofundem na combinação e comparação de metodologias em tratamentos de imagens. Logo, é importante o estudo e comparação de novos procedimentos de classificação e/ou segmentação de imagens.

As alitas são classificadas de três formas: subdiomórficas, xenomórficas e idiomórficas. Assim, identificar a quantidade de cada classe nas imagens é de grande importância para entender a qualidade da argamassa.

Como já comentado anteriormente, as redes neurais convolucionais são, atualmente, uma das metodologias mais utilizadas em grande parte das análises de imagens por serem capazes de fazer segmentações e/ou detectar padrões. Através das camadas de convoluções, conseguem captar e descrever as informações correlacionadas entre pixels próximos e, de maneira eficiente, consideram várias funções não lineares entre essas características. Além disso, elas são escaláveis e capazes de processar grandes conjuntos de dados e podem ser integradas em outros algoritmos e técnicas de aprendizado de máquina.

Em suma, esse trabalho tem como finalidade, segmentar imagens de clínquer obtidas através de microscópio para localizar e classificar as classes dos cristais de alita (Subdiomórfica, Xenomórfica e Idiomórfica) que afetam diretamente na qualidade dos cimentos. Para isso, iremos utilizar redes neurais convolucionais e suas derivações, afim de encontrar o melhor modelo para o conjunto de dados selecionado. Juntamente a isso, iremos analisar o desempenho de filtros de pré-processamento das imagens, além do filtro bilateral (tradicionalmente muito utilizado) e tratamento de desbalanceamento das classes. Por fim, avaliaremos qual estrutura possui melhor desempenho, levando em consideração também o critério de parcimônia.

Esse trabalho está organizado como segue. O Capítulo 2 apresenta as principais motivações para a realização desse trabalho. No Capítulo 3, descrevemos as redes neurais e suas extensões, enquanto que o Capítulo 4 traz alguns detalhes técnicos dos filtros de pré-processamento de imagens, aprendizado por transferência e métricas de desempenho do modelo. No Capítulo 5, descrevemos os dados com imagens de clínquer a ser analisados, os modelos ajustados, seus

resultados e comparações. Por fim, o Capítulo 6 apresenta discussões finais e trabalhos futuros.

CAPÍTULO

2

CONTROLE DE QUALIDADE DO CIMENTO PORTLAND VIA ANÁLISE DE IMAGENS MICROSCÓPICAS

Neste capítulo, iremos abordar as principais fundamentações teóricas da fabricação do cimento, importantes para entendimento e contextualização do problema. Elas são: processo de produção do cimento, maneiras de mensurar sua qualidade e como automatizar esse processo.

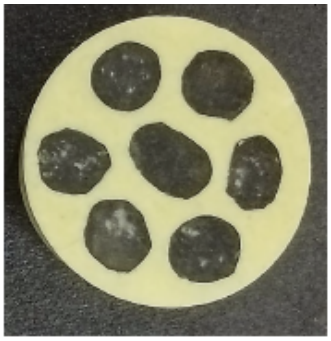
2.1 Produção e controle de qualidade do clínquer em uma planta de cimento

Um dos principais componentes do cimento Portland é o clínquer. Para sua fabricação, primeiramente, as empresas produtoras de cimento britam e moem a rocha calcária e a argila em proporções adequadas. O resultado dessa moagem é uma farinha, que é depositada em grandes fornos circulatórios que atingem altas temperaturas e conseguem modificar a estrutura do material aquecido criando o clínquer. Por fim, esse clínquer é resfriado e moído. Após a adição do gesso ou outros insumos dependendo do tipo de argamassa a ser fabricada, obtemos o cimento utilizado nas construções de alvenaria.

Um ponto importante da fabricação do clínquer é que a temperatura dos fornos, tempo de queima e resfriamento do material interfere diretamente na sua estrutura microscópica e consequentemente na qualidade do cimento obtido a partir desse clínquer.

A microestrutura do clínquer pode ser analisada utilizando imagem microscópica do produto, ou seja, via microscopia. Esse método, consiste em separar os grãos mais representativos do produto a ser analisado para serem lixados e polidos. Em seguida, aos espécimes resultantes, é adicionada uma resina com altos índices de reflação, que facilita a visualização da estrutura

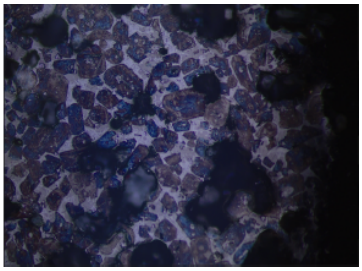
microscópica da amostra obtida. A Figura 1 contêm uma dessas amostras após a realização desse procedimento.



Fonte: Lima (2022).

Figura 1 – Amostra de clínquer pronta para vizualização no microscópio.

Após essa preparação inicial, as imagens são obtidas usando um microscópio digital, sendo em seguida direcionadas a um computador (Figura 2), a fim de que sejam analisados os componentes do produto. Os componentes da imagens são cristais que devem aparecer em distribuição, tamanhos e formatos adequados. Dentre os cristais a serem analisados estão: alita (C_3S), belita (C_2S), aluminato de cálcio (C_3A) e ferrita (C_4AF), sendo a alita o mais influente na determinação da qualidade do clínquer e muito impactado pela condição geral de operação do forno que aquece a farinha para dar origem ao clínquer.



Fonte: Lima (2022).

Figura 2 – Imagem de uma amostra de clínquer.

A Figura 3 mostra um cristal de C_3S identificado visualmente e demarcado manualmente. Para um produto ideal, esse cristal deve aparecer em formato aproximadamente hexagonal, e uma informação importante do processo produtivo é a sua dimensão. Em geral, essas informações são obtidas de forma manual, e já existem pesquisas iniciais para a automação da obtenção das informações necessárias a partir dos cristais que compõem o clínquer (RIBEIRO, 2003; FREITAS, 2022; GOBBO, 2003; LIMA, 2022; PADILHA, 2022).

Outros cristais também são analisados para que seja avaliada a qualidade do produto e estabilidade do processo produtivo. Sendo eles: belita, magnésio, cal livre e porosidades. No entanto, devido a importância e desafio da avaliação da alita, este trabalho tem por objetivo fornecer estratégias para tornar possível a automação da avaliação do cristal de alita, tendo

em vista que essa automação depende de estratégia para analisar as imagens microscópicas do clínquer.

Para que seja possível a avaliação do cristal de forma automática, faz-se necessária a realização da segmentação das imagens. Nesse processo, cada cristal deve ser representado de forma individual em uma imagem, assim o mesmo pode ser mensurado e classificado segundo sua forma. Esse processo também possibilita a avaliação da distribuição dos cristais na imagem original, que contém vários cristais e não apenas um.

Fonte: Lima (2022).

Figura 3 – Exemplo de imagem de um clínquer com um cristal pré-especificado para o treinamento do modelo.

Em Padilha (2022) também foi utilizado a metodologia de redes neurais convolucionais para segmentar imagens de alita em microscopia de clínquer. Anteriormente à aplicação dessa técnica, foi constatado que a utilização das metodologias de segmentação *k-means* ou KNN (*k-nearest neighbors*) não era adequada para o conjunto de dados, pois gerava resultados insatisfatórios, principalmente por agrupar os cristais de alita e poros (também chamados de porosidades, são pontos vazios na microscopia que apresentam cor mais escura que a alita e não apresentam alguma forma padrão), impossibilitando a segmentação dos mesmos.

Para seu trabalho, Padilha (2022) utilizou um filtro padrão no pré-processamento do banco de dados para minimização e suavização de ruídos nas imagens, comumente encontrado na literatura como filtro bilateral. Esse filtro mantém as cores, bordas e demais propriedades essenciais para estudar os componentes minerais representados após sua aplicação. Os hiperparâmetros da metodologia são estimados via algoritmo genético (MIRJALILI *et al.*, 2020). Além disso, foi necessário utilizar uma extensão da rede neural convolucional clássica denominada *Mask R-CNN*, que é frequentemente utilizada em problemas de segmentação de imagens.

Apesar dessa estrutura de CNN gerar resultados satisfatórios para os objetivos propostos, ainda há pontos de melhorias que se estudados e implementados podem gerar resultados com qualidade parecida ao que, atualmente, um analista de microscopia possui. Isso porque a metodologia utilizada não conseguiu detectar todos os cristais de C_3S. Assim, é necessário estudar o impacto que outros filtros, além do bilateral, e outras formas de estimação de seus hiperparâmetros, além do algoritmo genético, podem gerar nos resultados finais já que esse pré-processamento pode melhorar a segmentação das imagens (PADILHA, 2022).

CAPÍTULO

3

REDES NEURAIS PARA SEGMENTAÇÃO E CLASSIFICAÇÃO DE IMAGENS

Este capítulo, demonstra como as imagens podem ser representadas na forma matemática para serem utilizadas em modelos estatísticos e quais são os principais métodos de análise de imagem com foco na metodologia das redes neurais e suas extensões.

3.1 Representação matemática de imagens

Uma imagem pode ser descrita em formato de uma ou mais matrizes que armazenam a informação de cada um de seus pixels. Logo, podemos representar uma figura em preto e branco através de uma matriz em que cada elemento terá valor 0 se o pixel correspondente aquele elemento for preto e 1 se ele for branco. Assim, a matriz

$$I_m = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},$$

forma a imagem da Figura 4.

Figura 4 – Imagem formada pela matriz I_m.

Podemos também descrever uma imagem em tons de cinza através de uma matriz com todos os elementos com valores entre 0 e 1, sendo que quanto mais próximo de 0, mais escura

será a cor do pixel. A matriz

$$T = \begin{bmatrix} 0.00 & 0.35 & 0.75 & 1.00 \end{bmatrix},$$

por exemplo, ilustra a representação matemática da Figura 5 em tons de cinza.

Figura 5 – Imagem formada pela matriz T.

Por fim, podemos adicionar cores à imagem considerando cada elemento da matriz da imagem como uma cor distinta. Uma das formas mais utilizadas para representar uma imagem colorida é através de três matrizes separadas, cujos elementos de cada matriz representam a intensidade de azul, verde e vermelho em cada pixel, que são as cores que compõem as imagens. Juntando os valores dessas matrizes, formamos a verdadeira cor de cada pixel.

Nesse trabalho utilizamos essa forma de representação dos pixels e buscamos através das redes neurais, entender se existe um padrão entre essas matrizes e segmentar os elementos nas imagens microscópicas de clínquer. Esse conjunto de matrizes são chamadas de RBG *Channels*, sendo *Chanels* o nome dado a cada matriz e RBG a sigla para as cores que elas representam: vermelho (*red*), azul (*blue*) e verde (*green*).

A segmentação de imagens é um tratamento que visa particionar uma imagem I em um subconjunto composto de m regiões R_t (para $t = 1,2,3,...,m$) tais que (ANDRADE, 1998):

- toda região possui algum pixel ($R_t \neq \varnothing \ \forall 1 \leq t \leq m$);
- diferentes regiões não possuem pixels comuns ($R_t \cap R_q = \varnothing$, para todo $t \neq q$); e
- a união de todas as regiões R_t forma a imagem original ($I = \bigcup_{t=1}^{m} R_t$),

sendo que m e t pertencem ao conjunto $\mathbf{N}$ formado pelos números naturais e $0 < t \leq m$. Uma região é definida como um conjunto de pontos (pixels) que possuem alguma propriedade em comum, seja a cor, textura, intensidade, entre outras propriedades que diferenciam a mesma das regiões vizinhas. É necessário definir em estudos de segmentação de imagem quais são as regiões de interesse.

Nesse trabalho, a segmentação ideal deve delimitar os principais componentes químicos da microscopia do clínquer (alita e os poros que são as regiões de interesse) assim como na Figura 6.

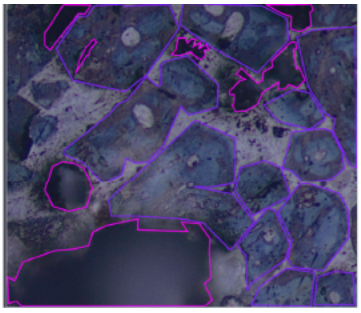


Figura 6 – Segmentação correta da microscopia do clínquer.

3.2 Métodos para análise de imagens

Devido a sua ampla gama de possibilidades de utilização e resultados obtidos, métodos de classificação e segmentação de imagens têm sido cada vez mais utilizados. Contudo, sempre é necessário analisar e entender qual modelo conseguirá trazer melhores resultados aos dados estudados (IZBICKI; SANTOS, 2020).

Logo, antes de utilizar alguma modelagem envolvendo imagens é importante entender quais são os principais modelos de classificação e segmentação, além de suas principais características e suposições afim de detectar aquele que melhor se ajusta ao seu objetivo e ao conjunto de dados utilizado.

Uma vez que cada pixel de uma imagem é representado por um número ou um vetor de três componentes numéricas, modelos computacionais e estatísticos se tornam ferramentas úteis para analisá-los. A estimação de parâmetros (que podem ser estimados utilizando uma base de treinamento) e hiperparâmetros (que devem ser estudados usando estratégias adicionais a estimação pela base de treinamento) desses modelos são constantemente referidos na literatura como aprendizado de maquina (BISHOP, 2006; HASTIE; TIBSHIRANI; FRIEDMAN, 2009a; GOODFELLOW; BENGIO; COURVILLE, 2016; VAPNIK, 1999; SCHÖLKOPF; SMOLA, 2001; BREIMAN, 2001).

Os métodos de aprendizado são geralmente divididos em aprendizados supervisionados (MURPHY, 2012; MITCHELL, 1980), não supervisionados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009b; COATES; NG; LEE, 2011) e semi-supervisionados (CHAPELLE; SCHÖLKOPF; ZIEN, 2009; ZHU, 2005).

O aprendizado supervisionado necessita que a base de treinamento do modelo contenha tanto informações de variáveis preditoras quanto informações da variável que se buscar predizer para os indivíduos da amostra. Como exemplo, em modelos que determinam o número da placa de um carro através de uma fotografia de radar, precisamos de uma base de treinamento com diversas imagens de números já rotulados com sua correta classificação. Logo, ao estimarmos ou aprendermos o modelo, identificamos quais são os padrões (comportamentos) dos pixels para cada um dos possíveis verdadeiros números, através das informações do conjunto de treinamento.

O aprendizado não supervisionado, como o nome já sugere, não necessita de informações de uma variável a ser predita para estimar os parâmetros do modelo. Um exemplo desse processo é quando temos um conjunto de dados sobre a compra de clientes de um supermercado e queremos definir se existem alguns grupos com mesmo padrão de compra. Esse método é muito utilizado para que empresas, baseadas em apenas algumas compras, consigam prever o que um determinado cliente deseja comprar levando em consideração outros clientes com mesmo perfil e assim realizar ofertas mais precisas.

Por fim o aprendizado semi-supervisionado utiliza informações rotuladas e não rotuladas, tendo como principal vantagem a possibilidade de aumentar um conjunto de dados supervisionado com informações não supervisionadas.

Nesse trabalho, usaremos métodos de aprendizado supervisionado para segmentação e análise das imagens de interesse.

3.3 Redes neurais

Nessa seção, iremos abordar os principais conceitos de uma rede neural com a finalidade de entender como essa ferramenta pode ser capaz de extrair informações e padrões de um conjunto de imagens.

Redes neurais são modelos ou funções que combinam de maneira não linear informações de variáveis preditoras na busca de identificar padrões e associações e usá-los para agrupar ou classificar observações e indivíduos, possivelmente com origem no século XX. Esses métodos têm ganhado cada vez mais atenção, principalmente na área de IA (inteligência artificial).

As redes neurais artificiais (RNA) tratam-se de modelos que buscam estimar uma função $g(x)$, em que x representa o conjunto de variáveis preditoras disponíveis no estudo, que seja a mais próxima possível da real função de regressão, classificação ou segmentação dos dados observados. Uma rede pode ser representada de forma gráfica, através de grafos.

Um grafo permite representar a relação entre dois ou mais objetos. Podemos descrevê-lo como uma função **G(V,A)**, em que **V** é o conjunto de objetos (vértices) e **A** é o conjunto de arestas que representam as relações entre esses objetos. As arestas podem, dependendo do objetivo do estudo, terem uma direção e até mesmo ligar um objeto a ele próprio (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2018).

Um exemplo de grafo pode ser visualizado na Figura 7 cujo $\mathbf{V} = \{a, b, c\}$ e temos duas arestas direcionadas pertencentes a **A** ligando a ao objeto b e ao c, respectivamente.

Assim como nas RNA, os grafos podem ser úteis para representar outros modelos estatísticos. O grafo na Figura 8 ilustra, por exemplo, a regressão simples com vértices X_1 e Y e uma aresta ligando esses dois objetos que representa a função $\beta_0 + \beta_1 X_1$ que resulta em Y. Repare que a aresta pode representar diversas funções e assim construir grafos de diversos modelos

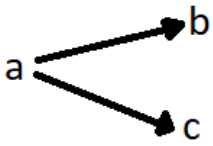
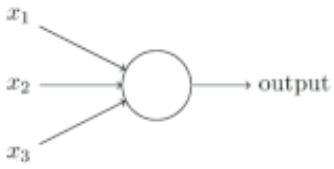


Figura 7 – Exemplo de um grafo.

estatísticos como de uma regressão logística se Y for uma variável binária ou até mesmo uma rede neural simples. Para mais detalhes sobre a teoria dos grafos consulte Feofiloff, Kohayakawa e Wakabayashi (2018).

Figura 8 – Ilustração da regressão simples por um grafo.

Um neurônio de uma rede neural, em particular, pode ser representado pelo grafo da Figura 9, em que as variáveis preditoras x_1, x_2 e x_3 são linearmente combinadas, o resultado aplicado em uma função de ativação, melhor descrita posteriormente, que gera um valor como *output*.



Fonte: Izbicki e Santos (2020).

Figura 9 – Ilustração de uma rede neural.

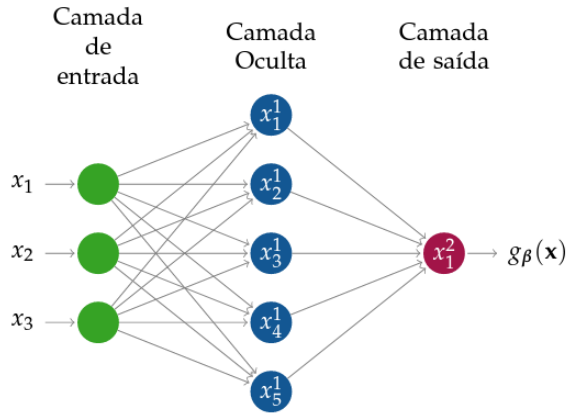
A Figura 10, por sua vez, representa uma rede neural mais complexa com uma camada oculta e vários nós, em que os nós (neurônios) da esquerda são as entradas da rede neural, ou seja, são as covariáveis ou variáveis preditoras. Os nós (neurônios) na camada intermediária são considerados a camada oculta do modelo. Cada flecha ou aresta representa um parâmetro β que combinado com os valores da covariável na camada anterior dá origem a novas covariáveis na camada seguinte. Os nós representam portanto uma transformação, através da combinação linear e aplicação da função de ativação escolhida para cada um, dos valores dos nós da camada anterior. Na extrema direita temos, finalmente, o *output*, ou seja, o resultado da rede. Esse exemplo em questão mostra uma das formas mais simples que uma rede neural pode assumir, denominada *Perceptron*.

Em um modelo de classificação binária e usando a rede da Figura 9 (sem camada oculta), podemos considerar, por exemplo que a variável resposta predita será dada por:

- 1, se $\sum_{i=1}^{d} \beta_i x_i \geq -b$; e

- 0, caso contrário,

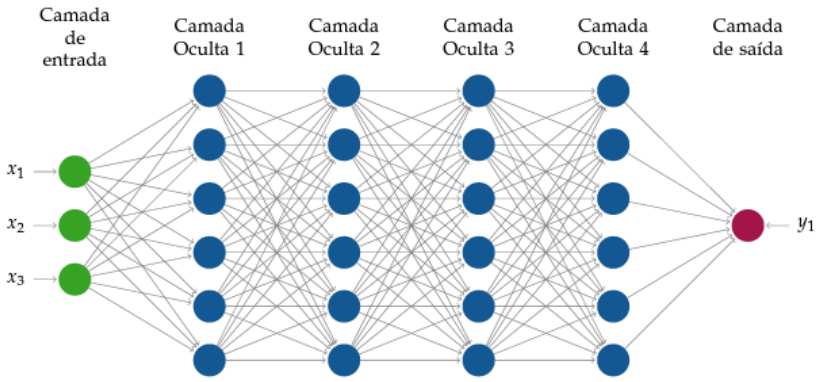
em que d é a quantidade de covariáveis de entrada na rede e β_i e b são parâmetros que devemos estimar com a base de treinamento.



Fonte: Izbicki e Santos (2020).

Figura 10 – Exemplo de uma rede neural com uma camada oculta.

Uma das grandes vantagens de uma rede neural é que a mesma pode ser extremamente flexível. Essa propriedade é dada devido a mesma não possuir uma estrutura única, sendo facilmente adaptável a diferentes conjuntos de dados e podendo ser construída com diferentes números de camadas e neurônios (nós). A Figura 10, como já dito anteriormente, representa uma rede neural com apenas uma camada oculta, porém com um número de nós maior do que o exemplo apresentado na Figura 9.



Fonte: Izbicki e Santos (2020).

Figura 11 – Exemplo de uma rede neural com quatro camadas ocultas.

A Figura 11 se trata de uma rede mais complexa que possui um número maior de camadas ocultas. Desse modo, podemos criar redes tão complexas quanto necessárias para ajustar os dados. Contudo, sempre é necessário ter uma atenção em relação ao número de elementos na estrutura para evitar *overfitting*.

Outro ponto importante na estrutura de uma rede neural é a função utilizada em cada nó a ser aplicada sobre a combinação linear das covariáveis da camada anterior e seus pesos. Esta é denominada como função de ativação. Existem diversas funções de ativação e elas podem ser representadas como

$$f\left(\beta_{0,j}^{m}+\sum_{i=1}^{d_{m-1}}\beta_{i,j}^{m}x_{i}^{m-1}\right),$$

sendo

- $\beta_{i,j}^{m}$ o $i-$ésimo parâmetro do j-ésimo neurônio da m-ésima camada, para $i=1,2,\ldots,d_{m-1}$; $j=1,2,3,\ldots,d_{m}$; $m=1,2,3,\ldots$;
- d_i é o número de neurônios na camada i; e
- f a função de ativação escolhida para o determinado nó.

A função de ativação escolhida depende da finalidade e do tipo de dados estudados. Alguns dos exemplos mais utilizados na literatura são (IZBICKI; SANTOS, 2020):

- rectified linear unit (ReLU): é uma das funções de ativação mais usadas em CNN por ser computacionalmente eficiente e manter os gradientes das estimações significativos para valores positivos. Isso, reduz o desvanecimento do gradiente, que é quando os gradientes diminuem exponencialmente conforme os parâmetros são estimados (GOODFELLOW; BENGIO; COURVILLE, 2016), que é uma das principais dificuldades de treinar uma rede neural profunda. A formula da ReLu é dada por$f(g)=max(0,g)$, em que g é o valor que um neurônio recebe como somatório ponderado das camadas anteriores;
- logística ou sigmóide: é útil para modelar probabilidades, pois retorna um valor entre 0 e 1, contudo pode sofrer com o desvanecimento do gradiente em redes profundas (GOODFELLOW; BENGIO; COURVILLE, 2016). Sua fórmula é dada por $f(g)=\frac{1}{1+e^{-g}}$, em que g é o valor que um neurônio recebe como somatório ponderado das camadas anteriores;
- tangente hiperbólica: apesar de ainda poder sofrer com o desvanecimento do gradiente como a sigmóide, ela pode ter uma convergência mais rápida por ser simétrica em torno do ponto (0,0). Outro ponto diferente em relação a sigmóide é que suas saídas possuem valores entre -1 e 1 (GOODFELLOW; BENGIO; COURVILLE, 2016). Sua função é dada por$f(g)=\frac{e^{g}-e^{-g}}{e^{g}+e^{-g}}$, em que g é o valor que um neurônio recebe como somatório ponderado das camadas anteriores;
- identidade: dada por $f(g)=g$, em que g é o valor que um neurônio recebe como somatório ponderado das camadas anteriores.

3.3.1 Estimação ou treinamento

Se uma rede possui H camadas ocultas com d_h neurônios cada ($h = 1,2,3,\ldots,H$), podemos escrever o estimador da função de regressão dessa RNA como

$$g_\beta(x) = x^{H+1} = f\left(\beta_{0,1}^{H+1} + \sum_{i=1}^{d_H} \beta_{i,1}^{H+1} x_i^H\right),$$

sendo $\beta_{i,j}^l$ o peso dado para a conexão da entrada i da camada l com a saída j da camada $l+1$. Assim, para $l = 1,2,3,\ldots,H$ e $j = 1,2,3,\ldots,d_{l+1}$, temos,

$$x_j^{l+1} = f\left(\beta_{0,j}^{l+1} + \sum_{i=1}^{d_l} \beta_{i,j}^{l+1} x_i^l\right).$$

Desse modo, podemos estimar os parâmetros $\beta_{i,j}^l$ como sendo os valores que minimizam o erro quadrático médio (EQM) dessa função de regressão g_β, aproximado por

$$EQM(g_\beta) = \frac{1}{n} \sum_{k=1}^{n} (g_\beta(x_k) - y_k)^2.$$

Outras funções de perda como, por exemplo, a entropia cruzada também pode ser utilizada. Como não existe solução analítica para encontrar o vetor de parâmetros β que minimiza o EQM, é necessária a utilização de métodos numéricos. Uma das soluções mais utilizadas (IZBICKI; SANTOS, 2020) é o *backpropagation* que realiza um método de gradiente descendente primeiramente atualizando os valores da última camada, depois da camada anterior a essa e assim por diante, ou seja, para $l = H+1, H, \ldots, 1$. A atualização a ser realizada é dada por

$$\beta_{i,j}^{l(t+1)} \leftarrow \beta_{i,j}^{l(t)} - \eta \frac{\partial EQM(g_\beta)}{\partial \beta_{i,j}^l},$$

para $j = 1,2,\ldots,d_l$ e $i = 0,1,2,\ldots,d_{l-1}$ e sendo η um índice que varia entre zero e um ($\eta \in [0,1]$) e determina o quanto do valor atual de cada parâmetro, será alterado na próxima iteração. Esse valor é conhecido como taxa de aprendizagem (GUIMARÃES *et al.*, 2008). Para mais detalhes sobre esse método veja Rojas (2013).

Um otimizador para o cálculo dos parâmetros que minimizam a função de perda é o gradiente descendente estocástico que segue o mesmo conceito do *backpropagation*, porém em cada iteração não é utilizado o conjunto de treinamento completo e sim amostras do mesmo que diminui o custo computacional.

3.4 Redes neurais convolucionais

Antes de adentrarmos na definição de redes neurais convolucionais, que é o método que será aplicado nesse trabalho, é importante entender o conceito de uma convolução. Convolução

é uma operação matemática que através de duas funções f e g produz uma terceira função (h), sendo esta uma f modificada (trataremos o operador de convolução pelo simbolo $*$) (FERREIRA, 2017). Logo,

- se o domínio de f e g pertence ao conjunto dos inteiros ($x \in \mathrm{Z}$), temos:

  $f(x) * g(x) = h(x) = \sum_{-\infty}^{\infty} f[\tau]g[x-\tau]$;

- se o domínio de f e g é contínuo, ou seja, pertence ao conjunto dos números reais ($x \in \mathrm{R}$), temos:

  $f(x) * g(x) = h(x) = \int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau$.

A convolução pode ser útil em tratamento de imagens, pois com elas podemos detectar bordas (fronteiras entre diferentes sub-imagens), destacar informações horizontais ou verticais, realizar suavizações, entre outras funções, isso porque essa operação pode ser interpretada, quando aplicada em uma imagem, como se estivéssemos aplicando um filtro na mesma (FERREIRA, 2017).

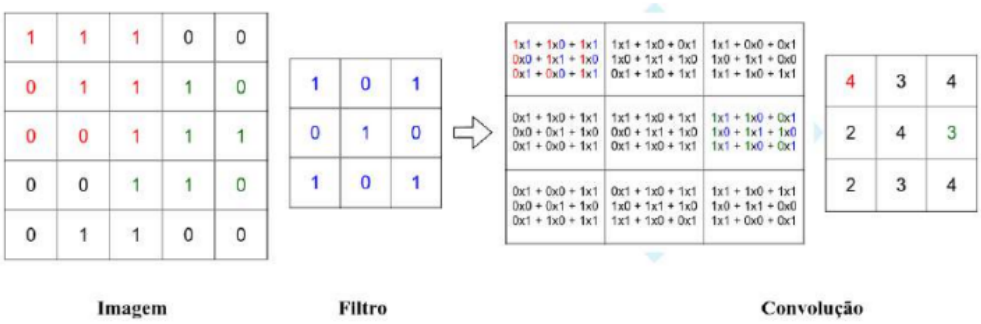
A Figura 12 mostra um exemplo de convolução em uma imagem de tamanho 5×5 pixels. Nesse caso, a operação é feita através de um filtro de dimensão 3×3, que percorre toda a extensão da imagem realizando uma multiplicação dos valores entre as duas matrizes e gerando assim 9 sub-matrizes que representam o produto de cada parte da imagem pelo filtro. Por fim, essas novas tabelas são resumidas pela soma de seus elementos, produzindo assim, uma matriz 3×3. Essa nova matriz é denominada mapa de características.

Os mapas de características, apesar de possuírem dimensões inferiores a imagem original, capturam as características detectadas por filtros convolucionais em cada camada de uma CNN. Sendo assim, essas tabelas intermediárias desempenham um papel importante na extração e hierarquização de características das imagens de entrada, pois permite que a rede aprenda e represente os padrões complexos dos dados das redes, possibilitando a classificação, detecção de objetos e segmentação do banco de dados. Mais detalhes sobre os conceitos de mapa de características podem ser vistos em Goodfellow, Bengio e Courville (2016).

Outro ponto relevante na Figura 12 é que a redução do tamanho da matriz original ocorreu, porque descartamos os pixels das bordas para os quais não era possível aplicar o filtro de dimensão 3×3. Uma outra maneira de aplicar um filtro de convolução sem reduzir o tamanho da matriz original é preencher as bordas com zeros, de maneira que seja possível aplicar o filtro com a dimensão escolhida e, assim, consideramos na conta apenas os pontos que estão na imagem. Observe que a aplicação dos filtros de convolução nas informações de cada pixel é uma maneira de combiná-las com as informações dos pixels vizinhos, pois eles são associados.

Um ponto importante é que o filtro não necessariamente precisa multiplicar os elementos da imagem e nem é preciso que as matrizes resultantes após a ação do filtro sejam resumidas

pela soma de seus elementos, como foi feito no exemplo da Figura 12. Podemos substituir essas operações por outras como a escolha do elemento de maior valor, cálculo da média, escolha do elemento mínimo, cálculo da média ponderada, entre outras, sendo que a escolha de como será feita a convolução dependerá do objetivo do estudo.



Fonte: Ferreira (2017).

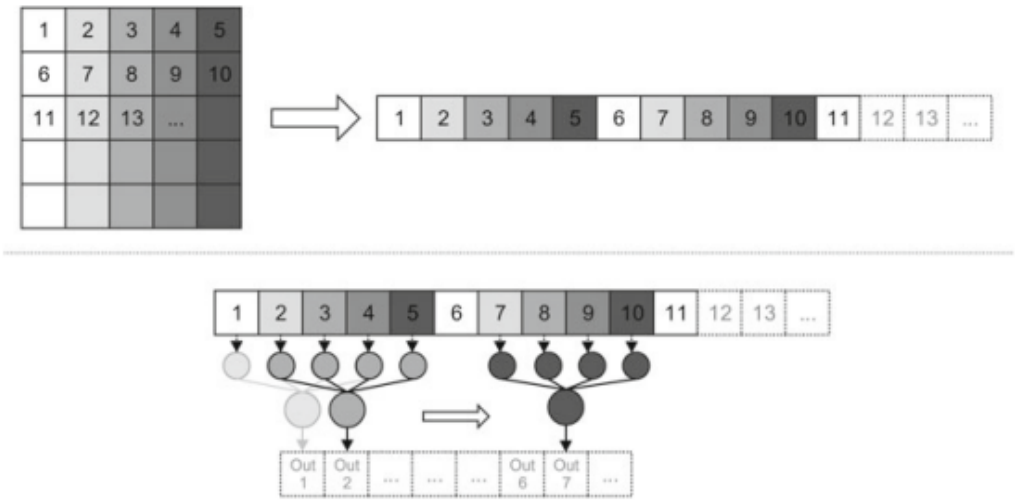
Figura 12 – Exemplo de uma operação de convolução em uma imagem.

As redes neurais convolucionais (CNN) foram propostas por LeCun *et al.* (1998) baseando-se em estudos realizados anteriormente sobre o córtex visual de animais que identificaram uma ligação entre partes do campo visual e neurônios que processam informação de forma individual. Esse conceito é denominado campo receptivo (SKANSI, 2018).

Essa descoberta, juntamente com a capacidade de transformarmos imagens em vetores possibilitou a criação das redes neurais convolucionais, que são RNA com uma ou mais camadas convolucionais, geralmente alternadas entre aplicações de uma função de ativação nos dados convolucionados e uma camada de *pooling*. A camada de *pooling* serve para reduzir a dimensão da imagem e a técnica mais tradicional é a *max-pooling*. Essa técnica resume as informações de um sub-bloco de elementos da imagem em um único valor através do valor máximo observado nesse sub-bloco.

Após aplicar uma ou algumas camadas de convolução, função de ativação e *pooling* nas imagens originais (camadas para obter as variáveis preditoras com base nas imagens originais), uma camada densa ou completamente conectada é conduzida. É nela que é iniciado o processo para classificar as informações extraídas pelas camadas anteriores. Para isso, o bloco de informações (matrizes) é transformado em uma única linha (vetorização) que contém todas as informações extraídas. Esse processo é exemplificado na Figura 13. Nela podemos observar, primeiramente, o achatamento de uma determinada imagem transformando a mesma em um vetor e posteriormente retorna como *output* um vetor de dimensão menor que o anterior sendo este formado pelo agrupamento com uma determinada função de 5 elementos consecutivos do vetor anterior.

Portanto, por trás das CCN realizamos camadas com convoluções de imagens, conforme vimos na Figura 12, aplicação de função de ativação e *pooling* até resultar em um imagem



Fonte: Skansi (2018).

Figura 13 – Exemplo de um achatamento de imagem.

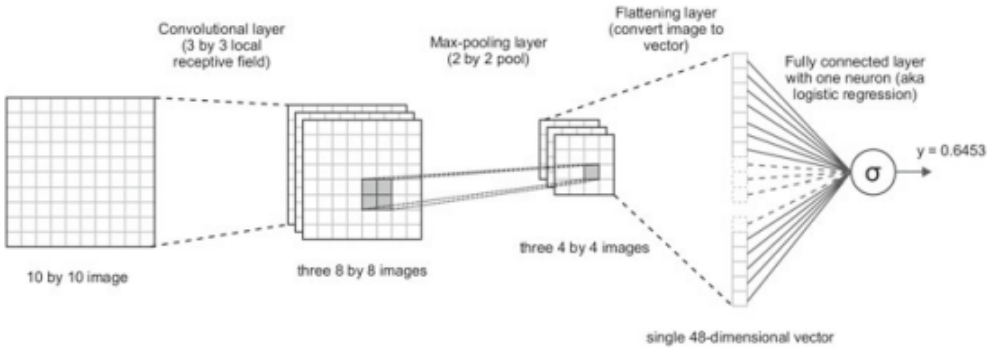
pequena o suficiente para torná-la em vetor, com ruídos suavizados e algumas características de bordas, formas, texturas, curvas, linhas horizontais e verticais, etc destacadas.

A Figura 14 exemplifica o funcionamento de uma CCN. Nela vemos que partimos de uma imagem de dimensão 10×10 pixels e aplicamos um filtro de dimensão 3×3. Observe que ao contrário do exemplo da Figura 12, nesse caso não obtemos uma única matriz menor que a imagem de entrada após a primeira convolução e sim três matrizes. Isso ocorre, pois muitas vezes é necessário separarmos uma imagem em três ou mais "$sub - imagens$" para não perdermos informação sobre a mesma, aplicando diferentes filtros de convolução que ressaltam diferentes aspectos da imagem. Nesse caso, cada imagem gerada funciona como uma RBG *Chanels*, mas não mais em termos de intensidade de cores. Assim temos uma melhora no processamento da rede, pois não perderemos a informação das cores de cada pixel.

Seguindo o processo, obtemos imagens cada vez menores (via *Max-pooling*), até chegar na dimensão 4×4, realizar o achatamento e aplicar uma função de ativação adequada (aqui a função logística) para classificar e/ou segmentar cada pixel.

O número de camadas convolucionais e de *pooling* realizadas antes da camada densa, assim como as funções de ativação usadas geralmente são fixados pelos autores considerando que selecioná-los entre vários possíveis é muito custoso computacionalmente.

Considerando a quantidade enorme de parâmetros a serem estimados e também para evitar *overfitting*, passos de *dropout* são necessários. O *dropout* é uma técnica de regularização simples de implementar, mas que possui boas propriedades de regularização. A técnica consiste em selecionar uma amostra dos parâmetros em uma das camadas e apagá-los, forçando que os valores sejam fixados em zero. Ele geralmente é aplicado após finalizada as camadas de convolução e *pooling*. Mais detalhes sobre redes neurais convolucionais podem ser encontrados em Skansi (2018).



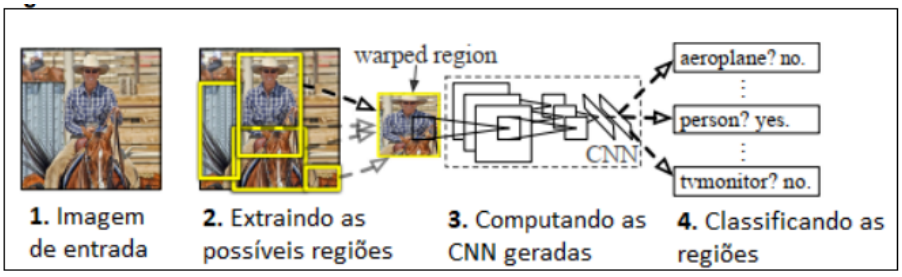
Fonte: Skansi (2018).

Figura 14 – Exemplo do funcionamento de uma rede neural convolucional.

3.5 Extensões das redes neurais convolucionais

As redes neurais convolucionais, no entanto, possuem extensões que possibilitam uma segmentação mais precisa da região de interesse de uma figura.

Girshick *et al.* (2014) sugeriram um método que permite que a convolução seja realizada em regiões que presume-se haver um objeto de interesse ao invés de realizá-la sempre em todos os pontos da imagem sem nenhum método de seleção dessas regiões. As extensões da CNN que possuem essa forma de pré triagem das áreas de interesse são encontradas na literatura com a nomenclatura R-CNN (do inglês, *Region Based Convolutional Neural Networks*).



Fonte: Girshick *et al.* (2014) adaptada.

Figura 15 – Ilustração do funcionamento de uma R-CNN.

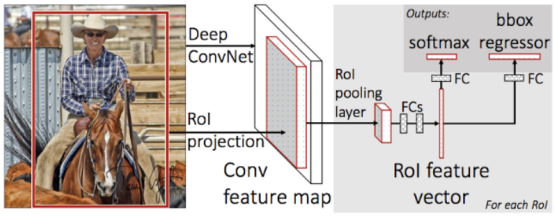
A Figura 15 exemplifica o funcionamento de uma R-CNN. Os principais passos dessa rede se consistem em:

- uma imagem de entrada é inserida;
- o sistema extrai aproximadamente 2000 possíveis regiões de interesse (COSTA, 2019);
- posteriormente, cada região selecionada passa por camadas de convolução utilizando as redes neurais convolucionais; e

- por fim, utilizando um classificador linear adequado (como, por exemplo, uma regressão logística) cada região é classificada de acordo com o objetivo da rede. Na figura em questão, a região selecionada foi classificada como possuindo uma pessoa.

Contudo, a R-CNN possui um tempo de processamento elevado (COSTA, 2019), e para amenizar esse custo computacional, Girshick (2015) propõe uma nova extensão, denominada *Fast R-CNN* (do inglês, *Fast Region Convolutional Neural Network*) (REN *et al.*, 2015).

A proposta da *Fast R-CNN* é parecida com o R-CNN, mas nela é gerado um mapa convolucional de recurso contendo as principais informações da imagem inicial (GIRSHICK, 2015). A Figura 16 exemplifica o funcionamento dessa rede para apenas uma região de interesse selecionada que passa por camadas de convolução que extraem informações sobre os pixels dessa região condensando-os no mapa convolucional de recurso que, posteriormente, tem suas dimensões reduzidas por camadas de *pooling*. O resultado dessas camadas gera um único vetor (na figura denominado *RoI feature vector*). Esse vetor, por fim, passa por camadas de saídas que retornam as informações da classificação do objeto de interesse (usando a função de ativação Softmax) e das caixas delimitadoras (*bbox regressor*).



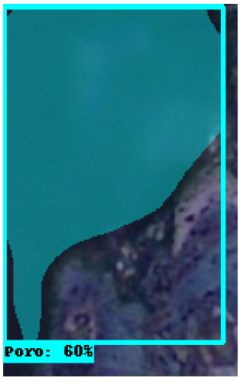
Fonte: Girshick *et al.* (2014).

Figura 16 – Ilustração do funcionamento de uma *Fast R-CNN*.

Outra extensão da R-CNN é a *Mask R-CNN* (HE *et al.*, 2017b) que além de localizar e classificar as regiões de interesse, cria uma máscara no objeto de interesse, como podemos ver na Figura 17. Essa figura representa um poro de uma microscopia de clínquer. Observe que além de introduzir uma caixa delimitadora (retângulo em azul), a rede criou uma máscara azul que cobre todo poro e ainda determina com uma probabilidade de 60% que aquele segmento da imagem é realmente um poro.

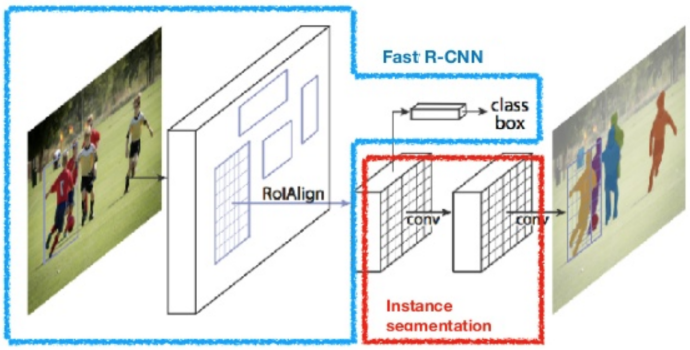
O funcionamento da *Mask R-CNN* segue as mesmas etapas da *Fast R-CNN* com o complemento de camadas convolucionais adicionais para criar as máscaras como podemos ver na Figura 18.

A *Mask R-CNN* é um método de detecção e segmentação de objetos preciso e eficiente, que tem sido amplamente utilizado em aplicações de visão computacional e robótica. Como é uma extensão da *Fast R-CNN*, essa arquitetura tem várias melhorias em relação à sua antecessora, permitindo que ela seja utilizada em problemas mais complexos e desafiadores. Devido a isso, essa metodologia se mostra adequada para o objetivo do trabalho, já que além de segmentar a



Fonte: Padilha (2022).

Figura 17 – Exemplo de máscara e caixa delimitadora gerado pela *Mask R-CNN*.



Fonte: Negri, Sant'Anna e Dutra (2013) adaptada.

Figura 18 – Exemplo de funcionamento da *Mask R-CNN*.

C_3S e os poros das microscopias, ela consegue delimitar esses objetos através das máscaras e classificá-los.

CAPÍTULO

4

COMPONENTES E CONFIGURAÇÕES TÉCNICAS DAS CNN

Nesse capítulo, apresentamos com mais detalhes algumas especificidades no ajuste de uma CNN, tais como filtros de pré-processamento de imagens para um melhor desempenho preditivo do modelo, aprendizado por transferência e métricas de como mensurar a performance do modelo.

4.1 Kernel e função de ativação

Os filtros (também chamados de kernels) utilizados nas CNN, assim como os pesos dentro das funções de ativação, podem ser estimados via *backpropagation* como nas RNA. Contudo, como os filtros também são responsáveis por captar as correlações entre os pixels, sua estrutura afeta diretamente no resultado final das análises e também no tempo de processamento.

Os kernels, normalmente, possuem tamanho menores como 3×3, 5×5, ou 7×7 (COLAÇO, 2022). O tamanho desses filtros interfere diretamente na área em que cada convolução na rede abrange, sendo que kernels menores captam informações de forma mais local e, conforme o tamanho do mesmo aumenta, é possível captar informações mais globais da imagem (IA, 2016).

Nas camadas convolucionais diversos kernels podem ser utilizados, sendo que alguns podem ser usados para detectar bordas, por exemplo, outros para detectar diferentes texturas, entre outras finalidades que sejam necessárias para as camadas de convolução.

Sendo assim, kernels de diferentes dimensões e/ou estruturas podem resultar em diferentes conclusões para um mesmo conjunto de dados. Portanto, analisar o impacto e também a precisão de diferentes filtros pode contribuir para um melhor entendimento dos dados e das redes neurais convolucionais.

Os filtros nas camadas de convolução possuem dois conceitos fundamentais atrelados a

ele para a arquitetura e complexidade de uma CNN, sendo eles o *stride* e *padding* (AGGARWAL, 2018).

O *stride* determina a quantidade de pixels em que o kernel se movimenta a cada passo da convolução. Quando o valor do *stride* é 1, por exemplo, em cada operação de convolução o kernel se movimenta um pixel (tanto na horizontal quanto na vertical), sobrepondo toda imagem ao final do processo (AGGARWAL, 2018).

Quando o valor do *stride* é maior que 1, o kernel não percorre toda imagem. Para um valor de 2, em cada operação de convolução o filtro se desloca dois pixels o que, consequentemente, faz com que o mapa de características feito pelas camadas convolucionais tenham redução de dimensionalidade.

O *padding* é o conceito de adicionar pixels extras à imagem (normalmente zeros) antes de realizar as convoluções com a finalidade de controlar o tamanho do mapa de característica (AGGARWAL, 2018).

Quando não utilizamos o *padding* e temos uma imagem de tamanho 5×5 e um kernel de tamanho 3×3, a convolução é aplicada apenas na parte da imagem na qual o kernel se encaixa completamente, ou seja, o resultado será um mapa de característica 3×3, assim como observamos na Figura 12. Se para esse mesmo exemplo, utilizarmos o *padding* e adicionarmos uma linha de zeros ao redor da imagem, o resultado da camada de convolução será um mapa de tamanho 5×5. Ou seja, o *padding* é importante para os casos onde desejamos controlar a dimensionalidade da saída da camada convolucional.

Outro componente que influencia no funcionamento da rede são as funções de ativação. Elas são ferramentas que introduzem não-linearidade nas redes neurais, permitindo que a rede aprenda uma variedade mais ampla de representações e mapeamentos dos dados de entrada para a saída. Em CNN, as funções de ativação são aplicadas após a operação de convolução transformando o valor resultante e permitindo que a rede capture relações não-lineares entre as características (GOODFELLOW; BENGIO; COURVILLE, 2016). Logo, se a função de ativação retornar uma média ponderada, por exemplo, considerando que pixels com determinada cor devem possuir um peso maior, certamente as conclusões e tempo de processamento das redes seriam afetados.

Portanto, a combinação de kernels e funções de ativação torna as CNN muito poderosas. Os kernels extraem as características dos dados e as funções de ativação permitem que as redes modelem relações não lineares complexas e aprendam com os dados. Essa relação de kernel e função de ativação é iterativa e hierárquica, pois a camada da uma rede é construída com base na camada anterior e isso faz com que seja possível reconhecer os padrões complexos presentes nas imagens.

Contudo, usaremos um API (do inglês *Application Programming Interface*) que aplica a metodologia *Mask R-CNN* nas imagens estudadas e já considera uma rede convolucional com

kernel e funções de ativação pré-definidas. Logo, alterar a estrutura interna das redes neurais foge do escopo desse trabalho que possui foco no pré-processamento das microscopias dos clínquer.

4.2 Medidas de desempenho preditivo

As CNN têm ganhado cada vez mais espaço em trabalhos de classificação e segmentação de imagem, por apresentarem a capacidade de aprender representações hierárquicas de características visuais. Devido a isso, diversos indicadores de desempenho foram criados ou adaptados para avaliar a performance preditiva desses modelos. Essas métricas podem ser divididas em dois grupos: indicadores de performance para a classificação e indicadores de performance para a segmentação de imagens.

4.2.1 Indicadores de performance para classificação de imagens

Neste trabalho iremos avaliar o poder de classificação do modelo através das métricas de acurácia, precisão, Recall e F1-Score. Todas essas métricas podem ser construídas utilizando os dados das matrizes de confusão, descrita a seguir:

- **Matriz de confusão**: é uma tabela que fornece a frequência em que cada classe da variável resposta foi classificada de maneira correta ou não pelo modelo. Devido a isso, temos uma visão detalhada do desempenho do modelo em relação a cada classe e facilita o calculo das demais métricas de performance. A matriz de confusão apresenta, normalmente, o seguinte formato: em que

Tabela 1 – Matriz de confusão.

	Previsto	
Real	Positivo	Negativo
Positivo	TP	FN
Negativo	FP	TN

  - TP (Verdadeiros positivos): Número de observações corretamente previstas como positivas;
  - FP (Falsos positivos): Número de observações incorretamente previstas como positivas;
  - FN (Falsos negativos): Número de observações incorretamente previstas como negativas; e
  - TN (Verdadeiros negativos): Número de observações corretamente previstas como negativas.

Essa matriz pode ser estendida para o caso em que a variável resposta possui mais classes e não é apenas uma variável binária. Nesse caso, os verdadeiros positivos de uma específica classe é a quantidade de observações classificadas corretamente pelo modelo como sendo dessa classe e, analogamente, os falsos negativos dessa específica classe é a quantidade de observações classificadas erroneamente como não sendo dessa classe;

- **Acurácia**: A acurácia é uma métrica importante para modelos de classificação, pois mede a proporção de predições corretas em relação ao número total de observações. Entretanto, a acurácia pode ser enganosa para dados desbalanceados, principalmente em conjuntos de dados que possui uma classe dominante (BISHOP, 2006). O valor desse indicador é calculado pela razão

$$\text{Acurácia} = \frac{\text{Número de predições corretas}}{\text{Número total de observações}}. \tag{4.1}$$

Números próximos de 1 nessa métrica indicam um bom desempenho preditivo do modelo;

- **Precisão e recall**: Precisão e recall são indicadores de performance complementares e, por isso, são normalmente usados em conjunto (BISHOP, 2006). A precisão é dada pela proporção de observações classificadas corretamente como sendo de uma determinada classe pela quantidade de observações que o modelo classificou como sendo desta classe em questão. O recall, por sua vez, mede a proporção de observações corretamente identificadas como sendo de uma determinada categoria da variável resposta pela quantidade de observações que são realmente dessa classe. Essas métricas são importantes para entender como o modelo está lidando com os verdadeiros positivos e falsos positivos. A precisão pode ser resumida pela razão:

$$\text{Precisão} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos positivos}}, \tag{4.2}$$

e a fórmula do recall é dada por:

$$\text{Recall} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}}. \tag{4.3}$$

Novamente, números próximos de 1 nessas métricas indicam um bom desempenho preditivo do modelo. Para o caso de múltiplas classes, elas podem ser calculadas para cada uma das classes e depois fazemos a sua média ponderada pelo suporte (o número de observações verdadeiras para cada classe);

- **F1-Score**: é uma métrica cujo resultado é dado pela média harmônica da precisão e do recall e, por isso, é uma forma de equilibrar esses dois indicadores (BISHOP, 2006). O F1-Score é útil em conjuntos de dados desbalanceados, onde a diferença da precisão e do recall pode ser expressiva. O F1-score é calculado por:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}. \tag{4.4}$$

Números próximos de 1 de F1-Score indicam um bom desempenho preditivo do modelo. Novamente, para o caso de múltiplas classes, essa métrica pode ser calculada para cada uma das classes e depois fazemos a sua média ponderada pelo suporte.

4.2.2 Indicadores de performance para segmentação de imagens

Neste trabalho, iremos avaliar o poder de segmentação do modelo através das métricas de IoU e coeficiente de Dice.

- **IoU (*Intersection over union*)**: mede a sobreposição entre a máscara de segmentação predita pelo modelo e a real máscara de segmentação. Quanto mais próximo de 1, mais precisa é a segmentação (LONG; SHELHAMER; DARRELL, 2015). Podemos calcular essa métrica por:

$$\text{IoU} = \frac{\text{Área de interseção}}{\text{Área da união}}, \tag{4.5}$$

  sendo a área da intersecção a área formada pelos pixels que estão demarcados tanto pela máscara predita como pela máscara verdadeira e a área da união o conjunto de pixeis da união das máscaras preditas e reais.

  A média do IoU para todas as classes em um conjunto de dados é denominada mIoU (*mean intersection over union*) e fornece uma métrica da precisão da segmentação para todas as classes. Logo, o mIoU é utilizada frequentemente como uma métrica global da performance de modelos de segmentação de imagens.

- **Coeficiente de Dice**: é calculado como o dobro da área da interseção dividida pela soma das áreas das máscaras preditas e reais (PAL; PAL, 1993), ou seja,

$$\text{Coeficiente de Dice} = \frac{2 \times \text{Área de interseção}}{\text{Área da máscara predita} + \text{Área da máscara verdadeira}}. \tag{4.6}$$

Assim, podemos determinar que os indicadores de performance discutidos são formas úteis para avaliar o poder preditivo das CNN para classificação e segmentação de imagens. Com essas métricas, podemos ter uma visão mais clara do desempenho dos modelos e, assim, tomar decisões informadas sobre como melhorá-los.

4.3 Aprendizado por transferência

O aprendizado por transferência é uma abordagem que consiste em reutilizar um modelo treinado em uma tarefa (tarefa de origem) para uma segunda tarefa relacionada (tarefa de destino). Essa técnica é útil quando a tarefa de destino tem poucos dados disponíveis para treinamento, enquanto a tarefa de origem possui um conjunto grande e bem rotulado (classificado) (YANG *et al.*, 2013).

O aprendizado por transferência foi inspirado pela capacidade humana de aplicar conhecimentos e habilidades aprendidos previamente para novas atividades e problemas. O conceito básico do aprendizado por transferência envolve três pilares principais (ZHAO; WHITE; DRUMMOND, 2019):

- Modelo pré-treinado: Um modelo que foi treinado em um grande conjunto de dados;
- Tarefa de destino: A nova tarefa que tem um conjunto de dados menor;
- Transferência de conhecimento: A adaptação do modelo pré-treinado para a nova tarefa, aproveitando os parâmetros e características aprendidos anteriormente.

O primeiro passo é treinar um modelo em uma grande quantidade de dados em uma tarefa específica. Um exemplo seria treinar uma rede neural convolucional (CNN) no *detectron2* (API do Facebook) utilizando o conjunto de dados COCO (Common Objects in Context), que contém milhões de imagens rotuladas com diversas categorias.

Posteriormente, aplicamos um modelo que foi treinado em uma tarefa de origem semelhante à tarefa de destino. Por exemplo, a CNN treinada no *detectron2* utilizando o conjunto COCO para detectar objetos pode ser usada para detectar tipos diferentes de objetos em uma tarefa de destino.

Assim, quando utilizamos o modelo de origem para a tarefa de destino, deixamos os parâmetros das camadas iniciais do modelo fixados, pois são essas camadas que geralmente aprendem características gerais dos dados, como bordas e texturas em imagens. As camadas mais altas, mais específicas da tarefa, podem ser ajustadas (ou afinadas) com base nos novos dados da tarefa de destino e assim conseguimos atender aos objetivos da nova tarefa, pois o pré-treinado é afinado usando os dados dela (GOODFELLOW; BENGIO; COURVILLE, 2016). Este processo, portanto, envolve ajustar os pesos das camadas superiores do modelo para melhor se adaptar à nova tarefa. Assim, aproveitamos o conhecimento prévio e adaptamos o modelo às novas nuances dos dados da tarefa de destino. Por fim, o modelo é validado para garantir que não esteja sobreajustado (*overfitting*) e que prediza bem novas observações.

Com o aprendizado por transferência temos diversos benefícios para as análises de dados, como por exemplo (ZHAO; WHITE; DRUMMOND, 2019):

- Redução de tempo e recursos: isso ocorre, pois treinar modelos complexos do zero possui maior custo computacional e requer um grande conjunto de dados. O aprendizado por transferência permite que modelos sejam treinados com menos dados e em menos tempo;
- Desempenho aprimorado:modelos pré-treinados geralmente já possuem uma compreensão robusta das características dos dados, o que melhora o desempenho da tarefa de destino; e

- Flexibilidade e adaptabilidade: essa abordagem permite uma rápida adaptação a novas tarefas com menos dados.

Embora o aprendizado por transferência ofereça muitos benefícios, também apresenta alguns desafios (GOODFELLOW *et al.*, 2015). São eles:

- Se a tarefa de origem e destino forem extremamente diferentes, a técnica se torna inadequada e apresenta baixo desempenho;
- Ajuste de hiperparâmetros: encontrar o equilíbrio certo entre camadas que ficarão fixas e as que serão ajustadas pode exigir experimentação cuidadosa; e
- Mesmo com dados limitados essa técnica ainda pode ter problemas de sobreajuste.

No entanto, como a disponibilidade de modelos pré-treinados está cada vez maior, o aprendizado por transferência continuará a desempenhar um papel crucial na evolução das tecnologias de inteligência artificial (YANG *et al.*, 2013). Em particular, para treinamento de CNN, ele tem sido muito utilizado e será adotado nesse trabalho, considerando que os dados de imagem de clínquer a serem analisados não são tão grandes.

4.4 Pré-processamento de imagens

Em processos de segmentação de instâncias, as predições dos modelos estatísticos são afetadas diretamente pela qualidade das imagens do banco de dados de treinamento. Logo, um banco de dados com imagens que não possuem boa qualidade ou sem bordas bem definidas para separar as regiões de interesse certamente diminui o poder preditivo do modelo.

Contudo, nem sempre é possível alterar um banco de dados devido a sua qualidade e, muitas vezes, realizar novas coletas é muito custoso. Devido a isso, realizar ajustes nas imagens utilizando técnicas computacionais para melhorar a qualidade das figuras tornou-se uma ferramenta de extrema importância. Essas técnicas são realizadas através de filtros de pré-processamento e conseguem filtrar e reforçar os pixels das bordas dos objetos de interesse.

Os filtros de pré-processamento garantem que os dados visuais estejam na melhor condição possível para serem utilizados e sua principal função é melhorar a qualidade das imagens, pois elas frequentemente contêm variações de iluminação, imperfeições e desfoques, que dificultam a análise.

Ao aplicarmos os filtros de pré-processamento, podemos suavizar essas “falhas” gerando imagens mais claras, detalhadas e definidas. Em estudos onde a precisão das imagens é crucial, como em análises de tomografia e ressonância magnética, pois os resultados afetam diretamente o diagnóstico e o tratamento de pacientes, retirar todos os ruídos que prejudicam a qualidade com o auxílio de filtros é um trabalho indispensável.

Os filtros de pré-processamento também facilitam a realização de tarefas complexas de processamento de imagem, pois preparam as imagens para operações subsequentes, como segmentação, detecção de objetos e reconhecimento de padrões. Em sistemas de reconhecimento facial, por exemplo, a qualidade da imagem de entrada é crucial para a precisão do reconhecimento. Ao aplicar filtros de pré-processamento, as características faciais são realçadas, permitindo que os algoritmos de reconhecimento funcionem de maneira mais eficaz.

Na era da inteligência artificial e do aprendizado de máquina, os filtros de pré-processamento de imagem têm um papel crítico. Modelos de aprendizado profundo, por exemplo, exigem grandes quantidades de dados de alta qualidade para treinar e operar com precisão. Filtros de pré-processamento são utilizados para limpar e preparar esses dados, removendo ruídos e normalizando a iluminação, o que resulta em conjuntos de dados mais robustos e úteis. Isso, por sua vez, melhora a capacidade dos modelos de aprendizado de máquina de realizar tarefas como classificação de imagens, detecção de objetos e segmentação semântica.

Vários filtros com diferentes objetivos estão disponíveis na literatura. A seguir, descreveremos alguns dos mais comuns e tradicionais.

4.4.1 Filtros de suavização de imagem

Utilizamos filtros de suavização de imagens quando desejamos reduzir o ruído presente em uma imagem, gerando assim uma imagem mais suave e menos detalhada. Alguns dos principais filtros de suavização incluem (GONZALEZ; WOODS, 2008):

- **Filtro de média**: também conhecido como filtro de caixa. Ele substitui cada pixel da imagem pela média dos pixels vizinhos. Um ponto negativo desse filtro é que pode desfocar detalhes finos da imagem (GONZALEZ; WOODS, 2008).

  A fórmula para o filtro de média pode ser representada como:

  $$I'(x,y) = \frac{1}{N} \sum_{i=-k}^{k} \sum_{j=-k}^{k} I(x+i, y+j),$$

  em que $I'(x,y)$ é o valor na posição (x,y) da imagem filtrada, $I(x,y)$ é o valor na posição (x,y) da imagem original, e N é o número de pixels na vizinhança em que será feita a média e k a proximidade dos vizinhos considerada;

- **Filtro Gaussiano**: este filtro suaviza a imagem através de uma média ponderada que atribui pesos maiores (baseado na função Gaussiana) aos pixels próximos ao pixel central (GONZALEZ; WOODS, 2008). Com isso, temos uma suavização mais natural e detalhada em comparação ao filtro de média.

  A função Gaussiana é dada por:

  $$G(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}},$$

em que σ é o desvio padrão da distribuição Gaussiana e $-k \leq i, j \leq k$;

- **Filtro de mediana**: este filtro substitui cada pixel pela mediana dos pixels vizinhos. Normalmente, é utilizado para remover ruídos impulsivos (*salt-and-pepper noise*) (SMOLKA; ARAKAWA, 2002) sem desfocar os contornos da imagem (GONZALEZ; WOODS, 2008).

  O processo desse filtro pode ser descrito como:

$$I'(x,y) = \text{mediana}\{I(x+i, y+j) : -k \leq i, j \leq k\},$$

  em que mediana é a mediana dos valores de intensidade dos pixels vizinhos.

4.4.2 Filtros de reforço de bordas

O reforço de bordas é uma técnica utilizada para destacar as bordas ou contornos dos objetos dentro de uma imagem o que pode melhorar significativamente o poder de segmentação dos modelos utilizados. Os principais filtros de reforço de bordas incluem (GONZALEZ; WOODS, 2008):

- **Filtro Sobel**: este filtro calcula aproximações de derivadas da imagem através de convoluções. Para isso são utilizados dois kernels de convolução que detectam mudanças horizontais e verticais. Apesar de simples, esse filtro é eficaz para o reforço de bordas (GONZALEZ; WOODS, 2008). Os kernels Sobel são:

$$G_i = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \text{ e } G_j = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}.$$

  Após aplicar os kernels G_i e G_j obtemos duas imagens, uma representando as mudanças horizontais e outra representando as mudanças verticais. Para sumarizar essas duas matrizes, calculamos a magnitude do gradiente em cada pixel. Essa magnitude é calculada como:

$$G = \sqrt{G_i^2 + G_j^2};$$

- **Filtro Prewitt**: semelhante ao Sobel, o filtro Prewitt utiliza diferentes pesos para detectar gradientes horizontais e verticais (GONZALEZ; WOODS, 2008). Os kernels Prewitt são:

$$P_i = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \text{ e } P_j = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}.$$

  A magnitude do gradiente deste filtro é calculada de maneira semelhante ao Sobel, dada por:

$$P = \sqrt{P_i^2 + P_j^2};$$

- **Filtro Roberts**: este filtro utiliza uma aproximação de diferença finita para realçar as bordas, sendo mais sensível a ruídos. Por isso, é utilizado para estudos onde a precisão é crucial (GONZALEZ; WOODS, 2008). Os operadores Roberts são:

$$R_i = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \text{ e } R_j = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

  A magnitude do gradiente é dada por:

$$R = \sqrt{R_i^2 + R_j^2};$$

- **Filtro Laplaciano**: este filtro é um operador de segunda ordem que calcula a derivada segunda da imagem. Pode ser combinado com suavização para reduzir a sensibilidade ao ruído (GONZALEZ; WOODS, 2008). O kernel Laplaciano comum é:

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

  Outra versão é:

$$L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix};$$

- **Filtro Canny**: este filtro é muito utilizado pois é considerado um dos melhores detectores de bordas. Isto ocorre porque ele é composto por várias etapas como suavização, cálculo de gradiente e supressão de não-máximos. Devido a isso, é eficiente para detectar bordas em imagens com diferentes níveis de ruído e contraste (GONZALEZ; WOODS, 2008).

  As etapas principais do algoritmo Canny são:

  1. Suavização da imagem usando um filtro Gaussiano;
  2. Utilização dos operadores de Sobel para o cálculo do gradiente da imagem;
  3. Supressão de não-máximos que afinam as bordas;
  4. Aplicação de dois limiares com histerese para detectar bordas fracas e fortes; e
  5. Por fim, se um pixel de borda fraca estiver conectado a um pixel de borda forte, ele é considerado parte da borda. Caso contrário, é descartado.

4.4.3 Filtros de remoção de ruído

A remoção de ruído melhora a qualidade da imagem e consequentemente torna a análise mais eficiente. Vários filtros são projetados para esta finalidade, entre eles se destacam:

- **Filtro bilateral**: este filtro suaviza a imagem enquanto preserva suas bordas utilizando uma combinação da filtragem espacial e de intensidade. Além de eficaz, este filtro mantém os contornos nítidos.

  A fórmula do filtro bilateral é (GONZALEZ; WOODS, 2008) dada por:

  $$I'(x) = \frac{1}{W_p} \sum_{x_i \in S} I(x_i) f_r(I(x_i) - I(x)) f_s(x_i - x),$$

  em que W_p é um termo de normalização, f_r é uma função de intensidade (pode ser a função Gaussiana), f_s é uma função espacial (pode ser a função Gaussiana) e S representa o conjunto dos pixels na vizinhança pré-definida de x;

- **Filtro Wiener**: este filtro mitiga o erro quadrático médio entre a imagem filtrada e a imagem original, utilizando modelos estatísticos para realizar a filtragem não ruidosa. Este filtro é usado para a remoção de ruído gaussiano (GONZALEZ; WOODS, 2008).

  A fórmula do filtro Wiener é dada por:

  $$H(u,v) = \frac{H^*(u,v)|F(u,v)|^2}{|H(u,v)|^2|F(u,v)|^2 + K}$$

  em que $H(u,v)$ é uma função de transferência do filtro, $F(u,v)$ é a transformada de Fourier da imagem e K é uma constante que representa o nível de ruído.

4.4.4 Filtros de realce de contraste

Filtros de realce de contraste melhoram a diferença de intensidade entre os objetos e o fundo da imagem, o que facilita a detecção e segmentação de características. Entre os filtros de realce de contraste, podemos citar:

- **Equalização de Histograma**: este filtro uniformiza a intensidade dos pixels. Para isso, ele redistribui os valores de intensidade, aumentando o contraste da imagem e, consequentemente melhorando a visibilidade de detalhes em imagens com contraste baixo (GONZALEZ; WOODS, 2008). A fórmula da equalização de histograma é:

  $$s_k = \left\lfloor (L-1) \sum_{j=0}^{k} p_r(r_j) \right\rfloor$$

  em qe s_k é o valor de intensidade equalizado, L é o número de níveis de intensidade, e $p_r(r_j)$ é a probabilidade do nível de intensidade r_j;

- **Correção gama**: a correção gama utiliza uma função exponencial para ajustar a luminosidade da imagem que realça detalhes em áreas escuras ou claras da imagem. Sua fórmula é (POYNTON, 1993):

  $$I'(x,y) = I(x,y)^{\gamma}$$

  em que γ é o valor da correção gama.

Por fim, podemos observar que os filtros de pré-processamento de imagem têm uma ampla gama de aplicações, podendo ser utilizados para detecção de objetos, reconhecimento facial e análise de vídeo (GUPTA; SINGHAL, 2017). Essas técnicas podem ser úteis tanto na medicina tratando imagens médicas (como tomografias e ressonâncias magnéticas), melhorando a precisão do diagnóstico, como também podem ser aplicadas na segurança, em sistemas de monitoramento para analisar imagens de câmeras de vigilância e detectar atividades suspeitas.

Logo, os filtros de pré-processamento de imagem são ferramentas essenciais que melhoram significativamente a qualidade das imagens e a eficiência das análises subsequentes. Ou seja, a capacidade de remover ruído, melhorar a clareza e realçar características importantes torna esses filtros indispensáveis para qualquer sistema de processamento de imagens na busca resultados mais precisos e confiáveis.

4.5 *Detectron2*

Nesse trabalho os modelos utilizados serão gerados utilizando a API *Detectron2* (WU *et al.*, 2019), sendo esta uma biblioteca desenvolvida pela equipe de pesquisa de inteligência artificial do Facebook (Facebook AI Research (FAIR), 2024) que possui diversas finalidades de visão computacional incluindo modelos de classificação e segmentação de imagens e vídeos.

O *Detectron2* é o sucessor do *Detectron* (GIRSHICK *et al.*, 2018) e oferece melhorias significativas em termos de flexibilidade, desempenho e facilidade de uso em relação a seu antecessor, sendo amplamente utilizado pela comunidade de pesquisa devido à sua robustez e capacidade de fornecer resultados de alta qualidade.

Como sua arquitetura é modular, o *Detectron2* é altamente configurável, pois permite combinar diferentes componentes de modelos de forma flexível. Assim, temos maior facilidade para personalizar os modelos e podemos mudar sua estrutura para atender as necessidades específicas do estudo, sem a necessidade de modificar o código-fonte subjacente.

Outro ponto positivo no *Detectron2* é que ele fornece uma variedade de modelos pré-treinados em grandes conjuntos de dados, como o COCO (Common Objects in Context). Esses modelos podem facilmente ser utilizados em novos conjuntos de dados através do aprendizado por transferência.

Além disso, essa API possui uma boa documentação, com tutoriais e exemplos, que somados a interface de alto nível para treinar e avaliar modelos tornam essa ferramenta simples e viável de ser utilizada. Dentre suas principais funcionalidades podemos citar a detecção de objetos, segmentação de instâncias, segmentação semântica (classifica cada pixel em uma categoria, sem distinguir diferentes instâncias do mesmo objeto) e segmentação panóptica (combinação da segmentação de instâncias e a segmentação semântica, ou seja, o modelo identifica cada instância do objeto individualmente e também classifica todos os pixels não

pertencentes a essas instâncias).

A detecção de objetos no *Detectron2* permite identificar e localizar objetos em imagem através de modelos como *Faster R-CNN*, *RetinaNet* e *EfficientDet*, oferecendo várias opções de precisão e desempenho para cada um deles.

A segmentação de instâncias no *Detectron2* é feita através, principalmente, da *Mask R-CNN*. Também é possível ajustar diversas configurações para esses modelos que torna esse API uma ferramenta potente para segmentação de imagens.

Logo, podemos perceber que o *Detectron2* é uma ferramenta poderosa e versátil para classificação e segmentação de imagens devido a sua arquitetura modular, facilidade de uso e suporte a modelos pré-treinados.

CAPÍTULO

5

ANÁLISE DAS IMAGENS DE CLÍNQUER

Nesse capítulo, aplicamos as CNN combinadas com diferentes filtros de pré-processamento em imagens de clínquer na tentativa de identificar (segmentar) e classificar os cristais de alita. Também analisamos e comparamos o desempenho dos diferentes modelos utilizados.

Este capítulo, portanto, representa o cerne do trabalho realizado, fornecendo uma análise detalhada do desempenho e das descobertas alcançadas com o uso desses modelos avançados de segmentação e classificação. Os códigos em Pyhton utilizados para desenvolver esse trabalho e obter os resultados a seguir estão disponíveis nos apêndices desse relatório.

5.1 O banco de dados

Para entender o conjunto de dados analisado, primeiramente, é importante entender a diferença dos bancos de dados estruturados, semiestruturados e não estruturados.

Os bancos de dados estruturados são organizados de forma fixa e organizada, como por exemplo, em tabelas, com uma estrutura de linhas e colunas e campos bem definidos onde os dados são armazenados. Exemplos comuns incluem bancos de dados relacionais, como MySQL, PostgreSQL, etc.

Os bancos de dados semiestruturados não seguem um esquema fixo, porém ainda possuem alguma organização que auxilia na análise dos dados. Um exemplo de banco de dados semiestruturado são e-mails que possuem campos bem definidos como assunto, remetente e destinatário, mas o conteúdo do corpo da mensagem é flexível e pode variar sua estrutura e formato.

Por fim, os bancos de dados não estruturados contêm dados que não possuem uma organização predefinida ou uma estrutura fixa. Um exemplo de banco de dados não estruturado são imagens que é o foco deste trabalho e devido ao fato de não possuírem formato pré definido ou organizado não podemos realizar análises descritivas convencionais nelas.

O banco de dados analisado foi disponibilizado por Padilha (2022) da Universidade Federal do Ceará, campus de Russas, e se trata de 122 imagens coletadas diretamente de uma planta cimenteira, com a técnica descrita no Capítulo 2 e proporcionando uma amostra representativa das condições reais de produção.

O formato das imagens no momento da captura é TIF (do inglês, *Tag Image File Format*), que é conhecido por manter a alta qualidade das imagens e preservar detalhes essenciais para as análises que serão realizadas.

Para o treinamento inicial da *Mask R-CNN*, das 122 imagens do banco de dados, 81 (67%) imagens foram escolhidas aleatoriamente e utilizadas. Das restantes, 21 (17%) foram separadas para teste e 20 (16%) para validação. As imagens de teste foram utilizadas para identificar alguns melhores hiperparâmetros da rede em cada cenário e as imagens de validação para calcularmos as métricas de desempenho e compararmos os diferentes modelos ajustados. Este conjunto de dados pode ser considerado pequeno para um bom e preciso treinamento da CNN, porém como o *Detectron2* utiliza a técnica de aprendizado por transferência, o efeito do tamanho reduzido da amostra nas estimações tende a ser mitigado.

Na Figura 19, é possível visualizar uma amostra do conjunto de dados que foi utilizado no treinamento do modelo, sem aplicação de nenhum filtro de pré-processamento.

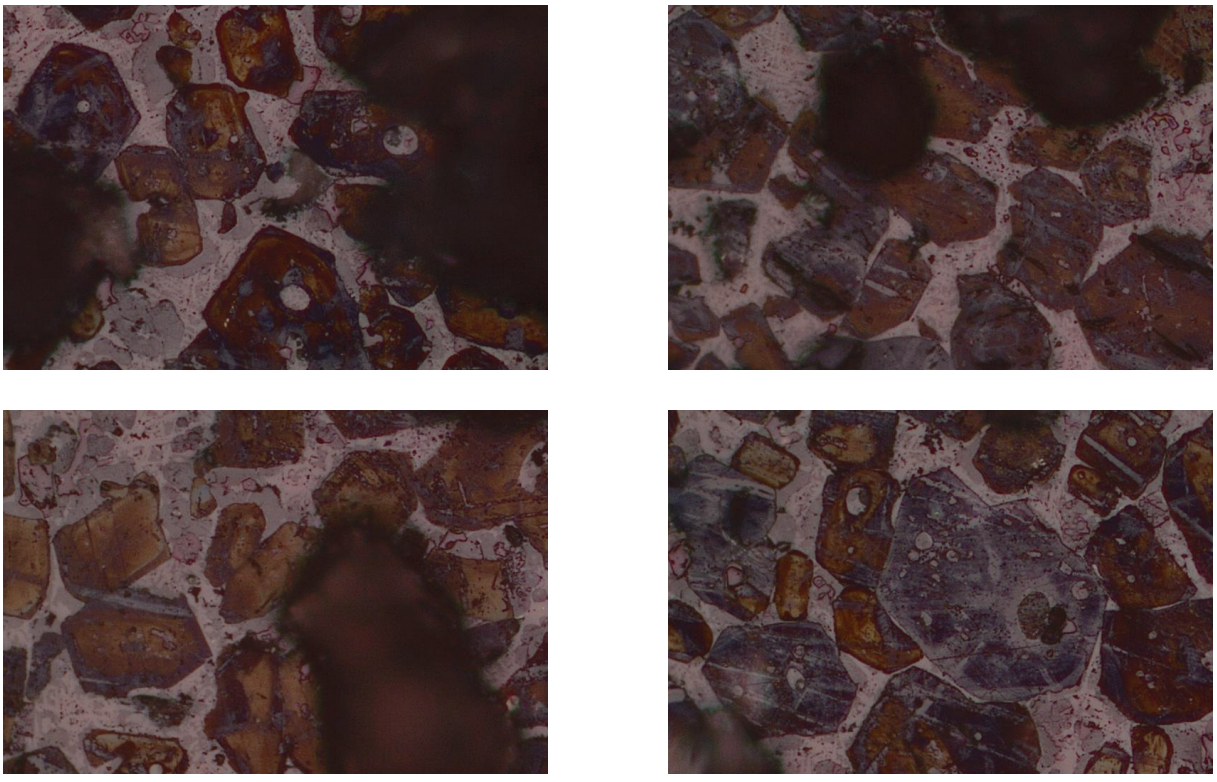


Figura 19 – Amostras do banco de dados de treinamento sem aplicação de filtro de pré processamento.

O conjunto de dados já foi previamente rotulado por Padilha (2022) e os cristais de alita presentes nas imagens foram classificados como: subdiomórficos, xenomórficos e idiomórficos. Essas rotulações foram realizadas com o auxílio do Roboflow (2024) que é uma plataforma que

facilita a classificação dos conjuntos de dados para utilização em modelos de classificação e segmentação de imagens. Além disso, a ferramenta suporta a exportação de dados em diversos formatos compatíveis com diversas estruturas de métodos de aprendizado de máquina, facilitando o treinamento dos modelos. A Figura 20 mostra algumas imagens do conjunto de dados com suas pré-rotulações. Aqui vale mencionar que as três classes de cristais são muito desequilibradas entre si, sendo os cristais subdiomórficos a grande maioria e os idiomórficos uma pequena minoria. Os subdiomórficos representam, aproximadamente, 72% dos cristais, os xenomórficos 18% e os idiomórficos 10%.

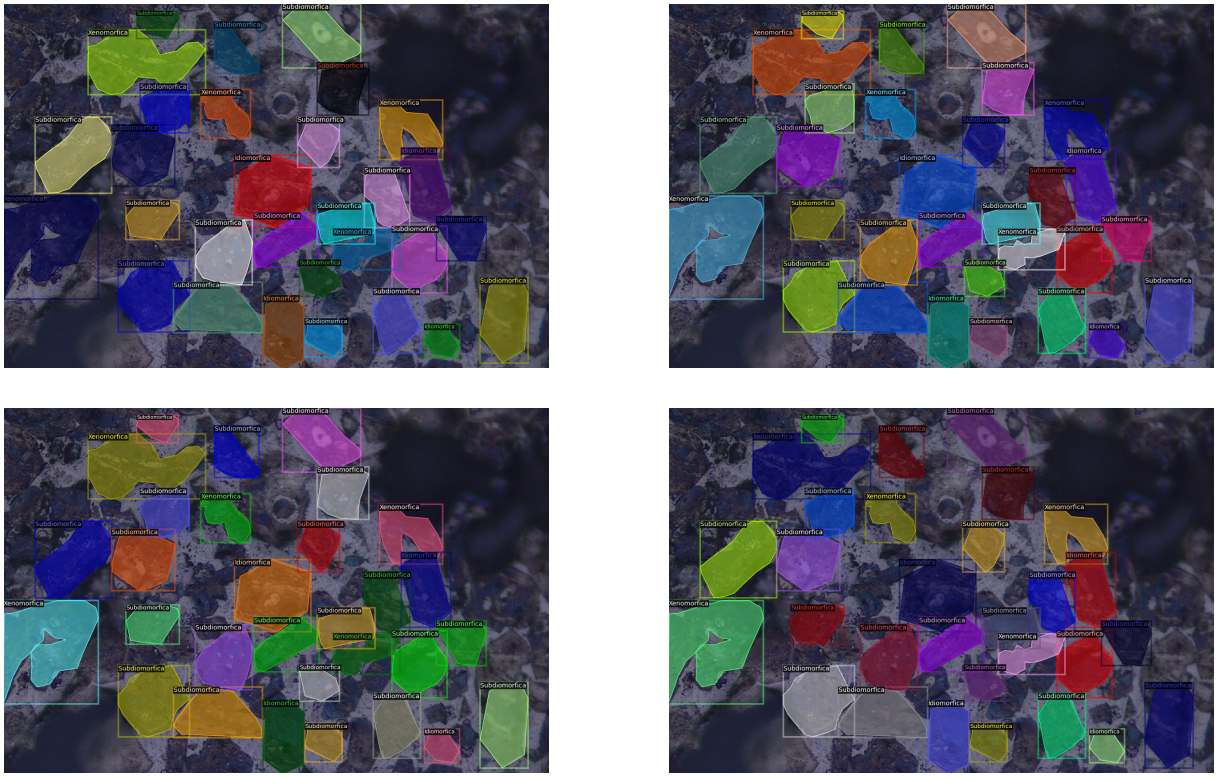


Figura 20 – Amostras do banco de dados com as rotulações feitas pelo Roboflow.

5.2 Execução do pré-processamento

Como anteriormente comentado, o pré-processamento das imagens utilizando filtros pode ser uma forma eficiente de diminuir o custo computacional e aumentar a performance do modelo. Por isso, aplicamos os filtros de Sobel, Laplaciano, Canny, bilateral, Prewitt e Roberts nos dados, criando assim 6 novos conjuntos de dados.

A escolha desses filtros foi baseada em suas propriedades e eficácia na realce de bordas, redução de ruído e preservação de detalhes importantes. Os filtros utilizados são:

- **Filtro Sobel**: como já visto, o filtro Sobel é utilizado para detecção de bordas e é considerado eficaz na realce de bordas suaves e na detecção de mudanças de intensidade, o

que é crucial para identificar contornos claros dos componentes mineralógicos. A Figura 21 contêm alguns exemplos das imagens utilizadas no treinamento do modelo após a utilização desse filtro.

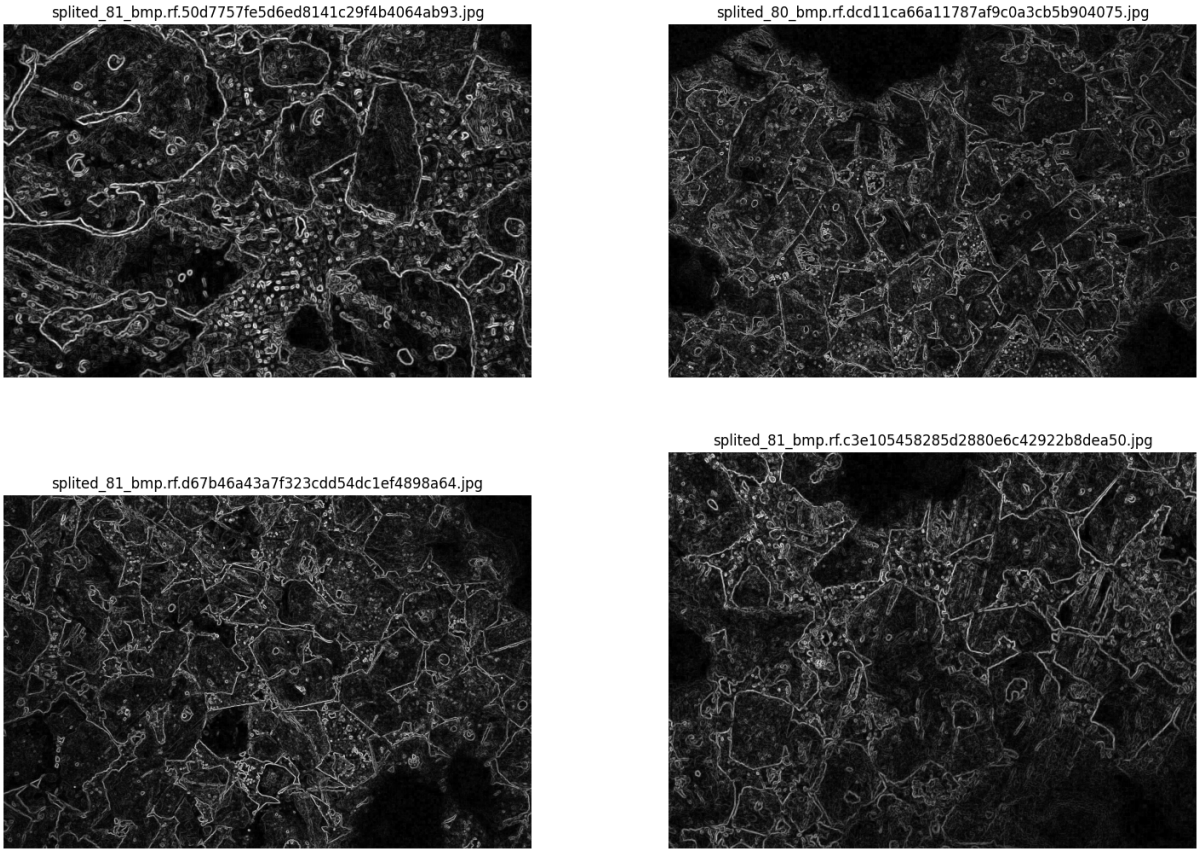


Figura 21 – Amostras do banco de dados de treinamento com a aplicação do filtro Sobel.

Quando aplicado às imagens de clínquer, o filtro de Sobel revelou-se bom para distinguir as principais linhas e contornos dos cristais. Este efeito é alcançado através da convolução da imagem com os kernels de Sobel, que aumentam as alterações de intensidade na direção horizontal e vertical, resultando em bordas nítidas e bem definidas.

Logo, a aplicação do filtro de Sobel em imagens de clínquer resultou em uma visualização mais nítida dos cristais, com as bordas dos cristais se destacando claramente contra o fundo. Essa melhoria na detecção de bordas é essencial para a classificação e segmentação destes objetos, pois a identificação das bordas afeta diretamente a qualidade dos resultados obtidos;

- **Filtro Laplaciano**: esse filtro detecta bordas baseando-se na segunda derivada da imagem, sendo sensível a regiões com mudanças bruscas de intensidade. A Figura 22 contêm alguns exemplos das imagens utilizadas no treinamento do modelo, após a utilização desse filtro.

  A aplicação do filtro Laplaciano nas imagens microscópicas de clínquer revelou uma alteração significativa na qualidade visual das imagens, resultando em uma aparência mais

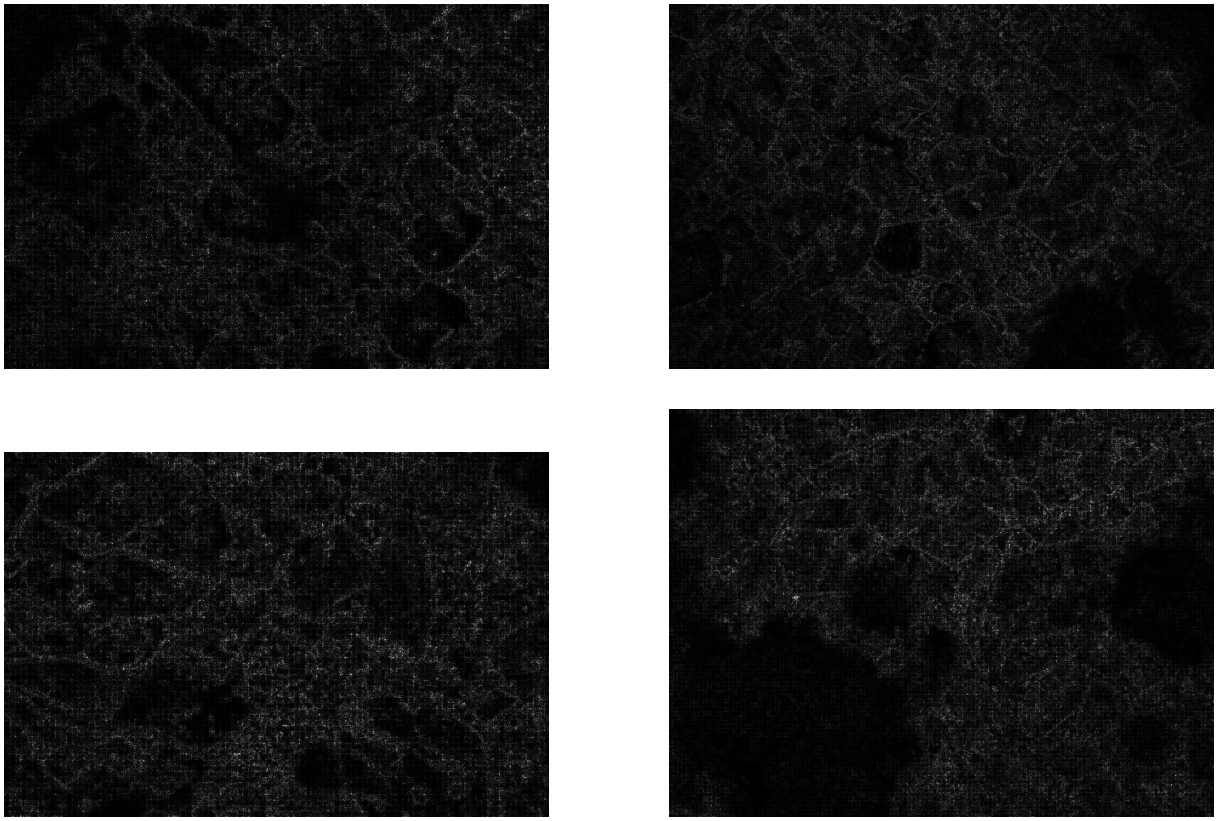


Figura 22 – Amostras do banco de dados de treinamento com a aplicação do filtro Laplaciano.

escura e menos definida. Como esse filtro é sensível à variação de intensidade pode levar a uma intensificação de ruídos e uma redução na clareza geral da imagem.

Além disso, o filtro Laplaciano pode criar uma aparência mais sombreada na imagem, resultando em uma perda de definição. Essa alteração compromete a capacidade de identificar claramente os cristais e suas bordas, afetando negativamente as performances dos modelos;

- **Filtro Canny**: é conhecido por sua capacidade de detectar bordas de forma mais robusta, sendo eficaz na identificação de bordas nítidas e contínuas, que é extremamente importante para uma boa performance dos modelos aplicados. A Figura 23 contêm alguns exemplos das imagens utilizadas no treinamento do modelo, após a utilização desse filtro.

  O filtro Canny se destacou no pré-processamento das imagens microscópicas de clínquer por conseguir realçar as bordas e eliminar ruídos de forma eficaz, proporcionando uma imagem final limpa e com alta definição. Essa capacidade de filtrar ruídos permite que os detalhes dos cristais sejam visualizados com clareza, facilitando a identificação e análise desses objetos. Ou seja, esse filtro tem grande chances de melhorar o desempenho dos modelos aplicados;

- **Filtro bilateral**: é útil para a redução de ruído enquanto preserva as bordas. Ele suaviza as áreas homogêneas da imagem, mantendo os detalhes importantes das bordas, essencial

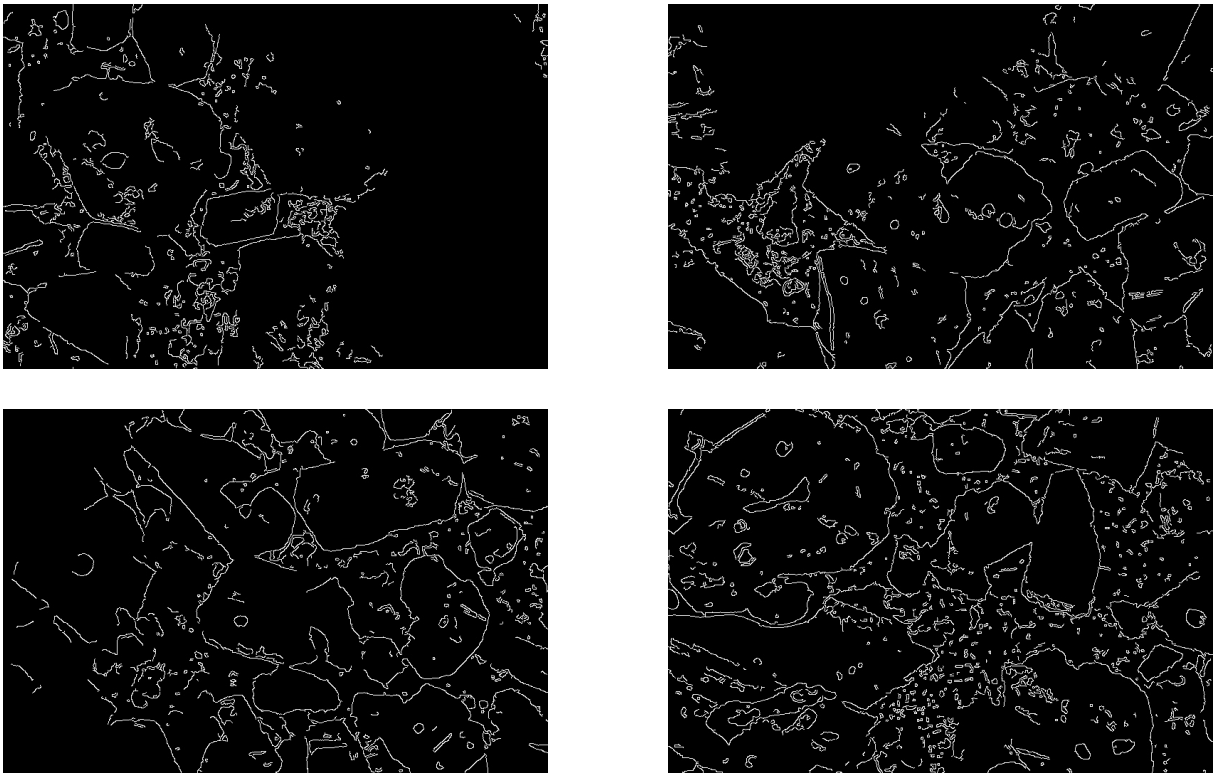


Figura 23 – Amostras do banco de dados de treinamento com a aplicação do filtro Canny.

para a análise dos cristais. A Figura 24 contêm alguns exemplos das imagens utilizadas no treinamento do modelo, após a utilização desse filtro.

O filtro bilateral preservou os aspectos originais das imagens microscópicas de clínquer, ao mesmo tempo em que suavizou e melhorou a definição das imagens. Ao aplicar uma suavização seletiva, o filtro foi capaz de reduzir o ruído e realçar os contornos dos cristais, tornando a imagem mais nítida e detalhada o que pode facilitar no treinamento do modelo;

- **Filtro Prewitt**: é útil para capturar diferentes características das bordas que podem ser perdidas por outros filtros. A Figura 25 contêm alguns exemplos das imagens utilizadas no treinamento do modelo, após a utilização desse filtro.

  O filtro Prewitt mostrou ser eficaz para realçar as bordas dos cristais de clínquer, destacando claramente os contornos desses objetos de interesse. Isso certamente se deve ao fato que o filtro é projetado para detectar gradientes de intensidade, tanto na direção horizontal quanto vertical, o que contribui para a identificação precisa das bordas.

  No entanto, apesar do filtro Prewitt ser eficaz na realce de bordas, ele tende a amplificar o ruído presente nas áreas próximas aos cristais. Esse aumento no ruído pode resultar em uma imagem com mais granulações indesejadas, o que pode complicar a análise e interpretação das características dos cristais pelo modelo.

  Portanto, enquanto o filtro Prewitt melhora a definição das bordas, o aumento do ruído pode prejudicar sua qualidade impactar negativamente na performance dos modelos utilizados;

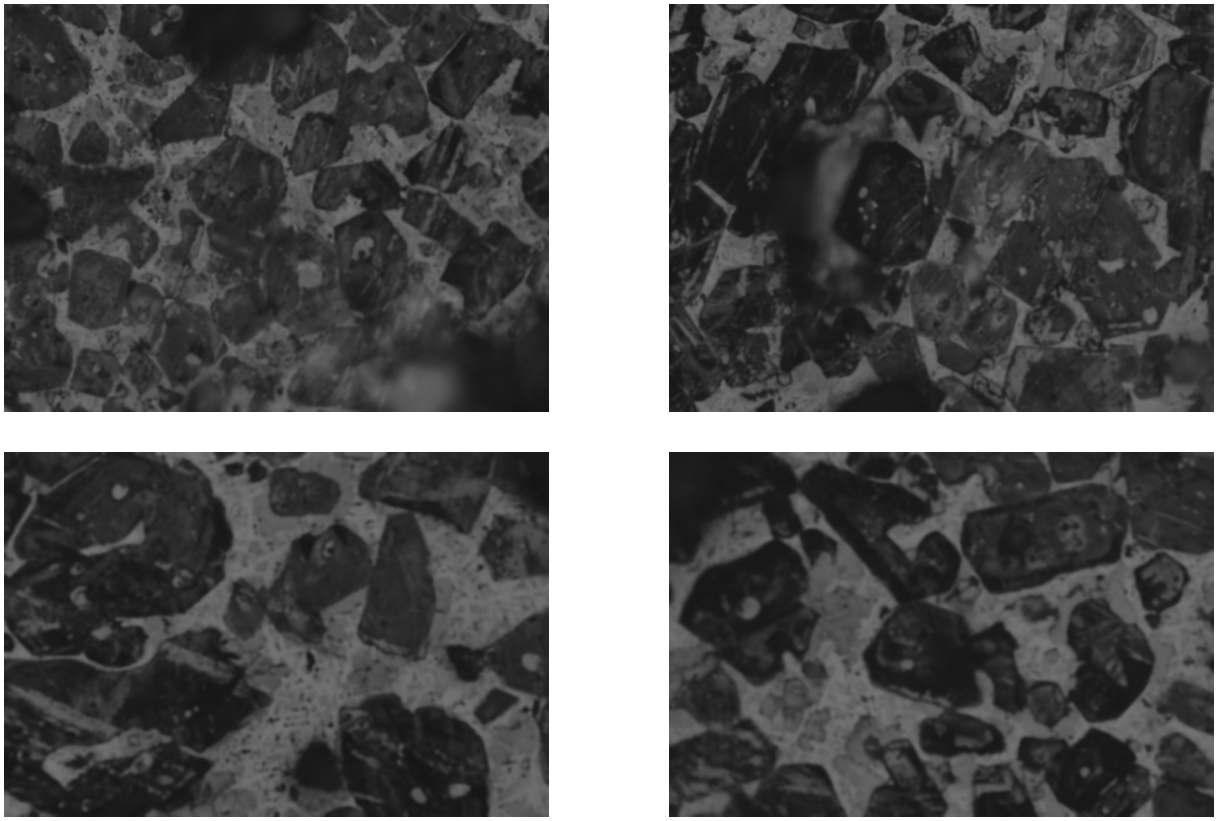


Figura 24 – Amostras do banco de dados de treinamento com a aplicação do filtro bilateral.

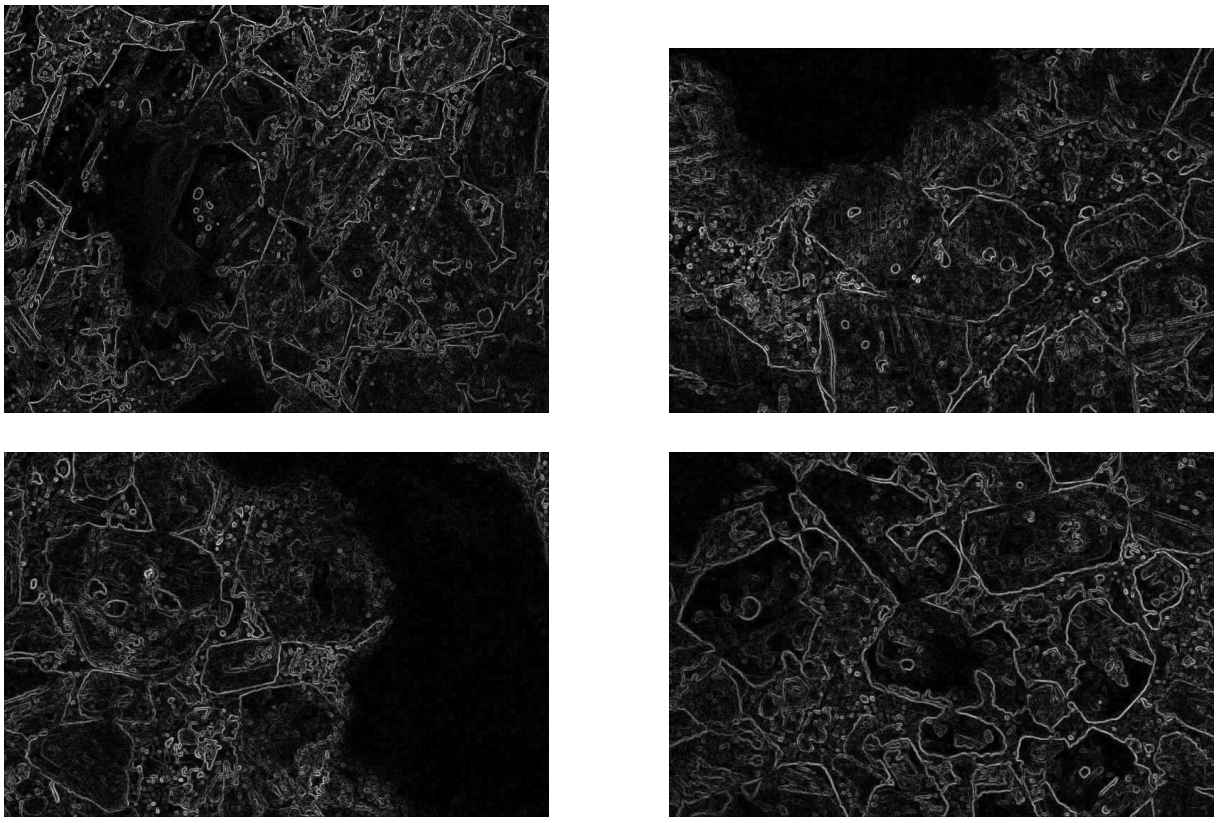


Figura 25 – Amostras do banco de dados de treinamento com a aplicação do filtro Prewitt.

- **Filtro Roberts**: é usado para detecção de bordas, baseando-se na diferença entre pixels diagonais adjacentes que pode ser útil no treinamento do modelo. A Figura 26 contêm alguns exemplos das imagens utilizadas no treinamento do modelo, após a utilização desse filtro.

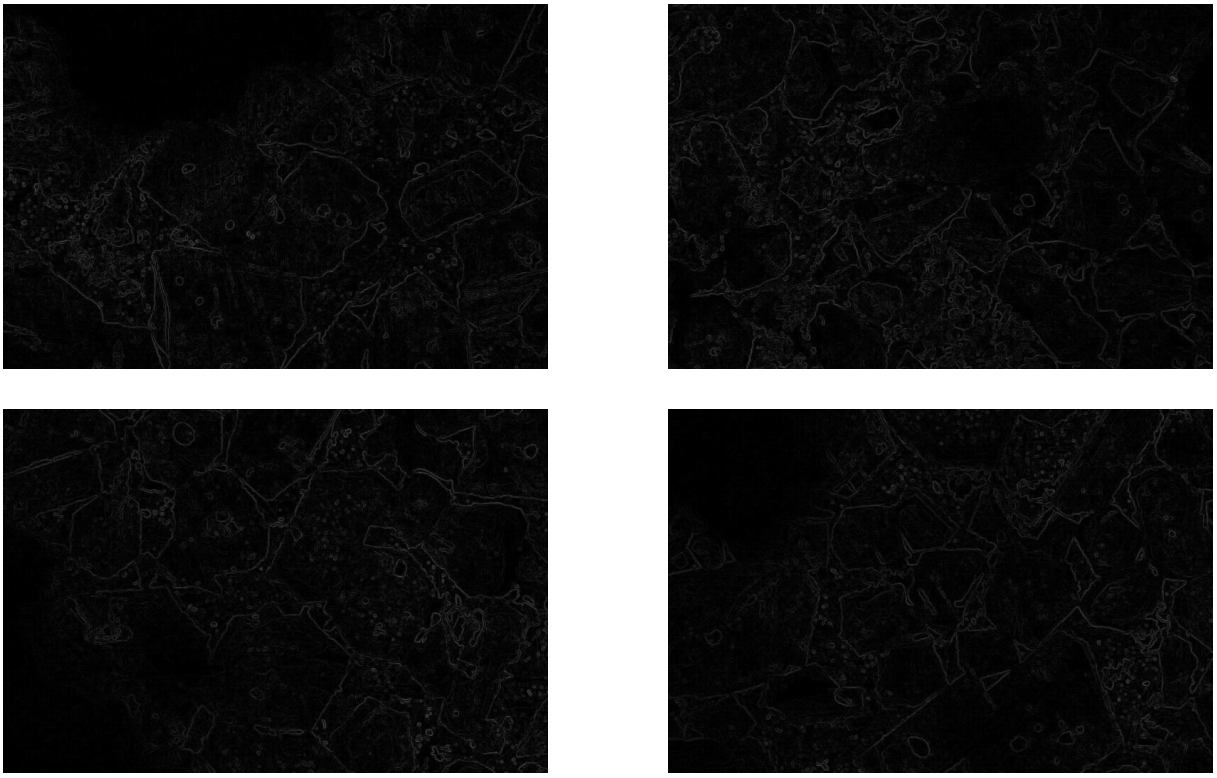


Figura 26 – Amostras do banco de dados de treinamento com a aplicação do filtro Roberts.

O filtro Roberts, ao ser aplicado às imagens de clínquer, apresentou um impacto visível na qualidade das imagens, gerando uma imagem mais escura e com bordas pouco definidas.

Sua aplicação neste caso não conseguiu realçar significativamente as bordas dos cristais de clínquer e resultou em uma definição muito reduzida em comparação com outros filtros. As bordas dos cristais, em vez de se destacarem, ficaram menos nítidas, comprometendo a clareza dos objetos presentes nas imagens.

Além da definição reduzida das bordas, o filtro Roberts também escureceu a imagem, o que pode dificultar ainda mais a identificação e análise dos componentes mineralógicos.

Portanto, enquanto o filtro Roberts pode ter seu uso em contextos específicos, para este caso ele não conseguiu fornecer a definição e a clareza necessárias para uma análise eficiente das características dos cristais de clínquer.

Por fim, vemos então que grande parte dos filtros escolhidos realmente resultam em imagens que destacam as características essenciais das imagens originais para facilitar o treinamento do modelo.

5.3 Resultados

Neste seção, apresentamos e analisamos os resultados obtidos com a aplicação de diferentes abordagens de pré-processamento e balanceamento de classes em modelos de segmentação de imagens utilizando a arquitetura *Mask R-CNN*. A análise foca na comparação da eficácia de diferentes técnicas de pré-processamento e balanceamento de classes para a segmentação de imagens microscópicas de clínquer.

Inicialmente, o Modelo 1 (M1) foi treinado utilizando as imagens originais, sem qualquer tipo de pré-processamento ou balanceamento adicional das três classes de cristais. Este modelo serve como referência para avaliar o impacto das técnicas subsequentes. Em seguida, o Modelo 2 (M2) também foi treinado nas imagens originais, mas aplicando o balanceamento das classes. Este balanceamento foi realizado considerando os pesos como o inverso da frequência com que cada tipo de cristal aparece no banco de dados de treinamento. No banco de dados, a classe subdiomórfica é a mais representada, enquanto as classes xenomórfica e idiomórfica aparecem com menos frequência. Dessa maneira, o balanceamento atribuiu pesos maiores para as classes menos representadas, a fim de mitigar o viés e melhorar a precisão na segmentação dos cristais menos frequentes.

Além do balanceamento das classes, foram exploradas várias técnicas de pré-processamento aplicadas ao banco de dados balanceado. São elas:

- Modelo 3 - Sobel (M3-Sob): *Mask R-CNN* treinada com o banco de dados utilizando balanceamento das classes e imagens pré-processadas com o filtro de Sobel;
- Modelo 3 - Laplaciano (M3-Lap): *Mask R-CNN* treinada com o banco de dados utilizando balanceamento das classes e imagens pré-processadas com o filtro de Laplaciano;
- Modelo 3 - Canny (M3-Can): *Mask R-CNN* treinada com o banco de dados utilizando balanceamento das classes e imagens pré-processadas com o filtro de Canny;
- Modelo 3 - bilateral (M3-Bil): *Mask R-CNN* treinada com o banco de dados utilizando balanceamento das classes e imagens pré-processadas com o filtro bilateral;
- Modelo 3 - Prewitt (M3-Pre): *Mask R-CNN* treinada com o banco de dados utilizando balanceamento das classes e imagens pré-processadas com o filtro de Prewitt; e
- Modelo 3 - Roberts (M3-Rob): *Mask R-CNN* treinada com o banco de dados utilizando balanceamento das classes e imagens pré-processadas com o filtro de Roberts.

Todos os modelos listados foram treinados com backbone (parte computacional do modelo responsável por extrair as características das imagens) ResNet-50 e arquitetura FPN (Feature Pyramid Network) com os pesos pré-treinados no dataset COCO.

A escolha do backbone ResNet-50 se deve ao fato de ser uma rede de convolução profunda que equilibra o desempenho e complexidade computacional. Utilizamos também a especificação FPN pois permite a detecção de objetos de diferentes tamanhos de forma eficaz.

Para o número de amostras de regiões de interesse por imagem que serão processadas durante o treinamento (outro parâmetro da rede) fixamos o valor 64, que é frequentemente adotado na literatura, pois oferece um bom equilíbrio entre eficiência do modelo e uso de memória (HE *et al.*, 2017b).

Limitamos a quantidade de imagens processadas por iteração no treinamento para duas imagens por vez. Esse valor foi definido pensando nas limitações de memória da GPU, pois valores menores desse parâmetro permite que o treinamento seja feito utilizando GPUs com menos memória.

A taxa de aprendizado foi configurada em 0,001, que é um valor frequentemente utilizado para treinar redes convolucionais. Taxas de aprendizado muito altas podem levar à oscilação e não convergência e taxas muito baixas resultam em um treinamento muito lento. O número de iterações máximas foi definido como 1000, o que limita o número de atualizações de pesos realizadas durante o treinamento. Para conjunto de dados pequenos, como o utilizado, um número excessivo de iterações pode levar ao *overfitting*.

O formato da máscara foi definido como *bitmask*, que é um formato binário de máscaras usualmente utilizado em tarefas de segmentação de instâncias (HE *et al.*, 2017b). Na representação dessa máscara cada píxel da imagem analisada assume valor 1 se faz parte da máscara e 0 caso contrário.

Para carregar os dados de entrada de forma mais rápida e eficiente, podemos utilizar diversos processos de CPU (*threads*) paralelamente. Para os modelos treinados utilizamos um valor de 2 *threads* que é suficiente para carregar as imagens, sem sobrecarregar o processamento.

Em suma, a configuração do modelo utilizada foi escolhida com a finalidade de equilibrar a eficiência computacional com o desempenho de segmentação de instâncias. Para fornecer uma visão ampla do desempenho desses modelos, calculamos diversas métricas para avaliar seu poder de classificação e segmentação das imagens.

5.3.1 Performance geral

A Tabela 2 apresenta os indicadores de desempenho geral para os diversos modelos de segmentação e classificação aplicados às imagens de clínquer. Os indicadores incluem acurácia, precisão, recall e F1-Score, que são fundamentais para avaliar a eficácia de cada abordagem. Ao analisar o desempenho geral de cada modelo, podemos notar diferenças significativas no desempenho de cada modelo ajustado.

Como esperado, o modelo sem nenhum tratamento de balanceamento ou pré-processamento

(M1) apresentou o pior desempenho geral, com a menor acurácia (0,35) e recall (0,35). Embora sua precisão seja de 0,56, que indica que a maior parte das predições positivas que ele realiza tendem a ser corretas, o valor baixo de recall indica que ele falha em classificar uma parte dos cristais de clínquer, provavelmente os cristais das classes minoritárias ou por identificar como sendo cristais algumas instâncias das imagens que são foram rotuladas como tais, por erro de marcação manual nas bases ou por não serem realmente cristais. Isso afeta diretamente o valor do F1-Score (0,39) e indica que esse modelo não é eficaz na classificação e segmentação dos objetos desejados. Aqui reforçamos que além de classificar os cristais como sendo subdiomórficos, xenomórficos e idiomórficos, os modelos precisam, primeiramente, identificar quais instâncias nas imagens, se tratam, de fato, de um cristal de alita e, então classificá-lo. Dessa maneira, alguns cristais podem não ser identificados e classificados e outras instâncias não rotuladas como cristais podem ser identificadas como tal e classificadas. Ambos os casos contam como classificações erradas no cálculo dessas métricas de desempenho. Nesse estudo, observamos uma tendência dos modelos identificarem mais instâncias como cristais do que o número de cristais realmente rotulados nas imagens. Esse comportamento, talvez reflita que as imagens utilizadas nesse estudo não foram precisamente tratadas e rotuladas e resultados melhores seriam obtidos com imagens melhores marcadas.

Métrica	**M1**	**M2**	**M3-Sob**	**M3-Lap**	**M3-Can**	**M3-Bil**	**M3-Pre**	**M3-Rob**
Acurácia	0,35	0,53	0,50	0,49	0,55	0,54	0,48	0,52
Precisão	0,56	0,63	0,60	0,55	0,60	0,63	0,60	0,59
Recall	0,35	0,53	0,50	0,49	0,55	0,54	0,48	0,52
F1-Score	0,39	0,56	0,53	0,51	0,57	0,58	0,53	0,55

Tabela 2 – Indicadores de desempenho para classificação de imagens no geral.

Contudo, após o tratamento do desbalanceamento entre as classes, o Modelo 2 (M2) apresentou uma melhora significativa em comparação com o M1, com uma acurácia de 0,53 e um F1-Score de 0,56. Isso confirma que o tratamento realizado neste modelo resultou em um aumento da qualidade das predições, tanto em termos de precisão (0,63) quanto de recall (0,53).

Quando olhamos para os modelos com técnicas de pré-processamento, o Modelo 3 com filtro Sobel (M3-Sob) mostrou uma acurácia de 0,50, com uma precisão de 0,60 e um F1-Score de 0,53. Embora esse modelo não tenha superado o M2, ele ainda mantém um desempenho razoável. O Modelo 3 com filtro Laplaciano (M3-Lap) teve um desempenho semelhante ao M3-Sob, com acurácia de 0,49 e F1-Score de 0,51.

O Modelo 3 com filtro Canny (M3-Can) se destacou como o melhor entre todos, apresentando a maior acurácia (0,55) e o maior F1-Score (0,57). Além disso, ele mantém uma precisão de 0,60 e recall de 0,55, o que indica que ele consegue tanto identificar corretamente os objetos

quanto minimizar o número de falsos negativos. A técnica de pré-processamento com o filtro de Canny parece ter sido a mais eficaz para a classificação das imagens aqui consideradas, oferecendo o melhor equilíbrio entre as métricas.

O Modelo 3 com filtro bilateral (M3-Bil) também teve um desempenho próximo ao M3-Can, com uma acurácia de 0,54 e um F1-Score de 0,58, o segundo melhor entre todos os modelos. Sua precisão de 0,63 é alta, o que sugere que o M3-Bil também realiza predições bastante corretas sendo uma excelente alternativa ao filtro de Canny.

Por fim, o Modelo 3 com filtro Prewitt (M3-Pre) e o Modelo 3 com filtro Roberts (M3-Rob) tiveram desempenhos medianos.

Essa análise revela que o uso de técnicas de pré-processamento pode trazer melhorias substanciais no desempenho dos modelos de segmentação e classificação de imagens, com destaque para o filtro de Canny e o bilateral. Estudos nos quais não se aplica os filtros de pré-processamento, as técnicas de balanceamento de classes são bastante necessárias quando elas forem muito desbalanceadas para se obter melhores resultados.

A Tabela 3 resume outros indicadores de desempenho, especificamente o IoU (intersection over union) e o coeficiente de Dice, para os modelos de segmentação aplicados às imagens de clínquer. Estes indicadores são essenciais para avaliar a eficácia dos modelos em segmentar os cristais corretamente. A análise desses indicadores de segmentação complementa as conclusões obtidas na análise dos indicadores de classificação.

Métrica	**M1**	**M2**	**M3-Sob**	**M3-Lap**	**M3-Can**	**M3-Bil**	**M3-Pre**	**M3-Rob**
IoU	0,78	0,82	0,78	0,71	0,83	0,81	0,78	0,80
Coeficiente de Dice	0,85	0,89	0,85	0,80	0,89	0,88	0,85	0,88

Tabela 3 – Indicadores de desempenho para segmentação de imagens geral.

O Modelo com filtro Canny (M3-Can) novamente se destaca em ambas as métricas. Em termos de segmentação, ele obteve o melhor desempenho com o maior valor de IoU e coeficiente de Dice entre os modelos que aplicaram técnicas de pré-processamento. Esta melhoria na segmentação confirma a eficácia do filtro Canny em realçar as bordas, o que se alinha com os melhores resultados obtidos na análise de classificação. A combinação do filtro Canny com o tratamento para desbalanceamento parece ser a abordagem mais eficaz, destacando-se em ambos os critérios.

O Modelo 2 (M2) também apresenta bons resultados tanto na classificação quanto na segmentação, assim como o Modelo M3-Bil (bilateral). O Modelo 1 (M1), sem técnicas de balanceamento ou pré-processamento, teve o desempenho inferior tanto na classificação quanto na segmentação.

5.3.2 Cristais subdiomórficos

A Tabela 4 e 5 apresentam respectivamente os indicadores de performance de classificação e segmentação de cada modelo para os cristais subdiomórficos.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
Acurácia	0,25	0,45	0,43	0,37	0,50	0,50	0,42	0,47
Precisão	0,69	0,75	0,72	0,71	0,74	0,75	0,72	0,72
Recall	0,28	0,53	0,52	0,44	0,61	0,60	0,50	0,57
F1-Score	0,40	0,62	0,60	0,54	0,67	0,67	0,59	0,63

Tabela 4 – Indicadores de desempenho para classificação de cristais subdiomóficos.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
IoU	0,78	0,84	0,80	0,74	0,79	0,85	0,80	0,79
Coeficiente de Dice	0,85	0,90	0,86	0,82	0,86	0,91	0,87	0,87

Tabela 5 – Indicadores de desempenho para segmentação de cristais subdiomórficos.

Na classificação dos cristais subdiomórficos, observamos números muito parecidos com os números do desempenho geral dos modelos e isso faz sentido por ser a classe de cristais majoritária nas imagens analisadas. No entanto, a aplicação dos filtros de pré-processamento Canny (M3-Can) e bilateral (M3-Bil) apresentam um impacto positivo mais relevante do que no desempenho geral.

Para a segmentação, o M3-Bil se destaca, mas com valores de desempenho muito próximos ao M2.

5.3.3 Cristais xenomórficos

A Tabela 6 e 7 apresentam respectivamente os indicadores de performance de classificação e segmentação de cada modelo para os cristais xenomórficos.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
Acurácia	0,24	0,33	0,18	0,20	0,21	0,24	0,38	0,21
Precisão	0,33	0,43	0,36	0,25	0,28	0,43	0,40	0,33
Recall	0,48	0,59	0,27	0,50	0,47	0,36	0,39	0,37
F1-Score	0,39	0,50	0,31	0,33	0,35	0,39	0,24	0,35

Tabela 6 – Indicadores de desempenho para classificação de cristais xenomórficos.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
IoU	0,64	0,74	0,69	0,54	0,64	0,76	0,68	0,63
Coeficiente de Dice	0,74	0,83	0,77	0,66	0,75	0,85	0,77	0,73

Tabela 7 – Indicadores de desempenho para segmentação de cristais xenomórficos.

Na classificação de cristais xenomórficos, o Modelo 2 (M2) se destaca em termos de desempenho. Com uma acurácia de 0,33 e um recall de 0,59, o Modelo 2 é o mais eficaz na identificação dessa classe.

Os modelos com filtros de pré-processamento apresentam desempenhos variados. O Modelo 3 com filtro Sobel (M3-Sob) mostra o pior desempenho com uma acurácia de 0,18 e um F1-score de 0,31. Em contrapartida, o Modelo 3 com filtro bilateral (M3-Bil) mostra um desempenho intermediário, abaixo do desempenho do Modelo 2.

Para a segmentação de cristais xenomórficos, o M3-Bil se destaca com o melhor desempenho, alcançando um IoU de 0,76 e um coeficiente de Dice de 0,85. Esses resultados indicam que o M3-Bil oferece a segmentação mais precisa e eficiente para os cristais. O Modelo 2 (M2) também apresenta um bom desempenho na segmentação, com um IoU de 0,74 e um coeficiente de Dice de 0,83. Embora ligeiramente inferior ao M3-Bil, o M2 ainda fornece resultados consistentes e satisfatórios.

5.3.4 Cristais idiomórficos

A Tabela 8 e 9 apresentam respectivamente os indicadores de performance de classificação e segmentação de cada modelo para os cristais idiomórficos, a classe minoritária.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
Acurácia	0,10	0,10	0,12	0,10	0,10	0,12	0,09	0,10
Precisão	0,11	0,13	0,15	0,11	0,13	0,11	0,09	0,13
Recall	0,30	0,29	0,38	0,20	0,11	0,27	0,22	0,27
F1-Score	0,18	0,18	0,21	0,14	0,12	0,16	0,13	0,18

Tabela 8 – Indicadores de desempenho para classificação de cristais idiomórficos.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
IoU	0,84	0,85	0,83	0,83	0,87	0,81	0,68	0,87
Coeficiente de Dice	0,81	0,91	0,88	0,90	0,93	0,88	0,77	0,83

Tabela 9 – Indicadores de desempenho para segmentação de cristais idiomórficos.

Na análise de classificação dos cristais idiomórficos, os modelos apresentam no geral baixa acurácia e precisão, com os melhores resultados obtidos pelos modelos com filtro Sobel e pelo Modelo 2. Estes modelos superam os demais em termos de precisão e recall. O modelo Sobel, em particular, mostra um recall mais alto, indicando que ele é mais eficiente na classificação dos casos positivos, embora a precisão ainda não seja ideal.

Para segmentação de cristais idiomórficos, excetto o M3-Pre, todos os modelos mostram um bom desempenho no geral, com o modelo M3-Can se destacando. Isso mostra que apesar de não classificar bem, os cristais idiomórficos o modelo ainda consegue segmentá-los de forma eficiente quando bem identificados.

5.3.5 Tempo de processamento

A Tabela 10 apresenta o tempo de processamento em minutos para cada um dos modelos utilizados na segmentação das imagens de clínquer. O tempo de processamento é uma métrica que reflete a eficiência computacional dos modelos.

Métrica	M1	M2	M3-Sob	M3-Lap	M3-Can	M3-Bil	M3-Pre	M3-Rob
Tempo	11:38	7:44	8:00	8:08	7:45	7:40	7:40	7:45

Tabela 10 – Tempo de processamento em minutos dos modelos.

O modelo 1 (M1), treinado com o banco de dados original, tem o maior tempo de processamento, com 11 minutos e 38 segundos. Este tempo maior pode ser atribuído à falta de tratamento do desbalanceamento e de pré-processamento que poderiam reduzir o custo computacional.

O modelo com tratamento no balanceamento (M2) apresenta uma redução significativa no tempo de processamento, caindo para 7 minutos e 44 segundos. Isso sugere que o tratamento realizado otimiza a aprendizagem do modelo.

Entre os modelos M3, que aplicam diferentes técnicas de pré-processamento, os tempos de processamento variam. O filtro bilateral (M3-Bil) e o filtro Prewitt (M3-Pre) apresentam os

tempos menores, ambos com 7 minutos e 40 segundos. O filtro Canny (M3-Can) também mostra um tempo de processamento competitivo, de 7 minutos e 45 segundos, praticamente igual ao M2, reforçando que seu uso não compromete significativamente a eficiência do processamento dos dados.

Em resumo, todos os modelos que aplicam técnicas de pré-processamento apresentam tempos de processamento mais curtos em comparação ao modelo original (M1). Isso indica que, além de potencialmente melhorar a segmentação e classificação, o pré-processamento também pode melhorar o tempo computacional.

5.3.6 Máscaras preditas

Com os modelos treinados, podemos verificar como ficaram as máscaras preditas para cada cristal classificado. Exemplos dessas máscaras podem ser observadas nas Figuras 27, 28, 29, 30, 31, 32, 33 e 34.

Vemos que para todos os modelos temos boas segmentações dos cristais o que é esperado dado que os indicadores de performance de segmentação são satisfatórios para todos os modelos.

O M1 que treina o modelo sem tratamento para balanceamento ou filtro de pré-processamento, deixou um número maior de cristais sem marcação de máscaras, especialmente aqueles localizados nas bordas ou em áreas de menor contraste. Isso indica que, embora o M1 tenha produzido boas segmentações, sua capacidade de identificar todos os cristais não foi ideal.

Já o M2 apresentou uma melhora significativa nesse aspecto. Visualmente, as máscaras geradas pelo M2 foram capazes de identificar quase todos os cristais, exceto aqueles situados nas bordas da imagem, assim como no M1. Essa melhoria em relação ao M1 pode ser atribuída ao tratamento de balanceamento de classes, que ajudou o modelo a focar também nas classes menos representadas, como os cristais menores ou de formas mais irregulares.

Entre os modelos que aplicam técnicas de pré-processamento, o filtro Canny (M3-Can) se destaca por sua capacidade de detectar até mesmo os cristais localizados mais próximos das bordas das imagens. Certamente isso se deve ao fato do filtro Canny ter a capacidade de realçar apenas as bordas mais importantes das imagens, o que permitiu ao modelo identificar e segmentar menores cristais ou de contorno menos definido, que ficaram invisíveis para os outros modelos. Comentários parecidos valem também para o filtro bilateral.

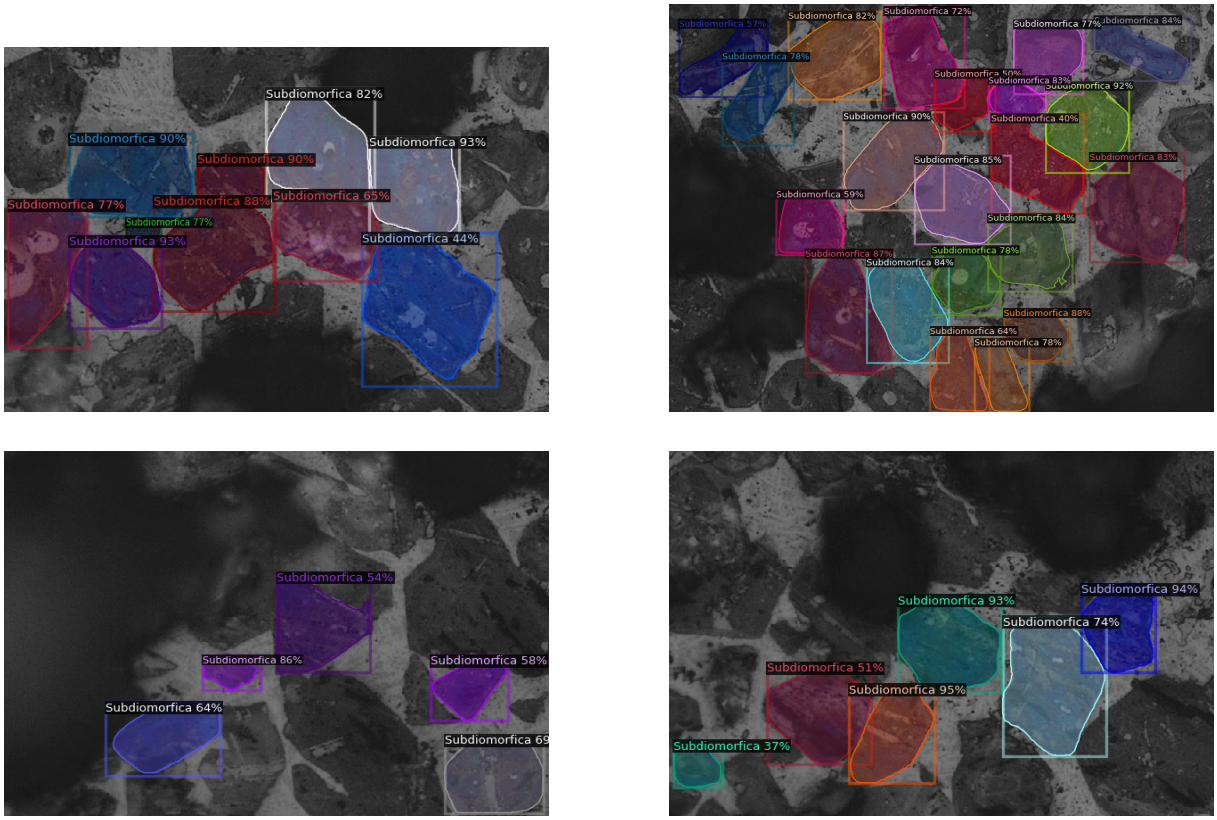


Figura 27 – Exemplos de máscaras preditas para o M1.

Figura 28 – Exemplos de máscaras preditas para o M2.

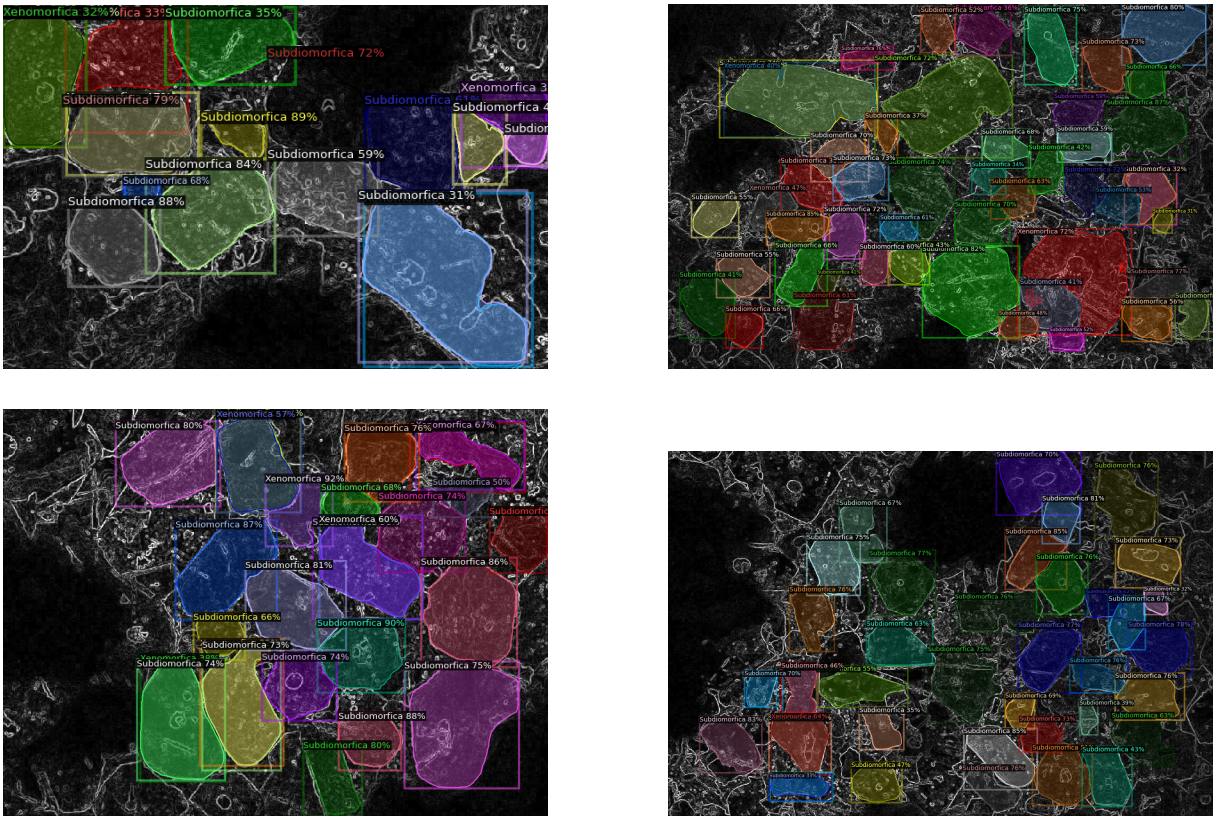


Figura 29 – Exemplos de máscaras preditas para o M3-Sob.

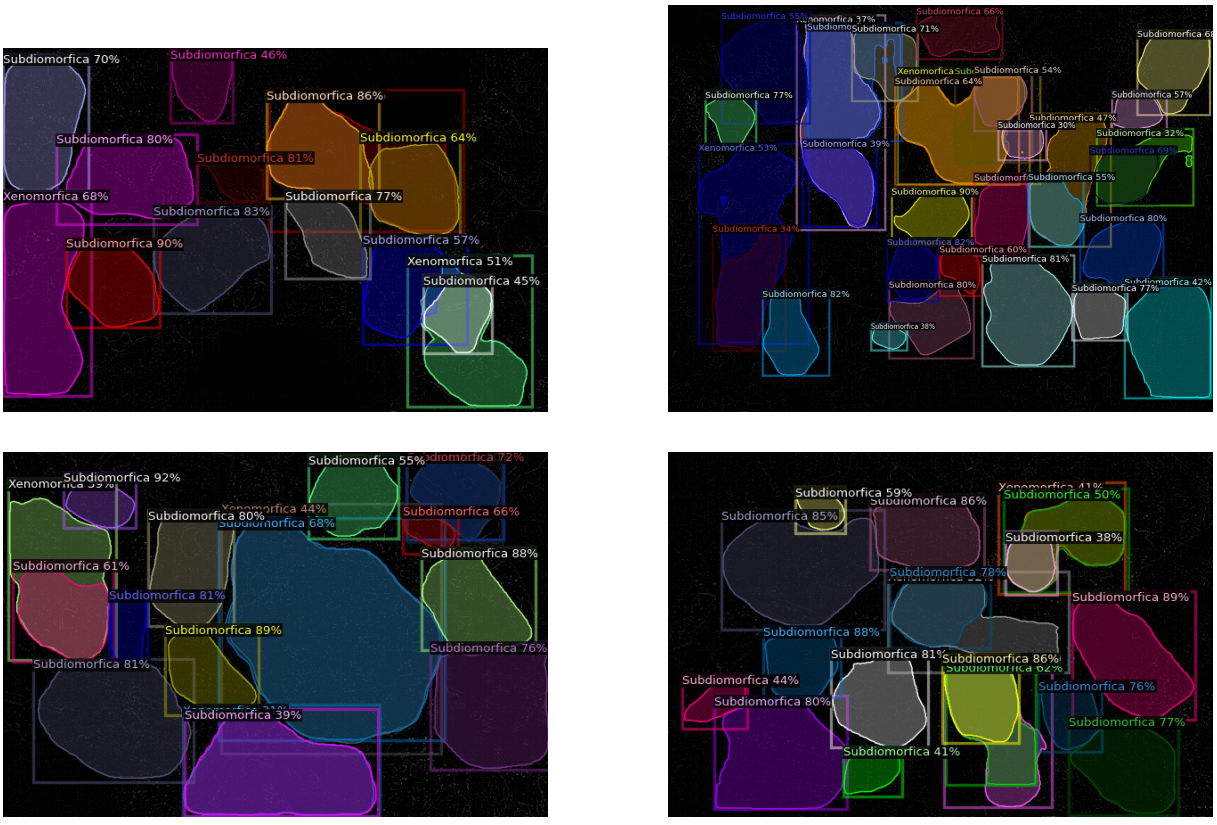


Figura 30 – Exemplos de máscaras preditas para o M3-Lap.

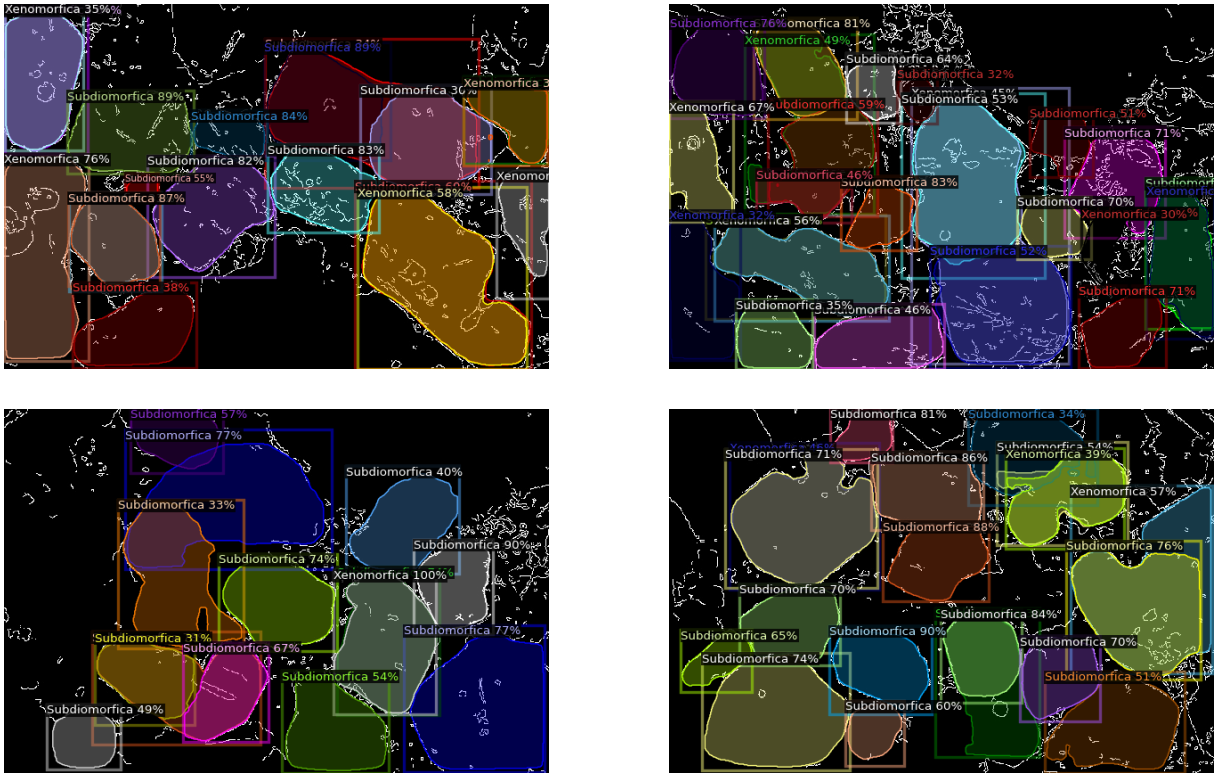


Figura 31 – Exemplos de máscaras preditas para o M3-Can.

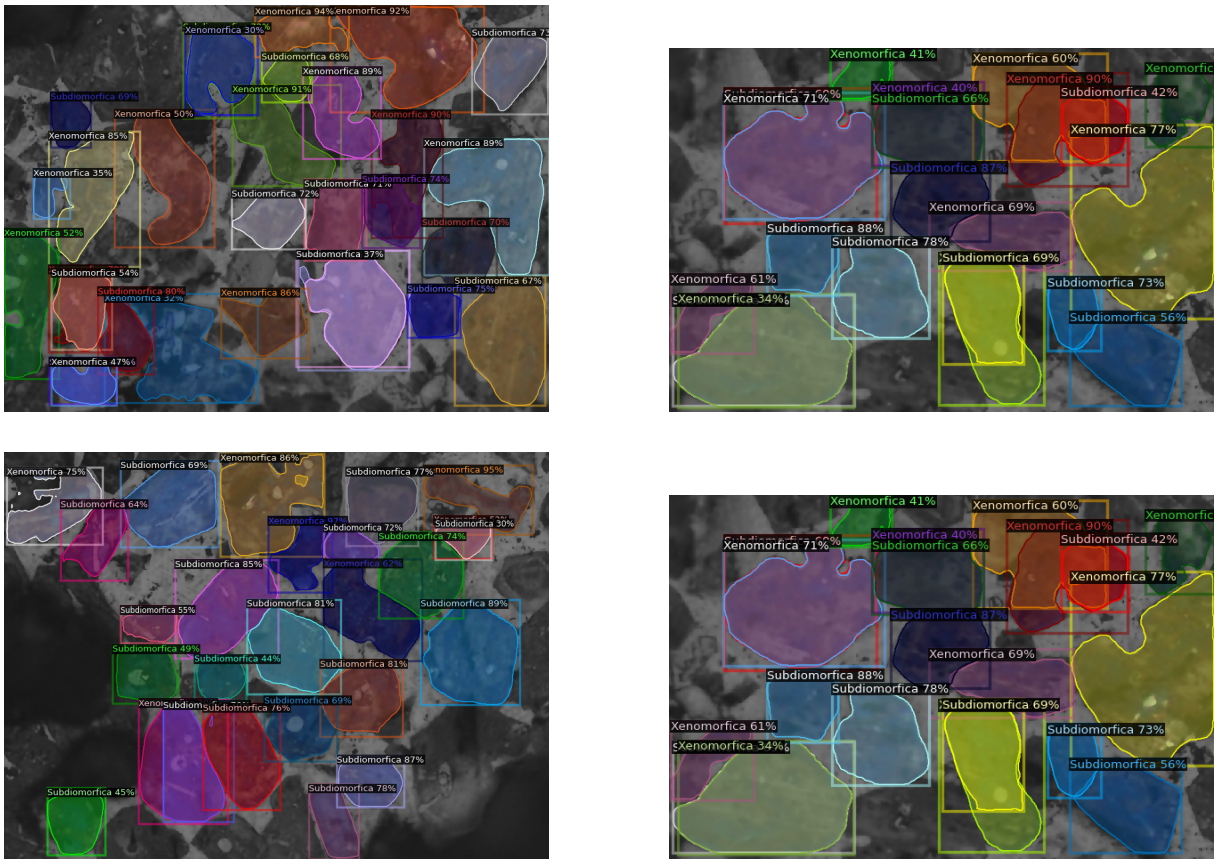


Figura 32 – Exemplos de máscaras preditas para o M3-Bil.

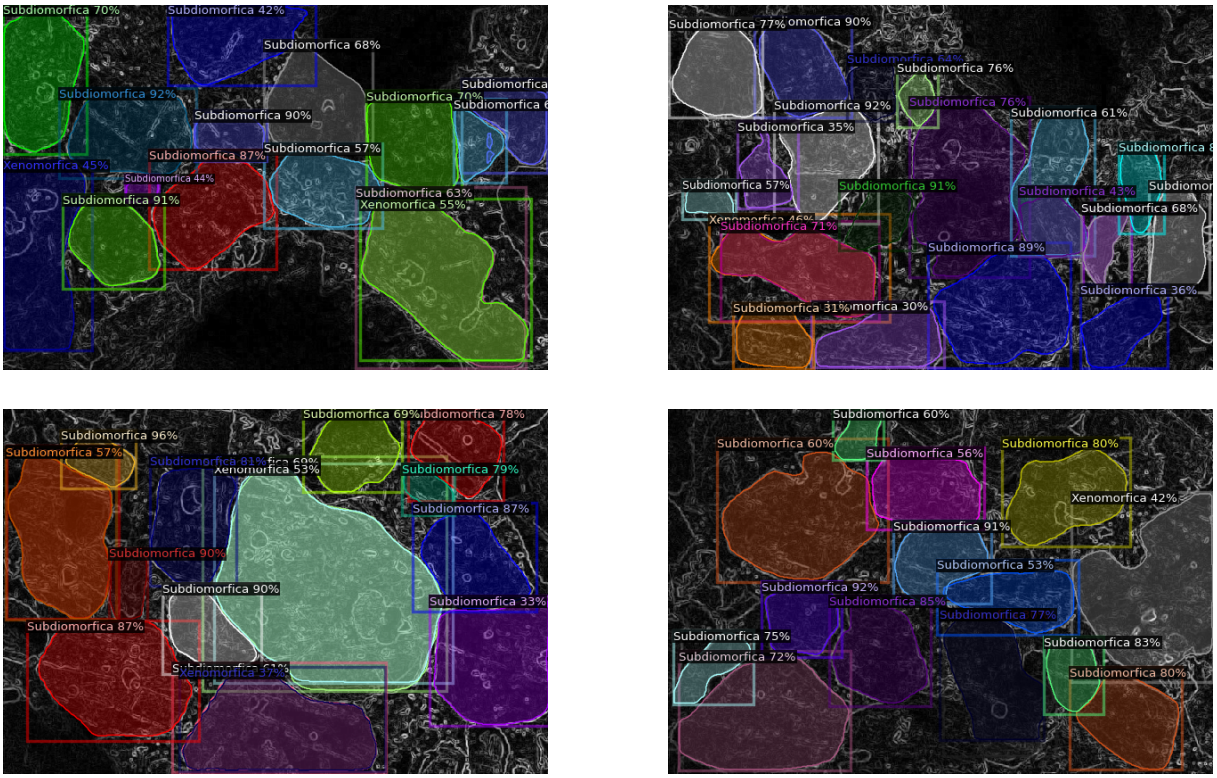


Figura 33 – Exemplos de máscaras preditas para o M3-Pre.

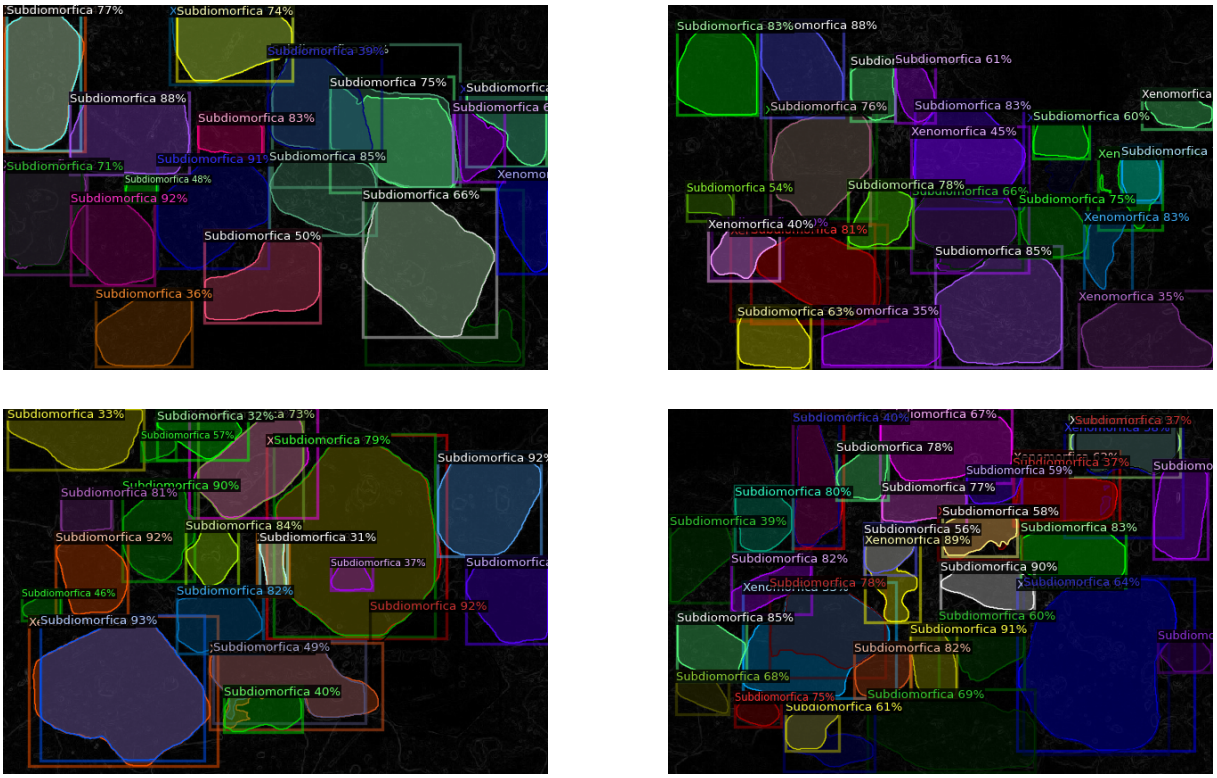


Figura 34 – Exemplos de máscaras preditas para o M3-Rob.

CAPÍTULO

6

CONCLUSÃO E ESTUDOS FUTUROS

Nesse trabalho, apresentamos uma breve descrição das redes neurais, redes neurais convolucionais e suas extensões, e como se dá o processo geral de ajuste desses modelos. Diferentes filtros de pré-processamento de imagens, para melhorar a qualidade das figuras através da suavização de ruídos, reforço de bordas, etc também são introduzidos. A principal motivação é aplicar essas metodologias para a segmentação e classificação de cristais de alita (C_3S) em imagens microscópicas de clínquer, cujas características impactam diretamente na qualidade do cimento Portland produzido e comparar seus resultados. Futuramente e mostrando bons resultados de segmentação e classificação, esses métodos podem ser utilizados para automatizar o processo de controle de qualidade e regulação da produção de cimento.

As CNNs possuem uma infinidade de estruturas e extensões o que torna a mesma suficientemente flexível para captarmos as informações contidas nas imagens, classificar e segmentá-las.

Vimos que apesar da extensão *Mask R-CNN* sem adição de nenhum tratamento nos dados (balanceamento de classes muito desequilibradas e aplicação de filtros de pré-processamento nas imagens) não obter resultados satisfatórios, a mesma possui uma estrutura robusta o suficiente para modelar imagens. Devido a isso, apenas a inclusão de tratamento do desbalanceamento das classes já se mostrou uma forma poderosa de melhorar os resultados obtidos (o Modelo 2 - M2).

Outro ponto relevante foi analisar o impacto de diferentes filtros de pré-processamento na capacidade preditiva e de segmentação dos modelos. Uma análise abrangente dos indicadores de desempenho para a segmentação e classificação de cristais mostrou a eficácia dessas diferentes técnicas e modelos aplicados, mostrando que a escolha da técnica de pré-processamento e o tratamento do desbalanceamento de classes são fundamentais para a melhoria do desempenho dos modelos.

Quando analisamos os indicadores de classificação geral nas imagens analisadas, o filtro Canny e o bilateral apresentaram os melhores valores de recall e F1-Score, o que indica que eles,

em particular, foram mais eficazes em identificar corretamente os cristais. Da mesma forma, ao analisarmos os indicadores de segmentação geral, o filtro Canny e bilateral também apresentaram relativamente os melhores resultados de IoU e coeficiente de Dice. Esses indicadores são fundamentais para avaliar a precisão da segmentação, mostrando se os modelos conseguem identificar e marcar com maior precisão as áreas de interesse dentro das imagens.

Embora o Modelo 2 tenha demonstrado um desempenho sólido, com bons resultados de precisão e recall, a inclusão de filtros adequados de pré-processamento foi capaz de oferecer um desempenho mais equilibrado e eficiente em múltiplas métricas e situações. Isso destaca a importância de técnicas de pré-processamento adequadas e o tratamento de desbalanceamento na maximização da eficácia dos modelos.

Para cristais xenomórficos, em particular, a análise revelou que o modelo com filtro bilateral (M3-Bil) foi o mais eficaz na segmentação, apresentando os melhores resultados em termos de IoU e coeficiente de Dice. Já para os cristais idiomórficos, classe minoritária, a filtro Sobel apresentou melhores resultados de classificação. Isso nos indica que o bom ou ruim desempenho de um filtro depende muito dos dados e características das imagens que estão sendo segmentadas e classificadas e a escolha de qual utilizar depende muito dos objetivos e especificidades das imagens envolvidas no estudo.

Sobre o filtro Canny, em especial, ele apresentou um desempenho relativamente superior provavelmente a sua capacidade de detectar bordas de forma mais precisa e eficiente em comparação a outros filtros. Diferentemente de filtros como Sobel ou Laplaciano, o Canny utiliza vários passos adicionais, que inclui suavização de imagem para reduzir ruído, seguida por detecção de gradientes e supressão de não máximos, o que resulta em bordas finas e bem definidas. Além disso, o Canny filtra as bordas relevantes. Essa combinação torna o Canny altamente eficaz para capturar detalhes sutis, o que pode ser crucial na segmentação de cristais em microscopia.

Apesar de retornar resultados melhores, tanto o tratamento do desbalanceamento das classes quanto o pré-processamento das imagens não conseguiram resolver o impacto negativo que a quantidade reduzida de cristais idiomórficos geram na sua classificação e no desempenho geral dos métodos. Isso se deva talvez à qualidade das marcações e rotulações das imagens utilizadas para treinamento, validação e teste dos modelos, que foi realizada manualmente (de certa maneira) e uma das primeiras experiências do grupo de pesquisa que disponibilizou o banco de dados com esse tipo de atividade. Provavelmente, a aplicação das metodologias aqui estudadas em imagens mais bem demarcadas e ricas vão oferecer resultados mais precisos.

Como estudos futuros, também ressaltamos que pode ser estudado o impacto que outras alterações na estrutura da *Mask R-CNN* causam nos resultados.

REFERÊNCIAS

AGGARWAL, C. C. **Neural networks and deep learning: a textbook**. [S.l.]: Springer, 2018. Citado na página 42.

AMARAL, M. V. F.; SOUZA, A. L. d.; SOARES, V. P.; SOARES, C. P. B.; LEITE, H. G.; MARTINS, S. V.; FILHO, E. I. F.; LANA, J. M. d. Avaliação e compação de métodos de classificação de imagens de satélites para o mapeamento de estádios de sucessão florestal. **Revista Árvore**, SciELO Brasil, v. 33, p. 575–582, 2009. Citado na página 19.

ANDRADE, M. C. **Um método topológico de segmentação de imagens por atributo**. Tese (Ciência da Computação) — Universidade Federal de Minas Gerais, Belo Horizonte, 1998. Citado na página 28.

BERETKA, J.; BROWN, P.; TAYLOR, H. F. W. Raw materials and process design for the manufacture of cement clinker. **Cement and Concrete Research**, Elsevier, v. 23, n. 6, p. 1180–1185, 1993. Citado na página 20.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006. Citado nas páginas 29 e 44.

BRAGANÇA, C. P. **Classificação de acromegalia em imagens tomográficas computadorizadas de vértebras (L1-L5) com uso de ensemble de classificadores binários**. Dissertação (Informática) — Universidade de Brasília, Brasília, 2018. Citado na página 21.

BREIMAN, L. Random forests. **Machine Learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 29.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. Semi-supervised learning. **IEEE Transactions on Neural Networks**, IEEE, v. 20, n. 3, p. 542–544, 2009. Citado na página 29.

CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 40, n. 4, p. 834–848, 2017. Citado na página 19.

COATES, A.; NG, A. Y.; LEE, H. An analysis of single-layer networks in unsupervised feature learning. **Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)**, JMLR, v. 15, p. 215–223, 2011. Citado na página 29.

COLAÇO, R. S. C. **Identificação semiautomática de patologia em imagens de raio X do tórax com técnicas de inteligência artificial**. Tese (Doutorado) — Universidade NOVA de Lisboa, 2022. Citado na página 41.

COSTA, R. R. V. **Comparação de desempenho entre algoritmos de reconhecimento de objetos em imagem**. Dissertação (Graduação) — Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019. Citado nas páginas 38 e 39.

DÉJA, J.; MROZ, R.; KURDOWSKI, W. Investigations of the clinkerization process of industrial cement clinker in relation to the influence of raw materials and firing conditions. **Cement and Concrete Research**, Elsevier, v. 32, n. 9, p. 1421–1427, 2002. Citado na página 20.

Facebook AI Research (FAIR). **Facebook AI Research**. 2024. <https://ai.facebook.com/>. Accessed: 2024-07-14. Citado na página 52.

FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma introdução sucinta à teoria dos grafos. 2011. **http://www. ime. usp. br/˜ pf/teoriadosgrafos/>. Citado**, v. 2, p. 13, 2018. Citado nas páginas 30 e 31.

FERREIRA, A. d. S. **Redes neurais convolucionais profundas na detecçao de plantas daninhas em lavoura de soja**. Dissertação (Ciência da computação) — Universidade Federal de Mato Grosso do Sul, Campo Grande, MS, 2017. Citado nas páginas 35 e 36.

FERREIRA, R. M.; MARCACINI, R. M. Evaluation of classifiers for image segmentation: Applications for eucalypt forest inventory. **arXiv preprint arXiv:1703.09436**, v. 3, p. 15, 2017. Citado na página 21.

FLATT, R. J.; SCHERER, G. W.; BULLARD, J. W. Cement chemistry, curing conditions, and clinker development: An overview. **Journal of the American Ceramic Society**, Wiley, v. 90, n. 4, p. 1231–1247, 2007. Citado na página 20.

FREITAS, D. C. **Aferição da porosidade do clínquer: uma solução automática**. Dissertação (Graduação) — Universidade Federal do Ceará, Russas, CE, 2022. Citado nas páginas 21 e 24.

GARCÍA-VIDAL, F.; HIDALGO, M.; PUERTAS, F. Clinker microstructure related to alite and belite formation in industrial portland cement production. **Journal of the American Ceramic Society**, Wiley, v. 90, n. 3, p. 889–894, 2007. Citado na página 20.

GIRSHICK, R. Fast R-CNN. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1440–1448. Citado na página 39.

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 580–587. Citado nas páginas 38 e 39.

GIRSHICK, R.; RADOSAVOVIC, I.; GKIOXARI, G.; DOLLáR, P.; HE, K. **Detectron**. 2018. <https://github.com/facebookresearch/Detectron>. Accessed: 2024-07-14. Citado na página 52.

GOBBO, L. d. A. **Os compostos do clínquer Portland: sua caracterização por difração de raios-X e quantificação por refinamento de Rietveld.** Dissertação (Recursos Minerais e Hidrogeologia) — Universidade de São Paulo, 2003. Citado nas páginas 21 e 24.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. [S.l.]: Prentice Hall, 2008. Citado nas páginas 48, 49, 50 e 51.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. Citado nas páginas 29, 33, 35, 42 e 46.

GOODFELLOW, I. J.; ERHAN, D.; CARRIER, P. L.; AL. et. Challenges in representation learning: A report on three machine learning contests. **Neural Networks**, v. 64, p. 59–63, 2015. Citado na página 47.

GUIMARÃES, A. M.; MATHIAS, I. M.; DIAS, A. H.; FERRARI, J. W.; CARLOS, R. d. O. Módulo de validação cruzada para treinamento de redes neurais artificiais com algoritmos backpropagation e resilient propagation. **Publicatio UEPG: Ciências Exatas e da Terra, Agrárias e Engenharias**, v. 14, n. 01, 2008. Citado na página 34.

GUPTA, S. N.; SINGHAL, S. A survey of image enhancement techniques. **Journal of Advanced Research in Computer Science and Software Engineering**, v. 7, n. 2, p. 15–22, 2017. Citado na página 52.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2. ed. New York, USA: Springer, 2009. ISBN 9780387848570. Citado na página 29.

____. Unsupervised learning: Foundations of data mining and machine learning. **The Elements of Statistical Learning**, Springer, p. 485–585, 2009. Citado na página 29.

HE, K.; GKIOXARI, G.; DOLLáR, P.; GIRSHICK, R. Mask r-cnn. In: **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2017. p. 2961–2969. Citado na página 19.

HE, K.; GKIOXARI, G.; DOLLáR, P.; GIRSHICK, R. B. Mask R-CNN. In: **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. [s.n.], 2017. p. 2961–2969. Disponível em: <https://arxiv.org/abs/1703.06870>. Citado nas páginas 39 e 64.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778. Citado na página 19.

HILLS, F. B.; ROBERTS, A. F. Clinker formation in coal-fired furnaces: Influence of fuel ash composition on clinker formation. **Journal of the Institute of Fuel**, v. 52, n. 1, p. 68–72, 1979. Citado na página 20.

IA, G. **Deep learning/Ian Goodfellow, Yoshua Bengio and Aaron Courville**. [S.l.]: Cambridge, Massachusetts: The MIT Press, 2016. Citado na página 41.

IZBICKI, R.; SANTOS, T. M. dos. **Aprendizado de máquina: uma abordagem estatística**. [S.l.]: Rafael Izbicki, 2020. Citado nas páginas 19, 21, 29, 31, 32, 33 e 34.

JURASZEK, G. D. **Reconhecimento de produtos por imagem utilizando palavras visuais e redes neurais convolucionais**. Dissertação (Computação Aplicada) — Universidade do Estado de Santa Catarina, Joinville, 2014. Citado na página 20.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2012. v. 25, p. 1097–1105. Citado na página 19.

LEA, F. M. **The Chemistry of Cement and Concrete**. 4. ed. Oxford, UK: Elsevier, 1998. ISBN 9780340589963. Citado na página 20.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Citado nas páginas 20 e 36.

LEITE, D. R. A.; MORAES, R. M. de; LOPES, L. W. Método de aprendizagem de máquina para classificação da intensidade do desvio vocal utilizando random forest. **Journal of Health Informatics**, v. 12, p. 575–582, 2020. Citado na página 21.

LIMA, H. **Monitoramento da condição de operação de fornos rotativos em uma planta de cimento através da análise microscópica de clínqueres.** Dissertação (Graduação) — Universidade Federal do Ceará, Russas, 2022. Citado nas páginas 24 e 25.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 3431–3440. Citado na página 45.

MIRJALILI, S.; DONG, J. S.; SADIQ, A. S.; FARIS, H. Genetic algorithm: Theory, literature review, and application in image reconstruction. **Nature-Inspired Optimizers: Theories, Literature Reviews and Applications**, Springer, p. 69–85, 2020. Citado na página 25.

MITCHELL, T. M. The need for biases in learning generalizations. **Department of Computer Science, Laboratory for Computer Science Research, Rutgers University**, 1980. Technical report. Citado na página 29.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. Cambridge, MA, USA: MIT Press, 2012. ISBN 9780262018029. Citado na página 29.

NEGRI, R. G.; SANT'ANNA, S. J. S.; DUTRA, L. V. Aplicação de modelos de aprendizado semissupervisionado na classificação de imagens de sensoriamento remoto. **Revista de Informática Teórica e Aplicada**, v. 20, n. 2, p. 32–55, 2013. Citado nas páginas 21 e 40.

PADILHA, R. M. S. **Automação da identificação de C3S na microscopia do clínquer**. Dissertação (Graduação) — Universidade Federal do Ceará, Russas, 2022. Citado nas páginas 21, 24, 25, 40 e 56.

PAL, N. R.; PAL, S. K. A review on image segmentation techniques. **Pattern Recognition**, Pergamon, v. 26, n. 9, p. 1277–1294, 1993. Citado na página 45.

PEREIRA, A. L. G. **Desenvolvimento de metodologias para o reconhecimento de estruturas quiescentes em mapas solares observados pelo Telescópio Solar para Ondas Submilimétricas (SST)**. Tese (Ciências e Aplicações Geoespaciais) — Universidade Presbiteriana Mackenzie, São Paulo, 2018. Citado na página 19.

POYNTON, C. Gamma correction in digital imaging. **Byte**, v. 18, n. 11, p. 265–276, 1993. Citado na página 51.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster R-CNN: Towards real-time object detection with region proposal networks. **Advances in Neural Information Processing Systems**, v. 28, 2015. Disponível em: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>. Citado na página 39.

RESMINI, R.; CONCI, A.; BORCHARTT, T. B.; LIMA, R. d. C. F. de; MONTENEGRO, A. A.; PANTALEÃO, C. A. Diagnóstico precoce de doenças mamárias usando imagens térmicas e aprendizado de máquina. **Revista Brasileira de Contabilidade e Gestão**, v. 1, n. 1, p. 55–67, 2012. Citado nas páginas 19 e 21.

RIBEIRO, A. E. **Métodos de análise de microscopia do clínquer usando morfologia matemática**. Dissertação (Graduação) — Universidade Presidente Antônio Carlos, Barbacena, 2003. Citado nas páginas 21 e 24.

ROBOFLOW. **Roboflow: Annotate, Organize, and Train Computer Vision Models**. 2024. Accessed: 2024-07-23. Disponível em: <https://roboflow.com/>. Citado na página 56.

ROJAS, R. **Neural networks: a systematic introduction**. [S.l.]: Springer Science & Business Media, 2013. Citado na página 34.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)**. [S.l.], 2015. p. 234–241. Citado na página 19.

ROSA, R. d. P. **Método de classificação de pragas por meio de rede neural convolucional profunda**. Dissertação (Ciência da Computação) — Universidade Estadual de Ponta Grossa, Ponta Grossa, 2018. Citado na página 20.

SANTOS, M. K.; JÚNIOR, J. R. F.; WADA, D. T.; TENÓRIO, A. P. M.; NOGUEIRA-BARBOSA, M. H.; MARQUES, P. M. d. A. Inteligência artificial, aprendizado de máquina, diagnóstico auxiliado por computador e radiômica: avanços da imagem rumo à medicina de precisão. **Radiologia Brasileira**, SciELO Brasil, v. 52, p. 387–396, 2019. Citado na página 21.

SCHÖLKOPF, B.; SMOLA, A. J. Learning with kernels: Support vector machines, regularization, optimization, and beyond. In: **Adaptive Computation and Machine Learning**. Cambridge, MA, USA: MIT Press, 2001. ISBN 9780262194754. Citado na página 29.

SHIBA, M. H.; SANTOS, R. L.; QUINTANILHA, J. A.; KIM, H. Y. Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. **Simpósio Brasileiro de Sensoriamento Remoto**, v. 12, p. 4319–4326, 2005. Citado na página 21.

SILVA, B. R. da; BREVE, F. A. Aprendizado de máquina aplicado à segmentação interativa de imagens. **Apresentação no III Workshop do Programa de Pós-Graduação em Ciência da Computação**, 2013. Citado na página 21.

SKANSI, S. **Introduction to Deep Learning: from logical calculus to artificial intelligence**. [S.l.]: Springer, 2018. Citado nas páginas 21, 36, 37 e 38.

SMOLKA, B.; ARAKAWA, K. A survey on impulse noise detection and removal techniques. **Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 32, n. 2, p. 168–177, 2002. Citado na página 49.

SOUSA, B. F. S.; TEIXEIRA, A. d. S.; SILVA, F. d. A. T. F. d.; ANDRADE, E. M. d.; BRAGA, A. P. d. S. Avaliação de classificadores baseados em aprendizado de máquina para a classificação do uso e cobertura da terra no bioma caatinga. **Revista Brasileira de Cartografia**, v. 52, p. 387–396, 2010. Citado na página 21.

VAPNIK, V. The nature of statistical learning theory. **Springer Science & Business Media**, 1999. Citado na página 29.

VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: **Proceedings of the XXIX Conference on Graphics, Patterns and Images**. [S.l.: s.n.], 2016. v. 1, n. 4. Citado na página 20.

VOGADO, L. H.; VERAS, R. M.; ARAUJO, F. H.; SILVA, R. R.; AIRES, K. R. Rede neural convolucional para o diagnóstico de leucemia. In: SBC. **Anais do XIX Simpósio Brasileiro de Computação Aplicada à Saúde**. [S.l.], 2019. p. 46–57. Citado na página 20.

WU, Y.; KIRILLOV, A.; MASSA, F.; LO, W.-Y.; GIRSHICK, R. **Detectron2**. 2019. <https://github.com/facebookresearch/detectron2>. Accessed: 2024-07-14. Citado na página 52.

YANG, L.; YANG, Y.; WANG, S.; ZHENG, N. Transfer learning for image classification with sparse prototype learning. **IEEE Transactions on Image Processing**, v. 22, n. 8, p. 3285–3298, 2013. Citado nas páginas 45 e 47.

ZAKI, M.; SHARMA, S.; GURJAR, S. K.; GOYAL, R.; JAYADEVA; KRISHNAN, N. A. Cementron: Machine learning the alite and belite phases in cement clinker from optical images. **Construction and Building Materials**, v. 397, p. 132425, 2023. ISSN 0950-0618. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950061823021414>. Citado na página 21.

ZHAO, J.; WHITE, M.; DRUMMOND, M. A survey on transfer learning. **IEEE Transactions on Neural Networks and Learning Systems**, v. 30, n. 9, p. 2894–2910, 2019. Citado na página 46.

ZHU, X. **Semi-Supervised Learning Literature Survey**. Madison, WI, USA: Computer Sciences Technical Report, University of Wisconsin-Madison, 2005. Citado na página 29.

APÊNDICE

A

CÓDIGOS

A.1 Configurações iniciais

Esta parte do código é a base para todos os modelos ajustados.

```
#Esta função verifica se um diretório especificado por path existe.

def testing_creating_directory(path):
    if os.path.exists(path)==False:
       os.mkdir(path)

#import os: Importa o módulo os, que fornece uma maneira de usar funcionalidades
#dependentes do sistema operacional,como manipulação de arquivos e diretórios.
#from google.colab import drive: Importa o módulo drive do Google Colab,
#que permite montar o Google Drive no ambiente de Colab.
import os
from google.colab import drive

#drive.mount('/content/drive'): Monta o Google Drive no diretório /content/drive,
#permitindo o acesso aos arquivos do Google Drive a partir do ambiente Colab.
drive.mount('/content/drive')

#Objetivo: Define o caminho do diretório onde as operações subsequentes
#serão realizadas. Neste caso, é o diretório  /content/drive/MyDrive/pytorch/
#dentro do Google Drive. path = '/content/drive/MyDrive/pytorch/'

```

```
22 #Objetivo: Chama a função testing_creating_directory para verificar
23 # se o diretório especificado por path existe. Se não existir,
24 #o diretório será criado.
25 testing_creating_directory(path)
26
27 #Objetivo: Altera o diretório de trabalho atual
28 #para o diretório especificado por path.
29 #Isso significa que qualquer operação subsequente
30 #de leitura/escrita de arquivos será
31 #realizada dentro deste diretório.
32 os.chdir(path)
33
34
35 #Instala o detectron
36 !python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'
37
38 # INSTALA AS BOBLIOTECAS NECESSÁRIAS
39 import os
40 import cv2
41
42 from datetime import datetime
43 from google.colab.patches import cv2_imshow
44
45 # DATA SET PREPARATION AND LOADING
46 from detectron2.data.datasets import register_coco_instances
47 from detectron2.data import DatasetCatalog, MetadataCatalog
48
49 # VISUALIZATION
50 from detectron2.utils.visualizer import Visualizer
51 from detectron2.utils.visualizer import ColorMode
52
53 # CONFIGURATION
54 from detectron2 import model_zoo
55 from detectron2.config import get_cfg
56
57 # EVALUATION
58 from detectron2.engine import DefaultPredictor
59
60 # TRAINING
```

```
from detectron2.engine import DefaultTrainer

#PUXA AS BASES DO ROBOFLOW

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="")
project = rf.workspace("medusas").project("alitas-tcc_h")
dataset = project.version(11).download("coco")

```

A.2 Tratamento do conjunto de dados do Modelo 1

O tratamento do conjunto de dados é parecido a todos os modelos, porém com a inclusão de cada filtro para o Modelo 3.

```

#TRATA O NOME DO CONJUNTO DE DADOS
DATA_SET_NAME = dataset.name.replace(" ", "-")
ANNOTATIONS_FILE_NAME = "_annotations.coco.json"

# TRAIN SET
TRAIN_DATA_SET_NAME = f"{DATA_SET_NAME}-train"
TRAIN_DATA_SET_IMAGES_DIR_PATH = os.path.join(dataset.location, "train")
TRAIN_DATA_SET_ANN_FILE_PATH = os.path.join(dataset.location, "train",
ANNOTATIONS_FILE_NAME)

register_coco_instances(
    name=TRAIN_DATA_SET_NAME,
    metadata={},
    json_file=TRAIN_DATA_SET_ANN_FILE_PATH,
    image_root=TRAIN_DATA_SET_IMAGES_DIR_PATH
)

# TEST SET
TEST_DATA_SET_NAME = f"{DATA_SET_NAME}-test"
TEST_DATA_SET_IMAGES_DIR_PATH = os.path.join(dataset.location, "test")
```

```
TEST_DATA_SET_ANN_FILE_PATH = os.path.join(dataset.location, "test",
ANNOTATIONS_FILE_NAME)

register_coco_instances(
    name=TEST_DATA_SET_NAME,
    metadata={},
    json_file=TEST_DATA_SET_ANN_FILE_PATH,
    image_root=TEST_DATA_SET_IMAGES_DIR_PATH
)

# VALID SET
VALID_DATA_SET_NAME = f"{DATA_SET_NAME}-valid"
VALID_DATA_SET_IMAGES_DIR_PATH = os.path.join(dataset.location, "valid")
VALID_DATA_SET_ANN_FILE_PATH = os.path.join(dataset.location, "valid",
ANNOTATIONS_FILE_NAME)

register_coco_instances(
    name=VALID_DATA_SET_NAME,
    metadata={},
    json_file=VALID_DATA_SET_ANN_FILE_PATH,
    image_root=VALID_DATA_SET_IMAGES_DIR_PATH
)

metadata = MetadataCatalog.get(TRAIN_DATA_SET_NAME)
dataset_train = DatasetCatalog.get(TRAIN_DATA_SET_NAME)

dataset_entry = dataset_train[0]
image = cv2.imread(dataset_entry["file_name"])

visualizer = Visualizer(
    image[:, :, ::-1],
    metadata=metadata,
    scale=0.8,
    instance_mode=ColorMode.IMAGE_BW
)

out = visualizer.draw_dataset_dict(dataset_entry)
cv2_imshow(out.get_image()[:, :, ::-1])
```

A.3 Ajuste do Modelo 1

```
# HYPERPARAMETERS
ARCHITECTURE = "mask_rcnn_R_101_FPN_3x"
CONFIG_FILE_PATH = f"COCO-InstanceSegmentation/{ARCHITECTURE}.yaml"
MAX_ITER = 1000
EVAL_PERIOD = 200
BASE_LR = 0.001
NUM_CLASSES = 3

# OUTPUT DIR
OUTPUT_DIR_PATH = os.path.join(
    DATA_SET_NAME,
    ARCHITECTURE,
    datetime.now().strftime('%Y-%m-%d-%H-%M-%S')
)

os.makedirs(OUTPUT_DIR_PATH, exist_ok=True)

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file(CONFIG_FILE_PATH))
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url(CONFIG_FILE_PATH)
cfg.DATASETS.TRAIN = (TRAIN_DATA_SET_NAME,)
cfg.DATASETS.TEST = (TEST_DATA_SET_NAME,)
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 64
cfg.TEST.EVAL_PERIOD = EVAL_PERIOD
cfg.DATALOADER.NUM_WORKERS = 2
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.INPUT.MASK_FORMAT='bitmask'
cfg.SOLVER.BASE_LR = BASE_LR
cfg.SOLVER.MAX_ITER = MAX_ITER
cfg.MODEL.ROI_HEADS.NUM_CLASSES = NUM_CLASSES
cfg.OUTPUT_DIR = OUTPUT_DIR_PATH

trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

A.4 Cálculo de indicadores de performance

Esse código é análogo para todos os modelos ajustados

```
import time
import numpy as np
import cv2
from detectron2.engine import DefaultTrainer, DefaultPredictor
from detectron2.config import get_cfg
from detectron2.data import build_detection_test_loader, DatasetCatalog,
MetadataCatalog
from detectron2.evaluation import COCOEvaluator, inference_on_dataset
from detectron2 import model_zoo
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

# Função para calcular IoU
def compute_iou(gt_mask, pred_mask):
    intersection = np.logical_and(gt_mask, pred_mask)
    union = np.logical_or(gt_mask, pred_mask)
    iou = np.sum(intersection) / np.sum(union)
    return iou

# Função para calcular Dice Coefficient
def compute_dice(gt_mask, pred_mask):
    intersection = np.logical_and(gt_mask, pred_mask)
    dice = 2. * np.sum(intersection) / (np.sum(gt_mask) + np.sum(pred_mask))
    return dice


# Função para converter segmentações em máscaras binárias
def segmentation_to_mask(segmentation, height, width):
    mask = np.zeros((height, width), dtype=np.uint8)
    for polygon in segmentation:
        polygon = np.array(polygon, dtype=np.int32).reshape((-1, 2))
        cv2.fillPoly(mask, [polygon], 1)
    return mask.astype(bool)

# Avaliação do modelo
evaluator = COCOEvaluator(TEST_DATA_SET_NAME, cfg, False, output_dir="./output/")
```

```
val_loader = build_detection_test_loader(cfg, TEST_DATA_SET_NAME)

start_time = time.time()
inference_on_dataset(trainer.model, val_loader, evaluator)
end_time = time.time()
elapsed_time = end_time - start_time

# Predição e cálculo das métricas
predictor = DefaultPredictor(cfg)
gt_classes = []
pred_classes = []
ious = []
dice_coefficients = []

# Iterar sobre o dataset de teste e fazer predições
for dataset_dict in DatasetCatalog.get(TEST_DATA_SET_NAME):
    img = cv2.imread(dataset_dict["file_name"])
    outputs = predictor(img)
    instances = outputs["instances"]

    # Obter classes verdadeiras para a imagem atual
    gt_classes.extend([ann["category_id"] for ann in dataset_dict["annotations"]])

    # Alinhar número de predições com o número de anotações
    pred_classes.extend(instances.pred_classes.cpu().numpy()
    [:len(dataset_dict["annotations"])])

    # Calcular IoU e Dice Coefficient para cada máscara
    height, width = img.shape[:2]
    for i, ann in enumerate(dataset_dict["annotations"]):
        gt_mask = segmentation_to_mask(ann["segmentation"], height, width)
        pred_mask = instances.pred_masks[i].cpu().numpy()

        if gt_mask.shape != pred_mask.shape:
            pred_mask = cv2.resize(pred_mask.astype(np.uint8), (gt_mask.shape[1],
    gt_mask.shape[0]), interpolation=cv2.INTER_NEAREST).astype(bool)

        iou = compute_iou(gt_mask, pred_mask)
        dice = compute_dice(gt_mask, pred_mask)
```

```
75
76          ious.append(iou)
77          dice_coefficients.append(dice)
78
79  gt_classes = np.array(gt_classes)
80  pred_classes = np.array(pred_classes)
81
82  # Garantir que o comprimento seja igual
83  print(f"Length of gt_classes: {len(gt_classes)}")
84  print(f"Length of pred_classes: {len(pred_classes)}")
85
86  # Calcular métricas usando sklearn
87  accuracy = accuracy_score(gt_classes, pred_classes)
88  precision = precision_score(gt_classes, pred_classes, average='weighted')
89  recall = recall_score(gt_classes, pred_classes, average='weighted')
90  f1 = f1_score(gt_classes, pred_classes, average='weighted')
91
92  # Calcular métricas de segmentação
93  mean_iou = np.mean(ious)
94  mean_dice = np.mean(dice_coefficients)
95
96  # Exibir as métricas e o tempo de processamento
97  print(f"Acurácia: {accuracy:.4f}")
98  print(f"Precisão: {precision:.4f}")
99  print(f"Revocação: {recall:.4f}")
100 print(f"F1-score: {f1:.4f}")
101 print(f"IoU Médio: {mean_iou:.4f}")
102 print(f"Coeficiente de Dice Médio: {mean_dice:.4f}")
103 print(f"Tempo de Processamento: {elapsed_time:.2f} segundos")
```


ICMC
USP
SÃO CARLOS