



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Ensemble de Modelos Híbridos Baseados em Redes Neurais Convolucionais e Máquinas de Vetores de Suporte para Detecção de Barragens em Imagens de Satélite

Diego Antônio Barbosa Cardoso
Matheus Rodrigues Guimarães

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.a Dr.a Roberta Barbosa Oliveira

Brasília
2022

Dedicatória

Dedicamos o presente trabalho aos nossos familiares que estiveram ao nosso lado prestando apoio durante toda esta jornada. Aos nossos amigos por compartilharem momentos de alegrias, conselhos e comemorações. E aos nossos professores por nos ensinar toda a base que usaremos em nossas jornadas profissionais.

Agradecimentos

Eu, Diego, agradeço a minha família por todo amor e apoio recebido ao longo desses anos e por serem a minha inspiração para seguir nessa jornada. Aos meus amigos por sempre darem palavras de conforto, conselhos e estarem ali para auxiliarem em diversos desafios. E por fim agradeço ao Matheus meu parceiro que dividiu comigo todos os desafios e aprendizados deste trabalho e com quem eu pude contar e confiar em todos os momentos.

E eu, Matheus, agradeço aos meus pais pelo apoio e compreensão que contribuíram para que pudesse focar no meu crescimento como estudante e profissional. A minha irmã Graziely, que dividiu comigo os sentimentos de alegria e preocupações da vida acadêmica e pessoal. A minha namorada, e futura esposa, Paloma Rodrigues Rezende que sempre me deu forças para continuar, pelos conselhos durante os momentos mais difíceis dessa jornada e por estar presente em todas as minhas maiores realizações. Aos meus amigos que escutaram minhas dificuldades, as minhas alegrias e que de alguma maneira tornaram a caminhada mais leve. E também, ao meu amigo e parceiro de UnB, Diego, que dividiu comigo todas as dificuldades, aprendizados e descontrações que fizeram parte de nossa jornada.

Agradecemos a nossa orientadora Profa Dra Roberta Barbosa Oliveira, por ser o nosso pilar e nos guiar nesta jornada, prestando o apoio, atenção, dedicação, sempre disposta a ajudar e compartilhando uma parte seu conhecimento conosco. Sempre nos motivando e ajudando a nos tornar versões melhores de nós mesmos.

Por fim a Empresa Júnior de Computação - CJR, por todo o conhecimento adquirido e pelo impacto causado em nossas vidas nos transformando em jovens comprometidos e capazes de mudar o Brasil.

Resumo

As barragens de rejeitos armazenam água e resíduos sólidos que são subprodutos da mineração. No Brasil, há um índice alto de barragens irregulares, incluindo locais até mesmo não catalogados. Estas estruturas apresentam alto potencial de dano socioambiental e um possível rompimento seria capaz de destruir rios e vilarejos. O uso de modelos de Rede Neural Convolutacional, do Inglês *Convolutional Neural Network* (CNN) tradicionais são capazes de realizar a tarefa de detecção de barragens utilizando imagens de satélite. Para melhorar o desempenho destes modelos tradicionais existem técnicas, como por exemplo a utilização de modelos híbridos e o *ensemble* de modelos. O objetivo do presente trabalho é a proposta de um modelo *ensemble* composto por modelos híbridos baseados em CNN e Máquina de Vetores de Suporte, do Inglês *Support Vector Machine* (SVM) para a tarefa de detecção de barragens em imagens de satélite. Para tal, foi utilizada a base de dados *BrazilDAM* que possui 1925 imagens, sendo 769 consideradas como barragem e 1156 consideradas não-barragem. Foi realizado um pré processamento de suas imagens para avaliar o modelo com algumas combinações das 13 diferentes bandas. Foram utilizados diferentes modelos CNN aplicando as técnicas *transfer learning* e *fine tuning* para a extração de características das imagens de entrada. A SVM foi responsável por fazer a predição a partir destas informações extraídas. Por fim, este modelo híbrido foi associado a um modelo *ensemble* heterogêneo utilizando votação majoritária das predições feitas pelos modelos híbridos que realizam a classificação. Como resultados o modelo proposto obteve acurácia de 0,961, precisão 0,950, revocação 0,988, *f1-score* 0,968 e AUC 0,987. Mostrando assim, que a aplicação do modelo proposto traz melhores resultados em comparação com os modelos tradicionais.

Palavras-chave: redes neurais convolucionais, máquina de vetores de suporte, modelo ensemble, modelo híbrido, barragem de rejeitos

Abstract

Tailings dams store water and solid rejects from mining activity. Brazil has a high rate of irregular dams, including places that are not even cataloged. These structures have a high potential for socio-environmental damage and an eventual rupture would be capable of destroying rivers and villages. The traditional Convolutional Neural Network (CNN) models are able to perform dam detection task using satellite imagery. To improve the performance of these traditional models there are techniques, such as hybrid models and model ensemble. The present work proposes an ensemble model composed of hybrid models based on CNN and Support Vector Machines (SVM) for the task of detecting dams in satellite imagery. For this, BrazilDAM database was used, which has 1925 images, which 769 are considered dams and 1156 are considered not dams. A pre-processing of their images was used to evaluate the model with some combinations of the 13 different bands. Some CNN models were used, applying transfer learning and fine-tuning techniques to extract features from the input images. The SVM was responsible for making predictions from this extracted information. Finally, this hybrid model was associated in a heterogeneous ensemble model made by hybrid models that perform the classification using majority voting for predictions. As a result, the proposed model obtained an accuracy of 0.961, precision 0.950, recall 0.988, f1-score 0.968 and AUC 0.987. Thus showing that the application of the proposed model brings better results compared to traditional models.

Keywords: convolutional neural networks, support vector machine, model ensemble, hybrid model, tailings dam

Sumário

1	Introdução	1
2	Fundamentação Teórica	3
2.1	Redes Neurais Convolucionais	3
2.2	Máquinas de Vetores de Suporte	6
2.3	<i>Transfer Learning</i>	7
2.4	<i>Fine Tuning</i>	7
3	Trabalhos Relacionados	9
4	Metodologia	13
4.1	Base de Dados	14
4.2	Pré-processamento das Imagens	14
4.3	Aplicação de <i>Transfer Learning</i> e <i>Fine Tuning</i> nas CNNs	17
4.4	Modelos Híbridos	18
4.5	Modelos <i>Ensemble</i>	18
5	Resultados Experimentais	21
5.1	Distribuição dos Dados	21
5.2	Métricas de Desempenho	21
5.3	Experimento I: CNNs Utilizando <i>Transfer Learning</i>	23
5.4	Experimento II: CNNs Utilizando <i>Transfer Learning</i> e <i>Fine Tuning</i>	24
5.5	Experimento III: Modelo Híbrido	25
5.6	Experimento IV: Espaços de cores	26
5.7	Experimento V: Modelo <i>Ensemble</i> Homogêneo	26
5.8	Experimento VI: Modelo <i>Ensemble</i> Heterogêneo	27
5.9	Ambiente Experimental	28
6	Discussão dos Resultados	29
7	Conclusão e Trabalhos futuros	31

Lista de Figuras

2.1	Arquitetura de uma CNN tradicional (adaptado de Geron [1]).	4
2.2	Exemplo do funcionamento de uma SVM (adaptado de Ippolito [2]).	6
2.3	Exemplo do funcionamento do <i>Fine Tuning</i> (adaptado de Malgonde [3]).	8
4.1	Visão geral da metodologia proposta.	13
4.2	Exemplos de imagens de cada classe da base de dados em <i>RGB</i> [4].	15
4.3	Exemplo de imagens geradas na etapa de pré-processamento a partir de combinações de bandas.	16
4.4	Estrutura do modelo híbrido CNN-SVM utilizado no presente trabalho.	18
4.5	Estrutura do modelo <i>ensemble</i> homogêneo.	19
4.6	Estrutura do modelo <i>ensemble</i> heterogêneo.	20
5.1	Matriz de confusão (adaptado de Nogare [5]).	22
5.2	Curva ROC e AUC.	23

Lista de Tabelas

2.1	Principais funções <i>kernel</i> para a SVM.	7
3.1	Resumo dos estudos apresentados e as respectivas técnicas e resultados . .	12
5.1	Resultados da comparação entre CNNs tradicionais.	24
5.2	Resultados da aplicação do <i>Fine Tuning</i> nos modelos tradicionais.	25
5.3	Resultados da variação da função <i>kernel</i> no Modelo Híbrido.	25
5.4	Resultados dos diferentes espaços de cores no modelo híbrido.	26
5.5	Resultados da comparação de <i>ensemble</i> homogêneo de CNNs tradicionais e <i>ensemble</i> homogêneo de modelos híbridos.	27
5.6	Resultados da comparação de <i>ensemble</i> heterogêneo de CNNs tradicionais e <i>ensemble</i> heterogêneo de modelos híbridos.	28
6.1	Evolução da sequência de experimentos realizados. Na tabela a acurácia é representada por Acc., a precisão Prec. e a revocação Revoc.	29

Lista de Abreviaturas e Siglas

ANM Agência Nacional de Mineração.

AUC Área Abaixo da Curva, do Inglês *Area Under The Curve*.

CNN Rede Neural Convolutacional, do Inglês *Convolutional Neural Network*.

FCN Rede Neural Convolutacional Totalmente Conectada, do Inglês *Fully Convolutional Network*.

NIR *Near Infrared*.

ReLU Unidade Linear Retificada, do Inglês *Rectified Linear Unit*.

ROC Característica de Operação do Receptor, do Inglês *Receiver Operating Characteristic Curve*.

SVM Máquina de Vetores de Suporte, do Inglês *Support Vector Machine*.

SWIR *Short-Wave Infrared*.

TIF *Tagged Image File*.

VNIR *Visible Near Infrared*.

Capítulo 1

Introdução

As barragens de rejeitos são grandes reservatórios que são compostos de água e resíduos sólidos provindos da atividade de mineração [6]. Estas estruturas podem representar grandes riscos à medida que possuem uma capacidade de causar danos ambientais em caso de um eventual rompimento. Os trágicos acidentes ocorridos em 2015 e 2019 nas cidades mineiras de Mariana e Brumadinho [7], respectivamente, demonstram concretamente quão elevado é o potencial de dano causados por alguma falha nas barragens.

Outro problema, que em conjunto se torna ainda mais prejudicial, são as barragens em situação irregular pelos mais diversos motivos. Segundo a reportagem realizada pela Agência Brasil [8], o Brasil possuía em 2018 cerca de 45% de suas barragens em situação irregular, sendo que 96% são barragens que não possuem informação da altura da fundação, 68% sem autorização, 45% não havia informação sobre o volume, 6% sem informação do principal uso, 3,5% sem apresentar coordenadas e, por fim, 42% não havia registros de nome.

Tendo em vista o problema de localizar possíveis barragens irregulares, há algumas maneiras para tentar solucionar tal problema [4, 9]. O uso de Inteligência Artificial, em especial os modelos de Aprendizado Profundo, têm sido utilizados na literatura para problemas relacionados à classificação de imagens, como mostram os autores Rizwan I Haque e Neuber [10] que utilizam Rede Neural Convolutacional, do Inglês *Convolutional Neural Network* (CNN). Os modelos híbridos também estão sendo utilizados para solucionar este problema. Esta técnica consiste na integração de diferentes modelos como é o caso do trabalho de Tao e Wei [11]. Além disso, também estão presentes na literatura os chamados modelos *ensemble* que, por sua vez, são a agregação de modelos para obtenção de um melhor resultado, podendo ser implementada de diversas maneiras. Como por exemplo, no trabalho de Turkoglu *et al.* [12].

A tarefa de detecção de barragens tem recebido uma atenção especial, com destaque para os últimos anos, como demonstram os autores Balaniuk *et al.* em [9] que utilizam

redes neurais para detecção de barragens em imagens de satélite. Além disso, devido ao tamanho da tragédia citada anteriormente, há alguns estudos que utilizam base de dados de barragens brasileiras como é o caso dos autores Ferreira *et al.* [4] que apresentam uma base de dados com imagens multiespectrais obtidas pelos satélites *Sentinel-2* e *Landsat-8* das barragens catalogadas no Brasil. No modelo proposto por Balaniuk *et al.* [9] é utilizado o mesmo catálogo para gerar a base de dados para treinamento e teste.

O objetivo do presente trabalho é uma proposta de modelo *ensemble* heterogêneo de modelos híbridos. Sendo estes composto por uma CNN para extração de características e Máquina de Vetores de Suporte, do Inglês *Support Vector Machine* (SVM) para a tarefa de predição de barragens em imagens multiespectrais. Para isso, foi necessário realizar uma sequência de experimentos que visam melhorar o desempenho dos modelos de CNNs tradicionais.

No Capítulo 2 serão abordados os conceitos fundamentais utilizados para o desenvolvimento do presente trabalho. No Capítulo 3 serão apresentados trabalhos da literatura sobre o tema que motivaram a definição da proposta dos métodos trabalhados. No Capítulo 4 é apresentada a metodologia proposta que descreve as etapas de construção do modelo. No Capítulo 5 são mostrados os experimentos realizados e os resultados obtidos em cada um deles. No Capítulo 6 é realizada uma discussão geral sobre os resultados obtidos. Por fim, no Capítulo 7 são apresentadas a conclusão do trabalho que foi realizado e as propostas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os conceitos teóricos necessários para a compreensão das técnicas utilizadas no presente trabalho. Na Seção 2.1 será apresentado o conceito de CNN, já na Seção 2.2 será apresentado o conceito de SVM e, por fim, nas demais seções serão apresentadas as técnicas *Transfer Learning* e *Fine Tuning*.

2.1 Redes Neurais Convolucionais

Em Aprendizado Profundo, as CNNs são um tipo de rede neural artificial que são amplamente utilizadas em problemas que envolvem classificação de imagens [13, 14]. Em 1980, Fukushima propôs a *Neocognitron* [15] que foi uma das primeiras arquiteturas CNNs apresentadas na literatura. Esta arquitetura foi inspirada na forma como alguns animais enxergam, através de um estudo das suas células do córtex visual [16].

A arquitetura da CNN que é a base para os modelos que utilizamos hoje foi desenvolvida por LeCun em 1998 [17]. Esta arquitetura, conhecida como *LeNet*, possui duas camadas convolucionais, duas camadas de *downsampling* alternadas entre as camadas convolucionais e ao fim da rede três camadas totalmente conectadas, sendo que a última realiza classificação com base nas características que foram extraídas pelas camadas anteriores.

A Figura 2.1 ilustra a estrutura de uma CNN tradicional e seus principais componentes são descritos a seguir [1]:

- **Camada de Entrada:** a entrada de uma CNN é normalmente uma matriz, sendo suas dimensões compatíveis com a altura, largura e quantidade de bandas da imagem que está sendo utilizada como entrada. Cada posição da matriz representa um *pixel* dessa imagem;

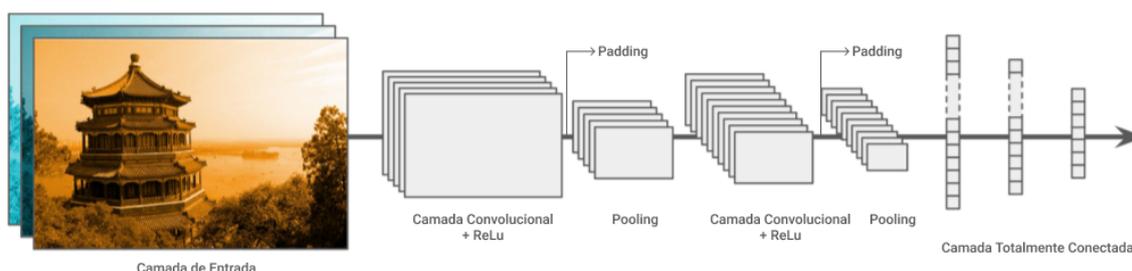


Figura 2.1: Arquitetura de uma CNN tradicional (adaptado de Geron [1]).

- **Camada Convolucional:** nas camadas convolucionais é aplicada a operação matemática de convolução descrita pela Equação 2.1,

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n), \quad (2.1)$$

sendo S o resultado desta operação que é conhecida como *feature map*. I é a matriz de entrada, K é o filtro também conhecido como *kernel* que será aplicado a essas entradas, e as variáveis (i, j) e (m, n) são respectivamente as coordenadas da matriz de entrada I e do filtro K . O resultado dessa operação geralmente é uma matriz com dimensões menores e pesos maiores;

- **Padding:** é uma técnica na qual alguns *pixels* normalmente com valor zero são adicionados ao redor da matriz gerada após a operação de convolução, com o objetivo de preservar as dimensões da matriz de entrada;
- **Função de Ativação:** as funções de ativação tem como objetivo trazer a não-linearidade ao modelo, sendo Unidade Linear Retificada, do Inglês *Rectified Linear Unit* (ReLU) uma função de ativação bastante utilizada nas camadas ocultas da convolução [18]. A função de ativação *ReLU* preserva os valores positivos dos *pixels* e substitui os valores negativos por zero ao aplicar $\max(0, x)$, sendo x o valor da saída da camada oculta anterior;
- **Camada de Pooling:** esta camada tem como objetivo simplificar a informação da camada convolucional anterior através de uma operação chamada *downsampling*, que realiza a diminuição do tamanho da matriz da imagem. De acordo com Geron [1], o método de *pooling* mais comum é o *max pooling*, no qual o maior valor de determinada região da matriz é utilizado para representar toda a região na formação da matriz de saída;
- **Camada Totalmente Conectada:** as camadas totalmente conectadas, também

conhecidas como Camadas Densas, normalmente tem o mesmo número de nós de saída que o número de classes do problema que está sendo trabalhado. Ao final, em sua última camada, temos uma função de ativação que é responsável por gerar a saída da rede. Esta função varia dependendo do tipo de problema que está sendo trabalhado [19, 20].

Com o passar dos anos, vários estudos foram realizados a respeito das CNNs e com isso esse modelo evoluiu bastante e surgiram diversas variações e novas arquiteturas baseadas nesses conceitos descritos anteriormente [21]. O presente trabalho utiliza três arquiteturas CNNs: *VGG16*, *InceptionV3* e *Densenet121*, sendo estes modelos outrora considerados estados-da-arte.

A arquitetura *VGG16* foi proposta em 2014 por Liu e Deng [22], sendo as principais diferenças em comparação com as arquiteturas antecessoras está na quantidade de camadas que foram adicionadas a esse modelo, tornando-o muito mais profundo. Outra diferença está em cada uma das camadas utilizarem um *kernel* de tamanho 3×3 que é bem menor comparado a modelos anteriores, como por exemplo, a *AlexNet* [23] onde os *kernels* são 7×7 . A arquitetura é chamada de *VGG16* por ter o número total de 16 camadas convolucionais e totalmente conectadas.

Já a *InceptionV3* foi proposta em 2015 por Szegedy *et al.* [24], surgindo como evolução do modelo *InceptionNet/GoogLeNet* [24] que por sua vez traz a ideia de além de aumentar a profundidade da rede, aumentar sua largura trazendo os chamados módulos *inception* que contém convoluções com diferentes tamanhos de *kernel* (5×5 , 3×3 , 1×1). Um dos objetivos da rede *InceptionV3* é gastar menos recursos computacionais. Sua principal modificação está no módulo *inception* da rede que passa a utilizar camadas convolucionais assimétricas, ou seja, ao invés de utilizar por exemplo um *kernel* 3×3 ela utiliza um *kernel* 3×1 seguido de outro 1×3 .

Por fim, a *DenseNet121* que foi proposta em 2017 no trabalho de Huang *et al.* [25]. Sua principal inovação em comparação com os outros modelos é o conceito de blocos densos, no qual recebem os *feature maps* gerados pelas camadas anteriores do mesmo modelo como entrada, ajudando assim o processo de treinamento. Este processo de passagem de dados de camadas anteriores para as outras ajuda a combater um problema conhecido como *vanishing gradient*. Este problema geralmente ocorre em redes com muitas camadas nas quais, após a passagem dos dados por cada camada, a alteração dos pesos ocorre com menos frequência ou não acontece, fazendo com que as redes aprendam de maneira lenta ou, em piores casos, não aprendam nada [26]. Outro importante ponto positivo da *DenseNet* é que por conta dessa reutilização das camadas, têm-se como resultado um modelo muito mais compacto, que por sua vez requer menos parâmetros do que outras CNNs, pois não há *feature maps* repetidos.

2.2 Máquinas de Vetores de Suporte

Uma SVM é um modelo de Aprendizado de Máquina supervisionado que foi baseado na teoria de aprendizado estatístico de Vapnik [27]. Este modelo é capaz de lidar com problemas lineares e não lineares, podendo ser usado tanto para tarefas de classificação quanto de regressão [1].

Para a tarefa de classificação, a SVM irá tentar separar as duas classes do problema através dos chamados hiperplanos ótimos que dentre os infinitos hiperplanos possíveis são os que separam as classes e obtêm a margem máxima¹ entre as amostras das classes. A Figura 2.2 ilustra o funcionamento da SVM em um problema de classificação binária linear. Na Figura 2.2(a) está um exemplo de alguns possíveis hiperplanos que conseguem separar com sucesso as duas classes, já na Figura 2.2(b) se encontra o hiperplano ótimo no qual possui a maior distância possível entre as duas classes. As linhas tracejadas representam os tais vetores de suporte, já o espaço entre elas e o hiperplano ótimo é chamado de margem máxima. Os círculos e quadrados representam as duas classes do exemplo.

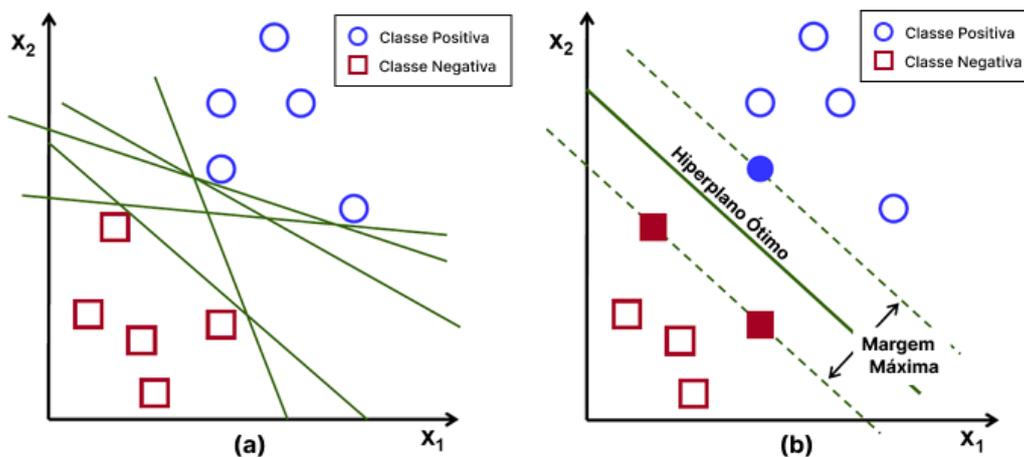


Figura 2.2: Exemplo do funcionamento de uma SVM (adaptado de Ippolito [2]).

As primeiras versões da SVM foram pensadas apenas para trabalhar com problemas linearmente separáveis, ou seja, quando é possível separar as classes através de hiperplanos. Porém, mais tarde, foi criada uma técnica chamada de truque de *kernel* que por meio da aplicação de funções *kernel* a SVM passa a conseguir lidar com problemas não linearmente separáveis. As principais funções *kernel* de acordo com Pedregosa *et al.* [28] estão descritas na Tabela 2.1.

¹Maior distância possível entre o hiperplano e amostra.

Tabela 2.1: Principais funções *kernel* para a SVM.

Tipo de Kernel	Função $K(x, x')$	Parâmetros
Linear	$(x \cdot x')$	
Polinomial	$(\gamma(x, x') + r)^d$	γ, d, r
<i>RBF (Radial-Basis Function)</i>	$exp(-\gamma x - x' ^2)$	$\gamma > 0$
Sigmoidal	$tanh(\gamma(x - x') + r)$	γ, r

2.3 *Transfer Learning*

Transfer Learning ou transferência de aprendizado é uma técnica bastante utilizada na área de aprendizado profundo que consiste em treinar previamente um modelo com alguma base de dados diferente e reaproveitar o aprendizado adquirido por esse modelo em outro problema sem precisar necessariamente treiná-lo do zero [29].

Usualmente quando um modelo é criado seus pesos são iniciados aleatoriamente e na medida que as épocas avançam durante o treinamento o modelo vai sendo aprimorado até um ponto em que ele consegue bons resultados na tarefa na qual está sendo utilizado. Ao aplicar a técnica de *Transfer Learning* são utilizados pesos vindo de um treinamento em outra base de dados. Esta abordagem funciona pois independentemente das classes dos dados que problema em questão tenha, algumas características mais genéricas como bordas e formatos, por exemplo, estão presentes em todo tipo de imagem. Assim, este conhecimento não precisaria ser adquirido novamente e poderia ser reaproveitado.

Além disso, outras vantagens do uso de *Transfer Learning* estão em não precisar que a base de dados seja tão grande e nem que sejam necessárias grandes quantidades de épocas para o modelo se ajustar ao problema e alcançar bons resultados [30]. Vale ressaltar que quanto mais semelhante for o conjunto de dados no qual a rede foi originalmente treinada e o do problema em questão, melhores resultados poderão ser obtidos.

2.4 *Fine Tuning*

Já o *Fine tuning* ou ajuste fino pode ser enxergado como uma forma de ir além da *transfer learning* e conseguir resultados ainda melhores para o modelo. Como mencionado anteriormente, ao utilizar o *transfer learning* será aproveitado um conhecimento que já foi obtido através de uma base de dados diferente. Aplicando o *Fine Tuning*, o objetivo é fazer com que a rede em que está sendo utilizada a técnica preserve os conhecimentos mais genéricos obtidos e aprenda características específicas da base de dados do problema em questão [31].

Durante o processo de *Fine Tuning*, que está representado na Figura 2.3, as primeiras camadas da rede em questão permanecem congeladas, ou seja, não estão habilitadas para a aprendizagem. Isto ocorre pois as primeiras camadas detectam padrões mais simples e gerais, e quanto mais profundas mais específico são os padrões que elas detectam. Por tanto ao mantê-las congeladas serão preservados os conhecimentos adquiridos. A partir daí, é permitido que as últimas camadas possam aprender, e assim o modelo assimilará padrões mais específicos da base da problemática. Com essa configuração é possível treinar a rede e ao final do procedimento o modelo apresentará resultados ainda melhores.

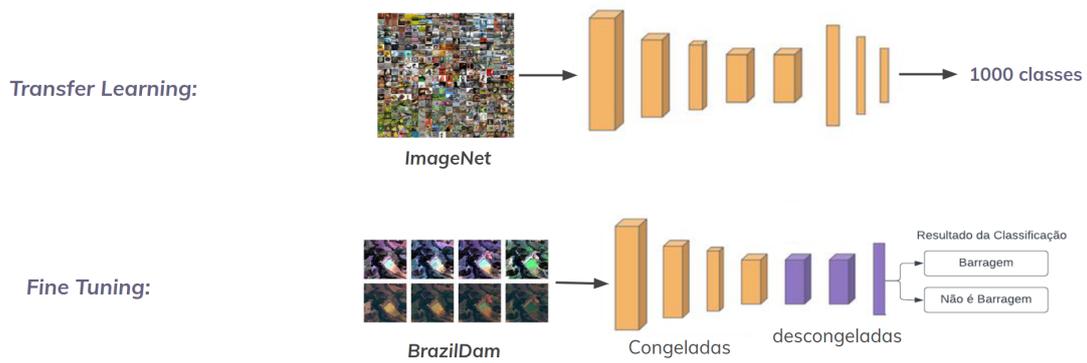


Figura 2.3: Exemplo do funcionamento do *Fine Tuning* (adaptado de Malgonde [3]).

Capítulo 3

Trabalhos Relacionados

Este capítulo apresenta as principais obras na literatura relacionadas à detecção de barragens, em especial as barragens de rejeitos utilizando imagens de satélite. Estas obras apresentam estudos que serão utilizados como base e que também norteiam a respeito de estudos ainda não realizados que inspiram o presente trabalho.

No Brasil existe um grande número de barragens de rejeitos e uma quantidade preocupante é considerada como de alto potencial de dano [8]. As tragédias em Mariana e Brumadinho provocaram um enorme prejuízo de vidas, socioambiental e consequências que permanecem até os dias atuais [7].

A Agência Nacional de Mineração (ANM)¹ é responsável pela vigilância destas barragens e possui um controle a respeito de diversas características, tais como risco, potencial de dano e localização. Além das barragens reguladas e catalogadas pelo governo, existem também barragens ilegais onde não é feito nenhum tipo de levantamento sobre os impactos ambientais causados por tal instalação. Uma maneira de tentar identificar estas barragens é utilizando imagens de satélite e aplicando modelos de Aprendizado Profundo para determinar a partir destas imagens a presença ou não de uma barragem.

Há alguns trabalhos na literatura que utilizam CNN para realizar a detecção de barragens de rejeitos a partir de imagens de satélite de alta resolução. No trabalho apresentado por Ferreira *et al.* [4] é introduzida a base de dados *BrazilDAM* a qual possui imagens dos satélites *Sentinel-2* e *Landsat-8*. Além disso, os autores realizam um *Benchmark* entre as principais CNNs utilizadas na literatura para a tarefa de detecção de objetos em imagens. Nos ensaios realizados as redes testadas foram *VGG16* [22], *AlexNet* [23], *DenseNet121* [25], *InceptionV3* [24], *SqueezeNet* [32] e *ResNet50* [33], sendo que a *DenseNet121* obteve o melhor resultado com uma acurácia média de 94,11% para a tarefa de classificação binária da base de dados sendo barragem ou não-barragem. Para alcançar esta acurácia foi utilizado o algoritmo *Adam* [34] como otimizador. Os resultados foram calculados a

¹<https://www.gov.br/anm/pt-br>

partir da média da acurácia sobre as cinco validações cruzadas realizadas no conjunto de teste. Além disso, os autores indicaram o intervalo com 95% de confiança obtido. Para esta tarefa foi utilizado apenas o conjunto de dados com as imagens obtidas do satélite *Sentinel-2* no ano de 2019. Este conjunto possui 1925 imagens, sendo 769 consideradas como barragem e 1156 consideradas não-barragem.

Outra abordagem utilizada na literatura é o uso de uma Rede Neural Convolutacional Totalmente Conectada, do Inglês *Fully Convolutional Network* (FCN) para a mesma tarefa de detecção mencionada anteriormente, como mostra o estudo de Balaniuk *et al.* [9], no qual é proposto o uso de uma FCN para segmentar regiões de barragens e minas a partir de coordenadas de áreas que contém ou não barragens de rejeitos. Para gerar os dados, os autores fazem uso de uma técnica de raspagem de dados para buscar essas coordenadas através de uma *API* do *Google Earth* e obter imagens multiespectrais dos locais indicados pelas coordenadas. Formando um conjunto que possui 2860 imagens, sendo 1397 barragens ou minas e 1463 que não são. Além disso, os resultados alcançados pelo autores foram promissores, sendo a acurácia de validação 97,44%, *f1-score* 97,49%, revocação 97,47% e Área Abaixo da Curva, do Inglês *Area Under The Curve* (AUC) 97,46%.

Dentro da literatura há ainda algumas outras abordagens como as propostas por Yang *et al.* [35] que utilizam outros parâmetros além das imagens de satélites, como sensores geológicos de desastres que monitoram a estabilidade de uma barragem a partir da altura da linha de saturação. Foi construído um sistema de alerta em tempo real que a através dos dados coletados por estes sensores prevê o risco de desastre. Neste caso, os autores obtiveram uma acurácia média para o alerta de risco em tempo real de 99,89%.

No caso da tarefa de segmentação da área que será classificada como uma barragem ou não, os autores Lyu *et al.* [36] propõem um *framework* baseado em Aprendizado Profundo. Para isto, utilizam a combinação da rede *YOLOv4* [37], que é responsável por detectar a presença de uma barragem, e um modelo de Floresta Aleatória otimizado que tem o papel de extrair a região correspondente à barragem. No trabalho de Lyu *et al.* [36] é utilizada uma base de dados com imagens obtidas a partir do *Google* com resolução de 2,05 metros de barragens de algumas cidades na China. Como resultados apresentados têm-se uma precisão de 99,6%, revocação de 89,9% para a classe positiva na qual a imagem possui barragem.

Abrangendo mais um pouco a área e indo além do tema de detecção de barragens, mas apresentando uma técnica que inspirou o presente trabalho, os autores Tao e Wei [11] propõem um modelo híbrido composto por uma CNN e uma SVM com objetivo de fazer reconhecimento de ervas daninhas em campos no período de invernos. A arquitetura CNN utilizada foi a *VGG16* para realizar a extração de características das imagens. Já

a SVM com a função *kernel RBF* é utilizada para realizar a classificação das imagens. Neste trabalho, o autor realiza alguns experimentos para definir o melhor parâmetro para a taxa de aprendizado e *momentum* para o modelo apresentado. Para explicar a adoção do modelo *VGG*, ainda é apresentado uma comparação entre os desempenhos dos modelos considerados estado-da-arte para a mesma tarefa apresentada. E ao final obtiveram como resultado uma acurácia de 92,1% na base de teste utilizando o modelo híbrido proposto *CNN-SVM*.

A utilização de uma SVM como camada de classificação de uma CNN é utilizada por Ahlawat e Choudhary [38]. Neste trabalho os autores utilizam um modelo híbrido *CNN-SVM* onde a CNN é utilizada para extração de características e a SVM com a função *kernel RBF* para a tarefa de classificação de imagens, utilizando a base de dados *MNIST* [39]. Com alguns ajustes de parâmetro o estudo revela um aumento na acurácia do modelo híbrido *CNN-SVM* em comparação com a utilização de apenas um classificador individual SVM. O melhor resultado para o conjunto de teste encontrado foi 97,85% utilizando apenas SVM e 98,95% utilizando o modelo híbrido.

O trabalho de Turkoglu *et al.* [12] tem como objetivo detectar doenças e pragas em plantas utilizando a técnica de *ensemble*. Para isso, os autores utilizam cinco modelos de CNN que são *alexNet*, *GoogleLeNet*, *Densenet201*, *Resnet50* e *Resnet101* onde cada um deles é utilizado para extração de características e uma SVM para realizar a predição quanto à doença/praga. Para definir a classificação final do modelo foi utilizada a votação majoritária entre as cinco predições. Como resultados foram obtidos uma acurácia de 97,56% e *f1-score* 97,07%.

Para elucidar e melhor visualizar cada estudo apresentado anteriormente, a Tabela 3.1 mostra as técnicas, o problema tratado e os resultados alcançados pelos autores. Os resultados dos estudos citados neste capítulo estão apresentados com a Acurácia (Acc.) alcançada, exceto o trabalho proposto por Lyu *et al.* [36] que apresenta como resultado a Precisão (Prec.), a Revocação (Revoc.) e Média das Precisões (Média Prec.). É importante ressaltar que o conceito destas métricas são apresentados no Capítulo 5.

O presente trabalho combina diversas técnicas discutidas na literatura mas que ainda não foram aplicadas para o problema de detecção de barragens em imagem de satélite. E com isso, propor um modelo capaz de obter resultados melhores do que os já apresentados na literatura para o problema em questão.

Tabela 3.1: Resumo dos estudos apresentados e as respectivas técnicas e resultados

Autores	Problema	Técnicas	Resultados
Ferreira <i>et al.</i> [4]	Detecção de barragens	CNN	Acc.: 94,11%
Balaniuk <i>et al.</i> [9]	Segmentação de barragens	FCN	Acc.: 97,44%
Yang <i>et al.</i> [35]	Predição de risco de barragens através de sensores	Regressão e Classificação baseada em Floresta Aleatória	Acc.: 99,89%
Lyu <i>et al.</i> [36]	Segmentação de barragens	YOLOv4 e Floresta Aleatória	Precisão: 99,6%, Revoc.: 89,9%, Média Prec.: 89,7%
Tao e Wei [11]	Detecção de ervas daninhas	CNN-SVM	Acc.: 92,1%
Ahlawat e Choudhary [38]	Reconhecimento de dígitos manuscritos	CNN-SVM	Ac.: 98,95%
Turkoglu <i>et al.</i> [9]	Detecção de pragas	Ensemble CNN-SVM	Acc.: 97,56%

Capítulo 4

Metodologia

A metodologia proposta neste trabalho consiste em um *ensemble* de modelos híbridos baseados em CNN e SVM para a tarefa de classificação binária de Barragens em imagens de satélite. A Figura 4.1 apresenta as etapas, processos e técnicas da metodologia proposta. As próximas seções descrevem em detalhes todas as etapas do fluxo presente na Figura 4.1, a saber: (1) aquisição da base de dados; (2) pré-processamento das imagens; (3) aplicação de *transfer learning* e *fine tuning* nas CNNs, e utilização da SVM como classificador que formam o modelo híbrido; (4) *ensemble* dos modelos da etapa anterior; e (5) etapa de votação majoritária para classificação.

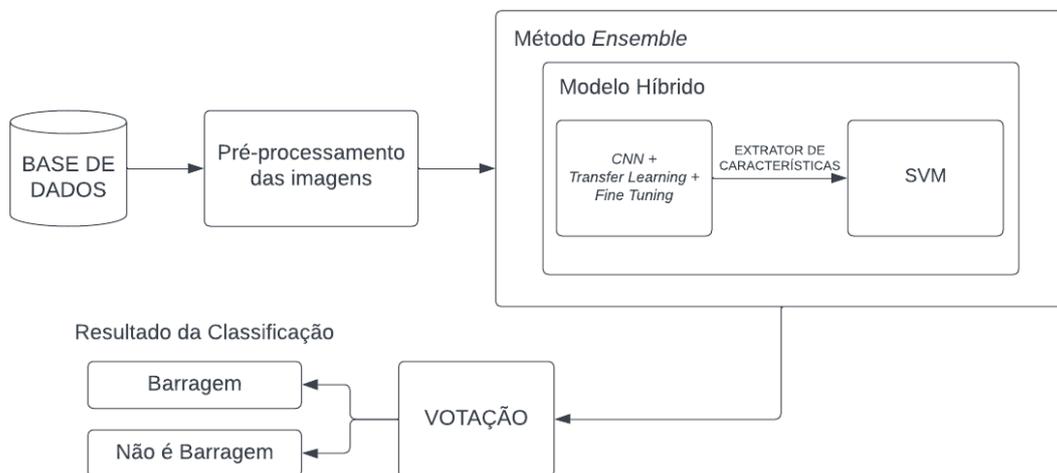


Figura 4.1: Visão geral da metodologia proposta.

4.1 Base de Dados

A base de dados que foi utilizada para treinamento e avaliação da metodologia proposta neste trabalho foi a *BrazilDAM* proposta por Ferreira *et al.* [4]. Esta base possui imagens multiespectrais que são imagens que contém diferentes comprimentos de ondas eletromagnéticas. Estas imagens foram obtidas por meio dos satélites *Sentinel-2*¹ e *Landsat-8*² e para cada um deles possui imagens da mesma região dos anos de 2016, 2017, 2018 e 2019. O *Sentinel-2* foi lançado como parte do programa *Copernicus* da Comissão Europeia em 23 de junho de 2015, possuindo 13 bandas espectrais.

Para os experimentos serão considerados somente as imagens do ano de 2019 do *Sentinel-2* pois foram utilizadas antes na literatura e o uso de imagens da mesma região de diferentes anos poderia causar o enviesamento do modelo para a tarefa que está sendo trabalhada. A base escolhida possui no total 1.924 imagens na qual 769 são de barragens e 1.155 não são. Vale ressaltar que o satélite *Landsat-8* não foi utilizado pois possui apenas oito bandas e as imagens seriam as mesmas. A Figura 4.2 mostra exemplos de imagens em *RGB* da base de dados nas quais as Figura 4.2(a) e (c) são da classe barragem e Figura 4.2(b) e (d) da classe não barragem.

As imagens da base de dados *BrazilDAM* [4] estão no formato de arquivo *Tagged Image File* (TIF) e portanto possui além das imagens multiespectrais algumas informações técnicas como minério principal, altura, volume, método construtivo, categoria de risco e dano potencial associado à barragem [4]. Embora existam todas essas informações para o presente trabalho somente serão utilizadas as imagens, pois o objetivo é apenas detectar as barragens.

4.2 Pré-processamento das Imagens

Trabalhar diretamente com arquivos TIF exige bastante poder computacional por conta das diversas informações que estão contidas nele como visto na Seção 4.1. Este tipo de arquivo costuma ter tamanho grandes, por exemplo, as imagens do satélite *Sentinel-2* do ano de 2019 têm um tamanho de 11,8 GiB. Além disso, como será feita a utilização das técnicas de *transfer learning* e *fine tuning*, trabalhar com imagens multiespectrais com 13 bandas não é o ideal por se diferenciar muito da base na qual os modelos serão pré-treinados [40].

Portanto foi considerada uma etapa de pré-processamento das imagens, com objetivo de trabalhar com as imagens em um formato mais convencional *Portable Network Graphics* - *PNG*, no qual não são perdidas tantas informações em sua compressão [41]

¹<https://sentinel.esa.int/web/sentinel/missions/sentinel-2>

²<https://www.usgs.gov/landsat-missions/landsat-8>

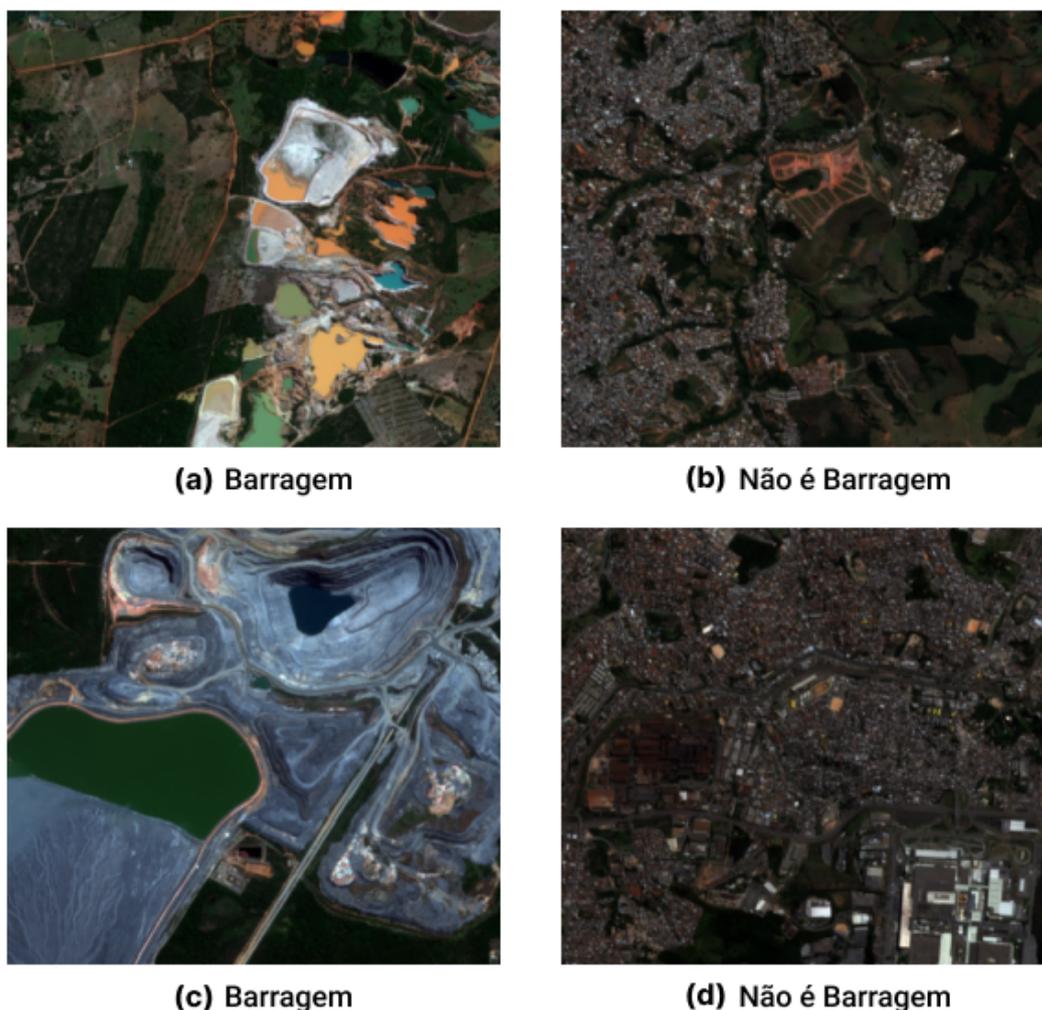


Figura 4.2: Exemplos de imagens de cada classe da base de dados em *RGB* [4].

utilizando somente três bandas para que a aplicação das técnicas citadas anteriormente possam acontecer da melhor maneira e assim esperar por melhores resultados. Além disso, possibilitar a análise da combinação de diferentes bandas.

Como dito na Seção 4.1, o *Sentinel-2* possui 13 bandas espectrais que podem ser agrupadas da seguinte forma³:

- **Bandas de 4×10 metros:** as bandas *RGB* (*Red* (B4), *Blue* (B2) e *Green* (B3)) e a banda *Near Infrared* (NIR) (B8);
- **Bandas de 6×20 metros:** as bandas *Visible Near Infrared* (VNIR) (B5, B6, B7, B8A) e as bandas *Short-Wave Infrared* (SWIR) (B11 e B12);

³<https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric>

- **Bandas de 3×60 metros:** a banda *Ultra blue (Coastal and Aerosol)* (B1) e a banda SWIR (B10).

A utilização de combinações de bandas, permite extrair informações específicas de uma imagem. Por exemplo, existem combinações de bandas que destacam características geológicas, agrícolas ou de vegetação em uma imagem [42]. Para o presente trabalho foram geradas 5 versões da base de dados utilizando o *software GDAL*⁴. Cada versão representa as combinações das diferentes bandas com base nos padrões presentes na aplicação *Sentinel Playground* [43]. Na Figura 4.3 podem ser vistos os resultados da combinação de diferentes bandas na mesma imagem. Além disso, essas combinações foram nomeadas para as novas imagens pré-processadas, sendo: (a) *RGB* (B4, B3, B2), (b) *SWIR* (B12, B8A, B4), (c) *Infrared* (B8, B4, B3), (d) *Agriculture* (B11, B8, B2) e (e) *Bathymetric* (B4, B3, B1). Por fim, as imagens foram redimensionadas através do método de interpolação para o tamanho de 224×224 .

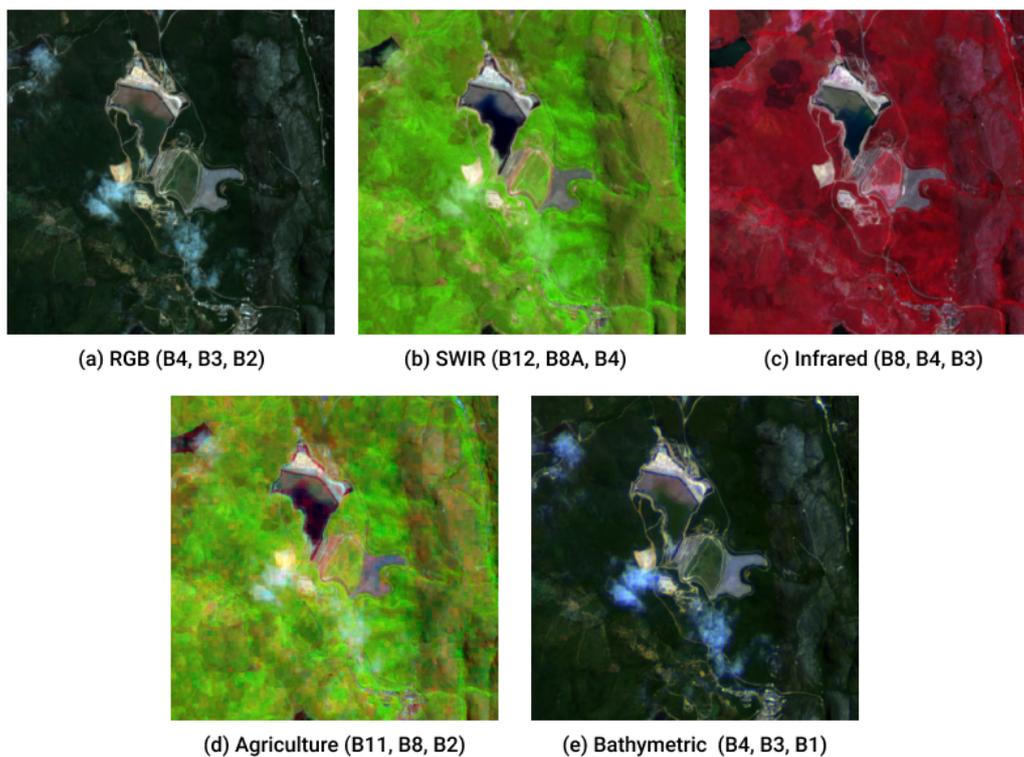


Figura 4.3: Exemplo de imagens geradas na etapa de pré-processamento a partir de combinações de bandas.

⁴<https://gdal.org>

4.3 Aplicação de *Transfer Learning* e *Fine Tuning* nas CNNs

Como visto nas Seções 2.3 e 2.4, as técnicas de *transfer learning* [29] e *fine tuning* [31] são, respectivamente, capazes de aumentar drasticamente os resultados de um modelo, principalmente quando a base de dados a ser utilizada não é tão grande. Portanto, para o presente trabalho foram utilizadas as técnicas citadas anteriormente em cada uma das três arquiteturas CNNs farão parte dos experimentos *VGG16* [22], *DenseNet121* [25] e *InceptionV3* [24]. Vale ressaltar que os modelos estão sendo utilizados com os parâmetros padrões encontrados na biblioteca *keras* presente no *TensorFlow* [44].

Para a realização do *transfer learning* foram utilizados os pesos a partir da base de dados *ImageNet* [45], que é uma base de dados muito utilizada na literatura, e conta com mais de 1 milhão de imagens distribuídas em 1000 classes. A base *ImageNet* foi a escolhida por conta da sua variedade de dados o que pode contribuir para o aprendizado de características genéricas do modelo. Para obter os pesos para cada uma das arquiteturas foram utilizados os recursos disponível da biblioteca *keras*.

Para realizar o *fine tuning*, primeiro foi executada cada arquitetura citada anteriormente por dez épocas considerando o subconjunto de treinamento e validação, que serão abordados na Seção 5.1, com a aprendizagem de todas as suas camadas congeladas. Após isso foram executadas mais vinte épocas deixando somente o equivalente a 20% das camadas descongeladas, para aprenderem as especificidades da base de dados, totalizando 30 épocas. Vale observar que estas camadas são as últimas do modelo.

Para a primeira execução do *fine tuning* foi utilizado como otimizador o *Adam* [34] com um *learning rate* de 0,0001. Já na segunda execução do *fine tuning* o otimizador utilizado foi *RMSprop* [46] com *learning rate* de 0,00001. Para ambas execuções a função *loss* utilizada foi *Binary Cross Entropy* [47] pois o problema trabalhado é de classificação binária e para o parâmetro de métrica foi utilizada a acurácia que será explicada em detalhes na Seção 5.2. Como foi dito na Seção 4.2 as imagens foram redimensionadas para o tamanho de 224×224 que é o mesmo das imagens presentes na base *ImageNet*. Além disso, foi utilizada a técnica de *data augmentation* através do método *ImageDataGenerator* do *keras* [44] para o treinamento dos modelos, não com a finalidade de aumentar a quantidade de imagens da base e sim para trazer diversidade durante o treinamento, utilizando operações aleatórias de *zoom*, distorção no sentido anti-horário e giro horizontal da imagem.

4.4 Modelos Híbridos

O modelo híbrido consiste em uma integração de modelos visando obter melhores resultados em comparação com a utilização dos modelos tradicionais [48]. No presente trabalho foi desenvolvido um modelo híbrido que é composto por CNN e SVM. Esta abordagem tem mostrado resultados promissores na literatura [11, 38]. Como visto na Seção 2.1, as primeiras camadas de uma CNN, em especial as camadas convolucionais, são responsáveis pela extração de características das imagens que estão sendo processadas e a camada totalmente conectada que está localizada ao final da rede é responsável por realizar a classificação de imagens, neste caso como barragem ou não barragem.

Diante disso, a CNN foi considerada apenas para extrair as características da imagem. Foram removidas das CNNs as suas camadas totalmente conectadas que passaram pelo processo de *transfer learning* e *fine tuning* como citado na Seção 4.3. Dessa forma, as imagens da base foram transformadas em vetores de características. Após esse processo, os vetores resultantes são utilizados como dados de entrada para a SVM que realizará a classificação. Foram utilizados os valores padrões da biblioteca *Scikit-learn* [28] para os parâmetros da SVM, com exceção da função *kernel* que será explorada na Seção 5.5. A Figura 4.4 mostra em formato de fluxograma o processo descrito nesta seção.

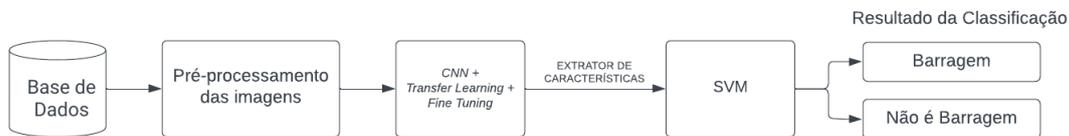


Figura 4.4: Estrutura do modelo híbrido CNN-SVM utilizado no presente trabalho.

4.5 Modelos *Ensemble*

Modelos *Ensemble* são a agregação de múltiplos modelos de classificação que utilizam o resultado de cada classificador em benefício da solução do problema [49]. Pode-se pensar que cada classificador individual que compõem o *ensemble* fosse como um especialista em algo e o resultado final é gerado a partir da combinação dos resultados de cada especialista. Existem diversas técnicas para a realização de *ensemble*, no presente trabalho foram consideradas duas variações sendo elas a homogênea e a heterogênea [49].

Uma parte importante das arquiteturas *ensemble* é a combinação dos modelos pois é onde os resultados serão convergidos em resultado final. Para o presente trabalho foi utilizado a estratégia de voto majoritário [49], que consiste em cada modelo dar o resultado da sua classificação e o resultado da maioria é o resultado final. Para que essa técnica

funcione é necessária que a quantidade de modelos presentes no *ensemble* seja ímpar para não acontecer empates. Esta abordagem foi escolhida por se tratar de um problema de classificação binária.

No caso do *ensemble* homogêneo, o conjunto de classificadores utilizados para realizar a tarefa de classificação pode ser idêntico. O classificador poderá receber um conjunto de dados diferentes, conseqüentemente, gerando diferentes modelos de acordo com quantidade de subconjuntos considerados. Desta forma, o classificador pode aprender uma característica específica do seu conjunto e contribuir de forma diferente no resultado. A Figura 4.5 apresenta o *ensemble* homogêneo considerado no presente trabalho o qual é constituído por modelos híbridos conforme visto na Seção 4.4. O modelo possui diferentes bases de dados as quais estão presentes na Figura 4.3. Estas bases alimentam instâncias diferentes do mesmo classificador. Os subconjuntos de teste de todas as bases de dados são agregados para a criação da base de testes que será utilizada para a realização da classificação no módulo de votação. Este módulo agrega todos os cinco modelos e realiza a votação majoritária.

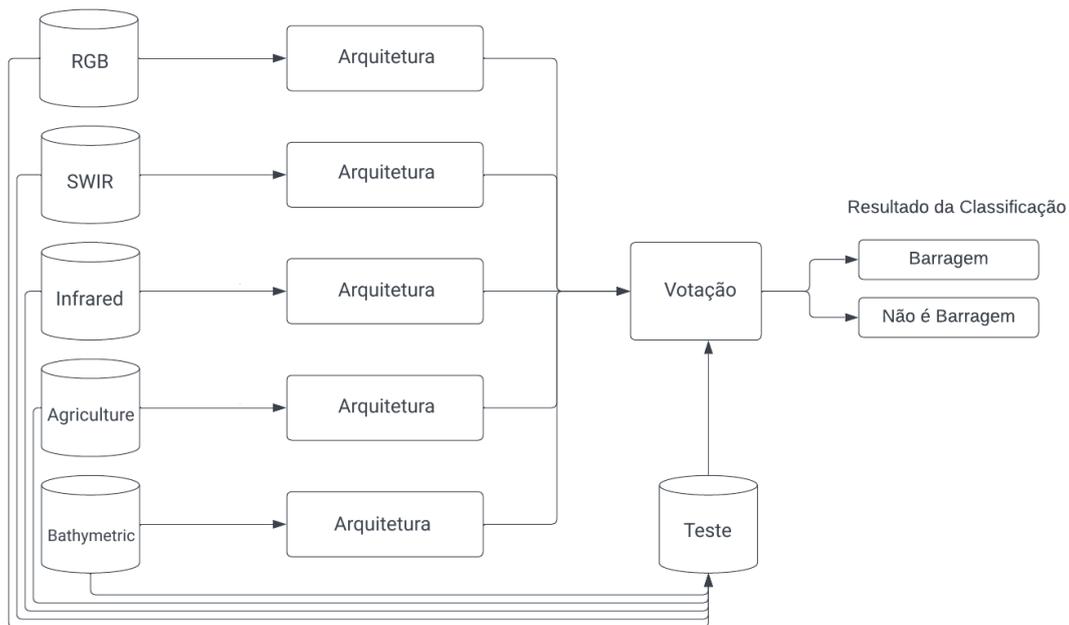


Figura 4.5: Estrutura do modelo *ensemble* homogêneo.

Já no caso do *ensemble* heterogêneo, o conjunto de classificadores possuem diferentes tipos de arquitetura. Nesse caso, a base de dados que é utilizada por cada um deles é a mesma, dessa forma cada arquitetura irá contribuir com a sua especialidade. A Figura 4.6 apresenta um exemplo de *ensemble* heterogêneo, que possui a mesma base de dados alimentando três configurações de modelos híbridos diferentes e ao fim os três

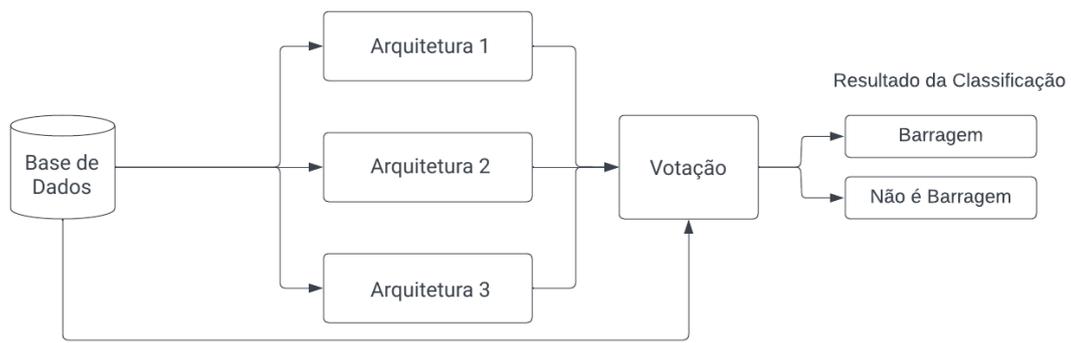


Figura 4.6: Estrutura do modelo *ensemble* heterogêneo.

são convergidos no módulo de votação que a realiza de forma majoritária resultando na classificação final.

Capítulo 5

Resultados Experimentais

Este capítulo apresenta os resultados obtidos em cada um dos seis experimentos realizados neste trabalho. Além disso, descreve as métricas utilizadas para avaliação dos resultados, como foi feita distribuição da base de dados e o ambiente experimental em que foram realizados os experimentos.

5.1 Distribuição dos Dados

Para a realização dos experimentos, a base de dados que foi vista na Seção 4.1 foi dividida de forma estratificada, ou seja, com proporção similar entre as classes por partição em subconjuntos de treino, validação e teste. Para treino foram utilizados 70% dos dados, o equivalente a 1.346 imagens, e para validação e teste 15% cada, o equivalente a 289 imagens.

5.2 Métricas de Desempenho

A Figura 5.1 representa a matriz de confusão que por sua vez expressa as combinações possíveis de resultados das predições para as classes do problema trabalhado. Através dos valores presentes na matriz de confusão é possível formular diversas métricas [50]. Para este trabalho os valores da matriz de confusão podem ser interpretados da seguinte forma:

- **Verdadeiro Positivo (VP)**: a imagem foi classificada como barragem e de fato é uma barragem;
- **Falso Positivo (FP)**: a imagem não é uma barragem porém foi classificada como barragem;
- **Verdadeiro Negativo (VN)**: a imagem não é uma barragem e de fato foi classificada como não sendo barragem;

		Valor Predito	
		Barragem	Não é Barragem
Valor Real	Barragem	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não é Barragem	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Figura 5.1: Matriz de confusão (adaptado de Nogare [5]).

- **Falso Negativo (FN)**: a imagem é uma barragem porém foi classificada como não sendo barragem.

Para realizar a avaliação dos experimentos presentes neste capítulo, serão utilizadas as seguintes métricas [50]: Acurácia, Precisão, Revocação, *F1-score* e AUC. A acurácia pode ser expressa pela Equação 5.1, que indica um desempenho geral do modelo, ou seja, de todas as classificações, quantas o modelo classificou corretamente.

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN} \quad (5.1)$$

Já a precisão pode ser expressa pela Equação 5.2, na qual dentre todas as classificações de classe positiva que o modelo fez, quantas estão corretas.

$$Precisão = \frac{VP}{VP + FP} \quad (5.2)$$

Por sua vez, a revocação pode ser expressa pela Equação 5.3, que pode ser entendida como dentre todas as situações de classe positiva como valor esperado, quantas estão corretas.

$$Revocação = \frac{VP}{VP + FN} \quad (5.3)$$

Por fim, o *F1-Score* que pode ser expresso pela Equação 5.4 sendo a média harmônica entre precisão e revocação. O *F1-Score* atinge seu melhor valor em 1 (precisão e revocação perfeitos) e pior em 0.

$$F1 - Score = 2 \times \frac{Precisão \times Revocação}{Precisão + Revocação} \quad (5.4)$$

A curva Característica de Operação do Receptor, do Inglês Receiver Operating Characteristic Curve (ROC) e a AUC são ferramentas poderosas para medir e comparar o

desempenho de modelos de classificação binárias. A Figura 5.2 ilustra a curva ROC que é um gráfico que permite estudar a variação da sensibilidade¹ e especificidade² descrita pela Equação 5.5, para diferentes pontos de corte na probabilidade estimada (*thresholds*). Quanto mais próximo do canto superior esquerdo a curva ROC, melhor está o modelo. A AUC é uma medida de área que facilita a comparação entre curvas ROC. Para a AUC quanto mais próximo do valor um, indica que o modelo está separando as classes de forma correta.

$$Especificidade = \frac{VN}{VN + FP} \quad (5.5)$$

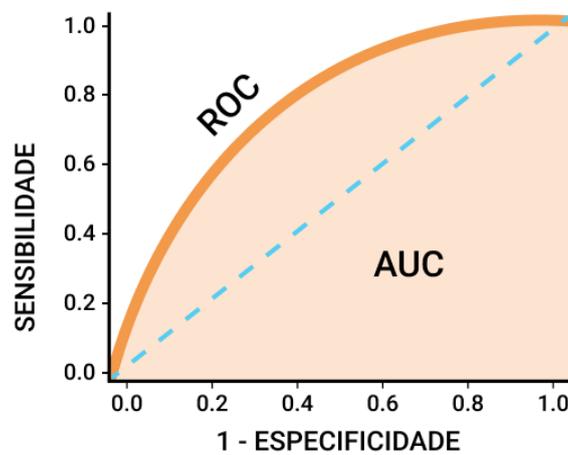


Figura 5.2: Curva ROC e AUC.

Para o presente trabalho o principal objetivo é detectar a presença de uma barragem em uma imagem, isto é, realizar a classificação com sucesso da classe positiva. Por esse motivo, o *F1-score* se torna a métrica mais relevante para a comparação entre os modelos experimentados, já que representa a média harmônica da Precisão e Revocação, os quais indicam o desempenho do modelo dentro das classe positivas. Vale observar que todas as métricas foram calculadas por meio do módulo *metrics* do *SciKit-Learn*.

5.3 Experimento I: CNNs Utilizando *Transfer Learning*

Esse experimento tem como objetivo comparar os resultados das três arquiteturas CNNs *VGG16*, *DenseNet121* e *InceptionV3*, utilizando somente a técnica de *Transfer Learning*,

¹O mesmo conceito de revocação

²Dentre todas as situações de classe negativo como valor esperado, quantas estão corretas

conforme visto na Seção 4.3, sem utilização do *Fine Tuning*. Com isso entender se somente com *transfer learning* o modelo é capaz de alcançar resultados promissores para a tarefa de classificação de imagens. Para as etapas de treino, validação e teste foi utilizada a versão da base de dados no espaço de cor *RGB* conforme visto na Seção 4.2 por ser o mais comum na literatura. Os resultados para o subconjunto de teste são apresentados na Tabela 5.1 no qual os melhores resultados estão em negrito.

Tabela 5.1: Resultados da comparação entre CNNs tradicionais.

Modelos	Acurácia	Precisão	Revocação	F1-Score	AUC
VGG16	0,792	0,846	0,797	0,821	0,888
DenseNet121	0,820	0,895	0,791	0,840	0,923
InceptionV3	0.844	0.829	0.930	0,877	0,906

É possível observar a partir dos resultados presentes na Tabela 5.1 que o modelo *InceptionV3* alcançou uma Acurácia, Revocação e *F1-score* melhores em comparação com os demais modelos. Os bons resultados para Acurácia e Revocação da *InceptionV3* indicam que o modelo está realizando boas classificações no cenário geral e, em especial, para classe positiva (barragem), já que obteve um bom resultado para *F1-score*.

Já a *Densenet121* obteve a melhor AUC, o que significa que consegue separar as classes com sucesso. Já a Precisão indica que a previsão das imagens indicadas como barragem contém poucos Falsos Positivos. Para este experimento a *VGG16* teve os piores resultado em comparação com os demais.

5.4 Experimento II: CNNs Utilizando *Transfer Learning* e *Fine Tuning*

Análogo ao Experimento da Seção 5.3, este experimento compara as arquiteturas tradicionais, porém dessa vez a técnica *Fine Tuning* vista na Seção 4.3 foi utilizada. Aqui o objetivo é conseguir melhorar os resultados por meio da aplicação do *Fine Tuning* em comparação com o que foi alcançado sem o uso da técnica. A base de dados utilizada foi a mesma do experimento anterior. Os resultados para o subconjunto de teste são apresentados na Tabela 5.2 no qual os melhores resultados estão em negrito.

A Tabela 5.2 demonstra que de fato a aplicação da técnica *Fine Tuning* alcançou melhoras significantes para quase todas as métricas avaliadas nos modelos. Apenas a Revocação da *InceptionV3* foi ligeiramente pior em relação ao experimento anterior. Além disso, a *VGG16*, que obteve os piores resultados anteriormente, foi capaz de aumentá-los

Tabela 5.2: Resultados da aplicação do *Fine Tuning* nos modelos tradicionais.

Modelos	Acurácia	Precisão	Revocação	F1-Score	AUC
VGG16	0,906	0,905	0,942	0,923	0,952
DenseNet121	0,923	0,946	0,924	0,935	0,960
InceptionV3	0,889	0,893	0,924	0,909	0,953

de tal forma que os seus resultados foram superiores aos melhores do experimento da Seção 5.3. Inclusive, foi a que obteve a melhor revocação para o experimento, ou seja, de todas as imagens que são barragens previu corretamente a classificação como sendo barragem.

A *Densenet121* foi a que mais aproveitou os benefícios advindos da técnica aplicada, conseguindo resultados promissores para acurácia, precisão, *F1-score* e AUC. Sendo o melhor modelo para este experimento.

5.5 Experimento III: Modelo Híbrido

Para este experimento foi escolhida a arquitetura que obteve o melhor resultado no experimento da Seção 5.4 que foi a *DenseNet121*. Dito isto foi montado um modelo híbrido utilizando a rede para extração de características, ou seja sem as camadas totalmente conectadas, e a SVM para realizar a classificação. Portanto, os objetivos são descobrir se o modelo híbrido terá bons resultados em relação aos modelos tradicionais vistos nos experimentos anteriores e descobrir qual a função *kernel* melhor se encaixa para o modelo. Os resultados para o subconjunto de teste são apresentados na Tabela 5.3 no qual os melhores resultados estão em negrito.

Tabela 5.3: Resultados da variação da função *kernel* no Modelo Híbrido.

Kernel SVM	Acurácia	Precisão	Revocação	F1-Score	AUC
Linear	0,941	0,948	0,953	0,951	0,970
RBF	0,930	0,927	0,959	0,943	0,980
Polinomial	0,923	0,912	0,965	0,938	0,976
Sigmoidal	0,937	0,937	0,959	0,948	0,976

Tendo em vista que todas as métricas receberam um aumento significativo é evidenciado que o modelo híbrido de fato foi capaz de melhorar em relação aos resultados dos modelos tradicionais. Além disso, A Tabela 5.3 deixa explícito que a função *kernel* linear

também obteve os melhores indicativos e isto pode ser explicado pelos conceitos vistos na Seção 2.2.

5.6 Experimento IV: Espaços de cores

Baseado nos resultados do experimento da Seção 5.5 o melhor modelo híbrido constatado foi a combinação *DenseNet121* e SVM com a função *kernel* linear. Até então o espaço de cor que estava sendo utilizado em todos os experimentos era o *RGB*, portanto o objetivo deste experimento é verificar se a mudança de espaço de cor influencia no resultado do modelo. Os resultados para o subconjunto de teste são apresentados na Tabela 5.4 no qual os melhores resultados estão em negrito.

Tabela 5.4: Resultados dos diferentes espaços de cores no modelo híbrido.

Espaço de Cor	Acurácia	Precisão	Revocação	F1-Score	AUC
RGB	0,941	0,948	0,953	0,951	0,970
SWIR	0,941	0,943	0,959	0,951	0,981
InfraRed	0,927	0,928	0,945	0,936	0,976
Agriculture	0,941	0,933	0,971	0,951	0,966
Bathymetric	0,889	0,880	0,933	0,905	0,952

A Tabela 5.4 traz o resultado de todas as combinações de espaços de cores aplicadas no modelo híbrido *Densenet121*-SVM. Os resultados para *RGB*, *SWIR* e *Agriculture* foram bem próximos. Sendo que os três modelos, inclusive, chegaram no mesmo resultado para *F1-score* e Acurácia. O *RGB* obteve a maior precisão que foi de 0,948, o *Agriculture* obteve a maior revocação que foi de 0,971 e o *SWIR* obteve a maior AUC que foi de 0,981.

Considerando a diferença entre estas métricas, o *SWIR* com a AUC obteve a maior distância de valores em comparação com os valores de Precisão e Revocação. Os experimentos seguirão utilizando o espaço de cor *SWIR*, pois a AUC indica que o modelo utilizando esta base de dados está lidando melhor com a separação das classes em comparação com os outros espaços de cores.

5.7 Experimento V: Modelo *Ensemble* Homogêneo

O experimento da Seção 5.6 mostrou que a mudança de espaço de cor realmente influencia no resultado do modelo, portanto esse experimento tem como objetivo testar se com

o uso de um modelo *ensemble* homogêneo é possível melhorar ainda mais os resultados. Para isso serão utilizadas cinco instâncias do modelo híbrido *DenseNet121*-SVM com a função *kernel* linear onde cada um deles é treinado com uma variação da base de dados. Para também avaliar se o modelo híbrido funcionará bem em conjunto com o *ensemble homogêneo*, foi feita uma comparação utilizando *ensembles* dos modelos híbridos *DenseNet121*-SVM e das CNNs *Densenet121* tradicionais. Os resultados para o subconjunto de teste são apresentados na Tabela 5.5 no qual os melhores resultados estão em negrito.

Tabela 5.5: Resultados da comparação de *ensemble* homogêneo de CNNs tradicionais e *ensemble* homogêneo de modelos híbridos.

Ensemble Homogêneo	Acurácia	Precisão	Revocação	F1-Score	AUC
CNNs Tradicionais	0,892	0,909	0,908	0,908	0,927
Modelos Híbridos	0,913	0,890	0,971	0,929	0,975

É evidente que o *ensemble* de modelos híbridos teve resultados superiores em comparação com o *ensemble* de CNNs tradicionais, como verifica-se na Tabela 5.5. Todavia, os modelos *ensembles* homogêneos não conseguiram aumentar os resultados em comparação com os experimentos da Seção 5.6. Isto demonstra que o conhecimento adquirido para trabalhar com um espaço de cor diferente do qual foi treinado por cada instância não funcionou bem. Além disso, a técnica de *ensemble* homogêneo exige mais recursos para sua execução já que precisa de múltiplas bases de dados para seu treinamento.

5.8 Experimento VI: Modelo *Ensemble* Heterogêneo

Os resultados do experimento da Seção 5.5 mostraram que o *ensemble* homogêneo não conseguiu melhorar os resultados, portanto neste experimento será utilizado um modelo *ensemble* heterogêneo, para verificar se a combinação de diferentes modelos serão capazes de melhorar os resultados. Para isso a base de dados utilizada foi a com o espaço de cores SWIR pois ela obteve o melhor resultado no experimento da Seção 5.4 e serão feitos três modelos híbridos utilizando as três arquiteturas *VGG16*, *DenseNet121* e *InceptionV3*. Desta mesma forma, será comparado neste experimento o *ensemble* de modelos híbridos com os tradicionais. Os resultados para o subconjunto de teste são apresentados na Tabela 5.5 no qual os melhores resultados estão em negrito.

A Tabela 5.6 traz resultados interessantes da utilização do *ensemble* heterogêneo utilizando modelos híbridos. Mais uma vez, como esperado, o modelo híbrido apresentou melhores resultados em relação ao modelo com CNNs tradicionais. Outro ponto importante é que o modelo proposto neste experimento obteve os melhores resultados em todas

Tabela 5.6: Resultados da comparação de *ensemble* heterogêneo de CNNs tradicionais e *ensemble* heterogêneo de modelos híbridos.

Ensemble Heterogêneo	Acurácia	Precisão	Revocação	F1-Score	AUC
CNNs Tradicionais	0,951	0,954	0,965	0,959	0,984
Modelos Híbridos	0,961	0,950	0,988	0,968	0,987

as métricas em comparação com todos os outros experimentos. Sendo assim, é possível concluir que o *ensemble* heterogêneo com modelos híbridos para a tarefa de classificação de barragens possui um resultado promissor em comparação com os outros modelos experimentados.

5.9 Ambiente Experimental

O Desenvolvimento, treinamento e execução de todos os experimentos foram realizados no ambiente *Google Colaboratory Pro* em um computador com processador *Intel(R) Xeon(R) CPU @ 2.30GHz*, placa de vídeo *Tesla P100-PCIE-16GB*, 25GiB de memória *RAM* e *HD* 125GiB. Além disso, foi utilizada a linguagem de programação *Python*, as bibliotecas *keras* presente no *TensorFlow* [44] e o *Scikit-learn* [28]. Todos os seis experimentos levaram em torno de 10 horas e 47 minutos para serem finalizados. Vale ressaltar que esse tempo contempla somente a leitura da base e execução do treinamento, validação e teste. Além disso, esses valores foram obtido a partir de uma única execução.

Capítulo 6

Discussão dos Resultados

Por meio dos resultados obtidos pelos experimentos propostos neste capítulo e apresentados na Tabela 6.1, é possível identificar uma série de fatores que são relevantes para o problema de classificação de barragens em imagens de satélites. No experimento da Seção 5.3, utilizando apenas o *transfer learning* foi possível obter bons resultados já que o procedimento realizado foi apenas definir as CNNs tradicionais e pré carregar os pesos da base *Imagenet*. Ao final do experimento, a *InceptionV3* se mostrou com melhores resultados.

Tabela 6.1: Evolução da sequência de experimentos realizados. Na tabela a acurácia é representada por Acc., a precisão Prec. e a revocação Revoc.

Experimentos	Acc.	Prec.	Revoc.	F1-score	AUC
Experimento I: InceptionV3	0,844	0,829	0,930	0,877	0,906
Experimento II: Densenet121	0,923	0,946	0,924	0,935	0,960
Experimento III: Híbrido e linear	0,941	0,948	0,953	0,951	0,970
Experimento IV: SWIR	0,941	0,943	0,959	0,951	0,981
Experimento V: Ensemble Homog.	0,913	0,890	0,971	0,929	0,975
Experimento VI: Ensemble Heter.	0,961	0,950	0,988	0,968	0,987

Já no experimento da Seção 5.4, utilizando o *Fine Tuning* é notório um aumento significativo nos resultados, o que mostra o quanto a técnica funcionou bem para o problema aqui estudado. Porém, como a *InceptionV3* não conseguiu melhorar os resultados nas mesmas proporções que os demais modelos, pode-se verificar que a técnica em questão funciona melhor em determinados modelos.

Seguindo para o experimento da Seção 5.5, foi possível identificar que a utilização de modelos híbridos é capaz de melhorar ainda mais os resultados obtidos na etapa anterior. Além disso, ficou evidenciado que a função *kernel* linear se adaptou melhor ao modelo.

Partindo para o experimento da Seção 5.6, a variedade de espaço de cores se mostrou capaz de influenciar o modelo, embora não tão significativamente quanto o esperado. Apesar de que para os olhos humanos a combinação de outras bandas de cores pareça tão diferente, aparentemente, para o modelo não há tanta distinção no processo de aprendizado.

No experimento da Seção 5.7 foi inesperado a piora nos resultados, visto que foram utilizados uma quantidade maior de modelos combinados e um conjunto de base dados o qual abrange cinco combinações de espaço de cores. Além disso, a técnica do *ensemble* visa aumentar os resultados alcançados pelo modelo proposto e isto não ocorreu no experimento.

Por fim, no experimento da Seção 5.8 o *ensemble* heterogêneo dos modelos híbridos se mostrou o melhor em comparação com os outros modelos experimentados neste trabalho, conseguindo resultados promissores de acurácia 0,961, precisão 0,950, revocação 0,988, *F1-score* 0,968 e AUC 0,987. Isto demonstra que este modelo pode atuar bem para problemas de classificação de barragens a partir de imagens de satélites.

O modelo proposto obteve melhores resultados em comparação com o estudo apresentado por Ferreira *et al.* [4]. Apesar de que as metodologias e abordagens de avaliação foram diferentes do estudo comparado. Como limitações do presente trabalho, devido à falta de recursos computacionais pessoais não foi possível utilizar a técnica de *k-fold Cross-Validation* que foi a técnica mais utilizada na literatura e que também é interessante para avaliar base de dados pequenas [51]. Além disso, não foram feitos trabalhos para ajustes de hiper-parâmetros uma vez que isto poderia incrementar os resultados obtidos, sendo utilizados os valores padrões das bibliotecas. Por fim, os modelos utilizados foram treinados e testados apenas na base de dados *BrazilDAM* [4], não sendo possível ter uma visão mais ampla do comportamento do modelo em outras bases de dados de imagens de barragens.

Capítulo 7

Conclusão e Trabalhos futuros

O presente trabalho apresentou uma associação de diversas técnicas como *transfer learning*, *fine tuning*, modelo híbrido CNN-SVM e *ensemble*, que ao serem combinadas culminaram em um modelo que mostrou resultados bastante positivos. Este modelo *ensemble* apresentado, por sua vez, obteve os melhores resultados em comparação aos modelos tradicionais e híbridos. Com acurácia 0,961, precisão 0,950, revocação 0,988, F1-score 0,968 e AUC 0,987. Sendo assim, *ensemble* de modelos híbridos para a tarefa de classificação de barragens em imagens de satélite se mostrou bastante promissor.

Mesmo com os bons resultados apresentados com certeza existem melhorias capazes de elevar ainda mais o potencial do modelo apresentado. Como por exemplo, o uso de outras técnicas de *ensemble*: Bagging, Boosting ou Stacking [49] que são abordagens diferentes das utilizadas no presente trabalho. Outro ponto possível de teste seria o uso de outras combinações de bandas de cores dado que o *Sentinel-2*¹ possui 13 bandas espectrais. Para testar todas estas combinações possíveis resultariam em 286 diferentes versões da base no qual, para a tarefa aqui avaliada, alguma destas combinações poderia influenciar o resultado. Além disso, realizar um estudo mais profundo em relação aos parâmetros e hiper-parâmetros utilizados, tanto na SVM como nas CNNs. Por fim, o modelo proposto poderia ter seu desempenho avaliado para realização de outras classificações aplicadas para outros problemas utilizando imagens, como ocorre no trabalho [12] que também utiliza um *ensemble* de modelos híbridos para detecção de pragas em plantas por meio de imagens.

¹<https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric>

Referências

- [1] Geron, Aurelien: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2nd edição, 2019, ISBN 1492032646. ix, 3, 4, 6
- [2] Ippolito, Pier Paolo: *Svm: Feature selection and kernels*, Sep 2021. <https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c>, acesso em 28/04/2022. ix, 6
- [3] Malgonde, Subodh: *Transfer learning using tensorflow*, May 2018. <https://medium.com/@subodh.malgonde/transfer-learning-using-tensorflow-52a4f6bcde3e>, acesso em: 28/04/2022. ix, 8
- [4] Ferreira, E., Brito M. Alvim M. S. Balaniuk R. dos Santos J.: *Brazildam: A benchmark dataset for tailings dam detection*. Em *Latin American GRSS ISPRS Remote Sensing Conference 2020, Santiago, Chile*. IEEE, 2020. ix, 1, 2, 9, 12, 14, 15, 30
- [5] Nogare, Diego: *Performance de machine learning - matriz de confusão*, Apr 2020. <https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/>. ix, 22
- [6] Luz, Adão Benvindo da, João Alves Sampaio e Silvia Cristina Alves França: *Tratamento de minérios*. CETEM/MCT, 2010. 1
- [7] Pereira, Dulce Maria, Hellen Oscarina Ramos Guimarães, Suzy Magaly Cabral de Freitas e André Alckmin Magalhães Mngia: *Brumadinho : Muito mais que um desastre tecnológico*. Revista da Universidade Federal de Minas Gerais, 27(2):332–355, out. 2021. <https://periodicos.ufmg.br/index.php/revistadaufmg/article/view/21649>. 1, 9
- [8] Tokarnia, Mariana: *Quase metade das barragens no brasil funcionam de forma irregular*, abril 2018. <https://agenciabrasil.ebc.com.br/geral/noticia/2018-04/especialistas-debatem-seguranca-e-fiscalizacao-de-barragens-no-pais>, acesso em: 28/04/2022. 1, 9
- [9] Balaniuk, Remis, Olga Isupova e Steven Reece: *Mining and tailings dam detection in satellite imagery using deep learning*. *Sensors*, 20(23), 2020, ISSN 1424-8220. <https://www.mdpi.com/1424-8220/20/23/6936>. 1, 2, 10, 12

- [10] Rizwan I Haque, Intisar e Jeremiah Neubert: *Deep learning approaches to biomedical image segmentation*. Informatics in Medicine Unlocked, 18:100297, 2020, ISSN 2352-9148. <https://www.sciencedirect.com/science/article/pii/S235291481930214X>. 1
- [11] Tao, Tao e Xinhua Wei: *A hybrid CNN–SVM classifier for weed recognition in winter rape field*. Plant Methods, 18, março 2022. 1, 10, 12, 18
- [12] Turkoglu, Muammer, Berrin Yanikoğlu e Davut Hanbay: *Plantdiseasenet: convolutional neural network ensemble for plant disease and pest detection*. Signal, Image and Video Processing, 16(2):301–309, Mar 2022, ISSN 1863-1711. <https://doi.org/10.1007/s11760-021-01909-2>. 1, 11, 31
- [13] Li, Qing, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng e Mei Chen: *Medical image classification with convolutional neural network*. Em *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, páginas 844–848, 2014. 3
- [14] Guo, Tianmei, Jiwen Dong, Henjian Li e Yunxing Gao: *Simple convolutional neural network on image classification*. Em *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, páginas 721–724, 2017. 3
- [15] Fukushima, Kuniyuki e Sei Miyake: *Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition*. Em Amari, Shun ichi e Michael A. Arbib (editores): *Competition and Cooperation in Neural Nets*, páginas 267–285, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg, ISBN 978-3-642-46466-9. 3
- [16] Hubel, D. H. e T. N. Wiesel: *Receptive fields and functional architecture of monkey striate cortex*. The Journal of Physiology, 195(1):215–243, 1968. <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1968.sp008455>. 3
- [17] Lecun, Y., L. Bottou, Y. Bengio e P. Haffner: *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11):2278–2324, 1998. 3
- [18] Agarap, Abien Fred: *Deep learning using Rectified Linear Units (ReLU)*. CoRR, abs/1803.08375, 2018. <http://arxiv.org/abs/1803.08375>. 4
- [19] Sharma, Sagar, Simone Sharma e Anidhya Athaiya: *Activation functions in neural networks*. towards data science, 6(12):310–316, 2017. 5
- [20] Rasamoelina, Andrinandrasana David, Fouzia Adjailia e Peter Sinčák: *A review of activation function for artificial neural network*. Em *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, páginas 281–286. IEEE, 2020. 5
- [21] Adaloglou, Nikolas: *Best deep CNN architectures and their principles: From alexnet to efficientnet*, Jan 2021. <https://theaisummer.com/cnn-architectures/>, acesso em 28/04/2022. 5

- [22] Liu, Shuying e Weihong Deng: *Very deep convolutional neural network based image classification using small training sample size*. Em *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, páginas 730–734, 2015. 5, 9, 17
- [23] Krizhevsky, Alex: *One weird trick for parallelizing convolutional neural networks*. CoRR, abs/1404.5997, 2014. <http://arxiv.org/abs/1404.5997>. 5, 9
- [24] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens e Zbigniew Wojna: *Rethinking the inception architecture for computer vision*. CoRR, abs/1512.00567, 2015. <http://arxiv.org/abs/1512.00567>. 5, 9, 17
- [25] Huang, Gao, Zhuang Liu e Kilian Q. Weinberger: *Densely connected convolutional networks*. CoRR, abs/1608.06993, 2016. <http://arxiv.org/abs/1608.06993>. 5, 9, 17
- [26] Hochreiter, Sepp: *The vanishing gradient problem during learning recurrent neural nets and problem solutions*. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, abril 1998. 5
- [27] Vapnik, Vladimir N.: *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995, ISBN 0387945598. 6
- [28] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay: *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 6, 18, 28
- [29] Pan, Sinno Jialin e Qiang Yang: *A survey on transfer learning*. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 7, 17
- [30] Weiss, Karl, Taghi M. Khoshgoftaar e DingDing Wang: *A survey of transfer learning*. *Journal of Big Data*, 3(1):9, May 2016, ISSN 2196-1115. <https://doi.org/10.1186/s40537-016-0043-6>. 7
- [31] Peng, Peng e Jiugen Wang: *How to fine-tune deep neural networks in few-shot learning?* CoRR, abs/2012.00204, 2020. <https://arxiv.org/abs/2012.00204>. 7, 17
- [32] Iandola, Forrest N., Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally e Kurt Keutzer: *Squeezenet: Alexnet-level accuracy with 50X fewer parameters and <1mb model size*. CoRR, abs/1602.07360, 2016. <http://arxiv.org/abs/1602.07360>. 9
- [33] He, Kaiming, Xiangyu Zhang, Shaoqing Ren e Jian Sun: *Deep residual learning for image recognition*. CoRR, abs/1512.03385, 2015. <http://arxiv.org/abs/1512.03385>. 9
- [34] Kingma, Diederik P. e Jimmy Ba: *Adam: A method for stochastic optimization*. *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*., 2015. <https://arxiv.org/abs/1412.6980>. 9, 17

- [35] Yang, Jun, Yeqing Sun, Qing Li e Yixuan Sun: *Effective risk prediction of tailings ponds using machine learning*. Em *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, páginas 234–238, 2020. 10, 12
- [36] Lyu, Jianjun, Ying Hu, Shuliang Ren, Yao Yao, Dan Ding, Qingfeng Guan e Liufeng Tao: *Extracting the tailings ponds from high spatial resolution remote sensing images by integrating a deep learning-based model*. *Remote Sensing*, 13(4), 2021, ISSN 2072-4292. <https://www.mdpi.com/2072-4292/13/4/743>. 10, 11, 12
- [37] Bochkovskiy, Alexey, Chien-Yao Wang e Hong-Yuan Mark Liao: *Yolov4: Optimal speed and accuracy of object detection*. *CoRR*, abs/2004.10934, 2020. <https://arxiv.org/abs/2004.10934>. 10
- [38] Ahlawat, Savita e Amit Choudhary: *Hybrid CNN-SVM classifier for handwritten digit recognition*. *Procedia Computer Science*, 167:2554–2560, 2020, ISSN 1877-0509. <https://www.sciencedirect.com/science/article/pii/S1877050920307754>, International Conference on Computational Intelligence and Data Science. 11, 12, 18
- [39] Deng, Li: *The mnist database of handwritten digit images for machine learning research*. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 11
- [40] Yuan, Kunhao, Xu Zhuang, Gerald Schaefer, Jianxin Feng, Lin Guan e Hui Fang: *Deep-learning-based multispectral satellite image segmentation for water body detection*. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7422–7434, 2021. 14
- [41] Wiggins, Richard H., H. Christian Davidson, H. Ric Harnsberger, Jason R. Lauman e Patricia A. Goede: *Image file formats: Past, present, and future*. *Radiographics*, 21(3):789–798, 2001. <https://doi.org/10.1148/radiographics.21.3.g01ma25789>, PMID: 11353125. 14
- [42] GISGeography: *Sentinel 2 bands and combinations*, Oct 2021. <https://gisgeography.com/sentinel-2-bands-combinations/>, acesso em: 28/04/2022. 16
- [43] Hub, Sentinel: *Sentinel playground*. <https://www.sentinel-hub.com/explore/sentinelplayground/>, acesso em: 28/04/2022. 16
- [44] Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu e Xiaoqiang Zheng: *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. <https://www.tensorflow.org/>, Software available from tensorflow.org. 17, 28

- [45] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg e Li Fei-Fei: *Imagenet large scale visual recognition challenge*. International Journal of Computer Vision, 115, setembro 2014. 17
- [46] Tieleman, Tijmen, Geoffrey Hinton *et al.*: *Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural networks for machine learning, 4(2):26–31, 2012. 17
- [47] Ruby, Usha e Vamsidhar Yendapalli: *Binary cross entropy with deep learning technique for image classification*. International Journal of Advanced Trends in Computer Science and Engineering, 9, outubro 2020. 17
- [48] Jena, Biswajit, Sanjay Saxena, Gopal K. Nayak, Luca Saba, Neeraj Sharma e Jasjit S. Suri: *Artificial intelligence-based hybrid deep learning models for image classification: The first narrative review*. Computers in Biology and Medicine, 137:104803, 2021, ISSN 0010-4825. <https://www.sciencedirect.com/science/article/pii/S0010482521005977>. 18
- [49] Zhang, Cha e Yunqian Ma: *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012, ISBN 1441993258. 18, 31
- [50] Hossin, Mohammad e Sulaiman M.N: *A review on evaluation metrics for data classification evaluations*. International Journal of Data Mining Knowledge Management Process, 5:01–11, março 2015. 21, 22
- [51] Refaeilzadeh, Payam, Lei Tang e Huan Liu: *Cross-Validation*, páginas 532–538. Springer US, Boston, MA, 2009, ISBN 978-0-387-39940-9. https://doi.org/10.1007/978-0-387-39940-9_565. 30