How many Convolutional Layers are required for a Granite Classification Neural Network?

Eduardo Henrique Próspero Souza[1], **Karin Satie Komati**[2]

[1] Coordenação de Sistemas de Informação

[2]Programa de Pós-graduação em Computação Aplicada (PPComp)
Instituto Federal do Espírito Santo (IFES)
Av. dos Sabiás, 330 – 29.166-630 – Serra – ES – Brasil

duvrdx@gmail.com, kkomati@ifes.edu.br,

Abstract. *The purpose of this article is to analyze the result of the classification accuracy of types of granite, according to the variation in the number of convolutional layers of a CNN (Convolutional Neural Network). From a standard CNN architecture, only the number of convolutional layers is varied, starting with 2 layers up to 10 layers, increasing one layer for each experiment. A public database, Rock Image Datasets, was used. In the end, the architecture with 5 convolutional layers was the one that achieved the best results, reaching 99% accuracy.*

Resumo. *O objetivo deste artigo é analisar o resultado da acurácia de classificação de tipos de granito, de acordo com a variação da quantidade de camadas convolucionais de uma CNN. A partir de uma arquitetura padrão de CNN, varia-se apenas a quantidade de camadas convolucionais, iniciando com 2 camadas até 10 camadas, incrementando uma camada a cada experimento. Foi usada uma base de dados pública, a Rock Image Datasets. Ao final, a arquitetura com 5 camadas convolucionais foi a que alcançou os melhores resultados, chegando a 99% de acurácia.*

1. Introdução

A classificação automática de imagens consiste em atribuir uma classe à uma imagem de entrada [Khan et al. 2020]. Várias são as aplicações de classificação de imagens: classificação de escrita à mão, reconhecimento de faces, reconhecimento de emoções/expressões faciais, estimativa de pose (configuração da posição corporal), classificação de doenças via imagens de exames laboratoriais, dentre outros [Li et al. 2021]. Na área da geologia, a classificação automática de rochas permitiria a classificação do tipo de rocha por um aplicativo de celular, por um leigo, sem a necessidade de se levar amostras à um laboratório. Em especial, na vertente de rochas ornamentais, mármores e granitos, o preço da rocha varia de acordo com a sua classificação, que consiste nas características de coloração e textura [Bianconi et al. 2012].

As redes neurais convolucionais, ou simplesmente CNN's (do inglês, Convolutional Neural Network), são ferramentas que podem ser utilizadas para diversas tarefas, como previsão e processamento de linguagem natural, entre varias áreas. Uma área onde comumente são utilizadas é na classificação e detecção de imagens, onde este trabalho se focará[O'Shea and Nash 2015].

No entanto, há muitas possibilidades de arquiteturas de CNN [Bhatt et al. 2021], o que dificulta a escolha de qual delas usar para uma determinada aplicação. Por exemplo, o trabalho de [Pascual et al. 2019] faz a classificação de imagens de rochas em 9 classes usando uma CNN de 3 camadas, que alcançou uma precisão média de 99,60%. A mesma pesquisa abordou um problema de classificação binária onde as imagens são *breccia*[1] e não-*breccia*, mas usou uma CNN de 5 camadas, que atingiu 89,43% de precisão de classificação.

O artigo de [Ren et al. 2020] discute a questão de que o *design* da arquitetura neural é crucial para a representação das características dos dados e o seu desempenho final, no entanto, o *design* da arquitetura neural depende muito do conhecimento e da experiência prévia de quem o usa, sendo um obstáculo para se projetar um modelo ideal. Assim, a proposta deste trabalho é investigar variações de uma arquitetura de CNN para o problema de classificação de granito. Com isso, tentando responder à pergunta: qual é a menor quantidade de camadas convolucionais numa CNN que resulte na maior acurácia?

Para o estudo, a partir de uma arquitetura padrão de CNN, varia-se apenas a quantidade de camadas convolucionais, iniciando com 2 camadas até 10 camadas, incrementando uma camada a cada experimento. Utilizaremos o valor de acurácia e medida F1 como métricas para avaliar os modelos e a base de dados Rock Image Datasets, que é pública. Ao final será traçado um gráfico de acurácia por quantidade de camadas convolucionais, que permitirá a análise da CNN na tarefa de classificação de rochas.

Esse artigo está organizado da seguinte maneira: na Seção 2 são apresentado os trabalhos relacionados, Seção 3 descreve os materiais e métodos utilizados ao decorrer do trabalho. Os resultados obtidos estão presentes na Seção 4. E por fim, a conclusão se encontra na Seção 5.

2. Trabalhos Correlatos

O artigo de [Shu et al. 2017] propõe duas abordagens para a extração de características na tarefa de classificação automática de rochas. A primeira é uma técnica de aprendizagem de características não supervisionada para extrair características para imagens de rochas, o K-means. O segundo método autônomo de seleção de características é chamado de aprendizagem autodidata. O método base de comparação é a extração por seleção de características manuais. As características extraídas são a entrada para o método de classificação SVM (Suporte à Máquina de Vetor). Para os experimentos foi utilizada base de dados "Rock Image Datasets", chegando a uma acurácia de 96,71% com a abordagem baseada em K-means.

Alexis Pascual [Pascual et al. 2019] utiliza a mesma base de dados do trabalho de [Shu et al. 2017], a fim de treinar uma rede neural convolucional para classificação de rochas. A CNN de 3 camadas foi utilizada no desenvolvimento de um aplicativo para celular. Ao fim do trabalho, a CNN obteve resultados de 99,6% de acurácia, superando os resultados de [Shu et al. 2017].

O artigo de [Ran et al. 2019] propõe o uso de CNN de 2 camadas convolucio-

[1]Rocha sedimentar clástica, do grupo psefito, formada por fragmentos de material clástico com arestas vivas e de dimensões médias superiores a 2 mm, cimentadas por diferentes materiais, tasi como argilosos e calcários.

nais para classificação de rochas. A abordagem proposta pode classificar seis tipos de rochas comuns com uma precisão de classificação geral de 97,96%. A grande diferença deste trabalho é que as imagens de teste são cenários em campo aberto, em que há vários elementos, e podem conter diferentes tipos de rochas.

O trabalho de [Xu et al. 2022] faz a classificação, mas usando imagens microscópicas da rocha. Foram comparadas sete redes neurais convolucionais diferentes, Xception, MobileNet_v2, Inception_ResNet_v2, Inception_v3, Densenet121, ResNet101_v2 e ResNet-101, usando a técnica de *transfer learning*. Os resultados mostram que o modelo baseado em Xception obteve o melhor desempenho, com uma precisão de 97,66% no conjunto de dados de treinamento e 98,65% no conjunto de dados de teste.

3. Materiais, Métodos e Métricas

Nesta seção detalha-se a base de dados Rock Image Datasets, a arquitetura da rede neural que teve variação na quantidade de camadas convolucionais, as métricas usadas e configurações de treinamento.

3.1. Rock Image Datasets

A base de dados utilizada, a Rock Image Datasets foi disponibilizado por Alexis Pascual no site Mendeley Data[2]. A base possui 711 imagens com 9 classes de rocha. Cada imagem tem 128 pixels de largura, 128 pixels de altura. Essa base de dados foi a mesma utilizada nos artigos de [Pascual et al. 2019] e [Shu et al. 2017]. A Figura 1 apresenta um exemplo de cada uma das 9 classes de rochas ornamentais presentes na base de dados.

Figura 1. Amostras das classes presentes no Rok Image Datasets. De cima para baixo, a primeira coluna - rhyolite, volcanic breccia, limestone; a segunda coluna – granite, andesite, oolitic limestone; na terceira coluna – red granite, peridotite, dolostone.

[2]https://data.mendeley.com/datasets/7g7zpy9vcb

A base de dados é disponibilizada como uma matriz em um arquivo com a extensão ".mat", do Matlab©. A função de leitura deste formato e armazenamento em formato de imagem foi feito utilizando a linguagem Python e as bibliotecas Numpy e OpenCV.

3.2. Arquitetura da Rede Neural Convolucional

As CNNs são compostas por várias camadas de processamento de imagens que realizam tarefas a fim de detectar características (etapa de extração de características, em inglês *feacture extraction*) e em sequência classificar de acordo com as saídas esperadas (etapa de classificação). A Figura 2 apresenta uma arquitetura básica, a imagem de entrada é uma imagem de granito. As camadas de convolução (apresentados como quadrados azuis e brancos intercalados) e *pooling* (apresentados amarelos e brancos intercalados) fazem a extração de características. No exemplo da figura, este conjunto de duas camadas foi repetido duas vezes em sequência. Ao final, tem-se a classificação da imagem de entrada feita pela camada completamente conectada (apresentado como círculos verdes conectados) com uma última camada de saída com 9 elementos, pois são 9 classes.

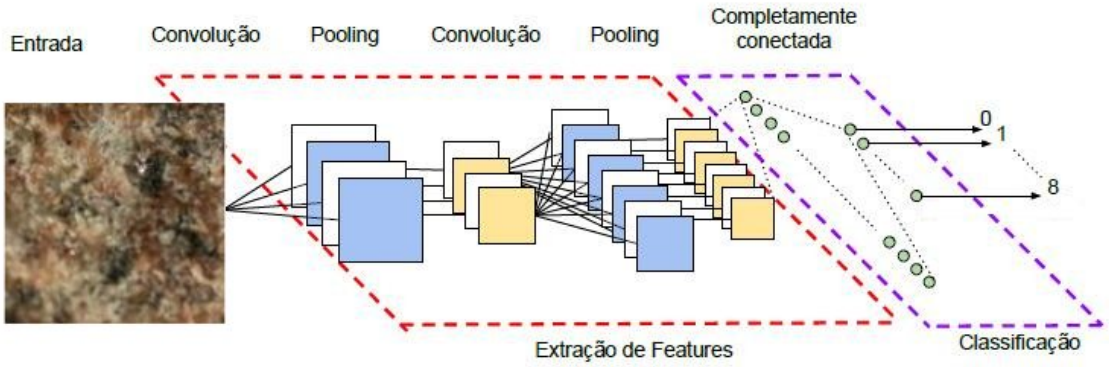

Figura 2. Arquitetura básica de uma CNN.

Para problemas de classificação de imagem, usualmente a camada convolucional recebe uma matriz de três dimensões, com altura, largura e profundidade. Na camada convolucional, cada imagem de entrada é convolucionada com vários *kernels* de tamanho predefinido. Uma convolução é simplesmente uma soma ponderada dos valores presentes na matriz da imagem dentro de uma janela deslizante. Cada *kernel* gera mapas de características das imagens. O algoritmo de treinamento aprende quais características devem ser extraídos da imagem ajustando os pesos dos *kernels* por meio de *back propagation*.

Após a convolução, uma função de ativação é aplicada ao mapa de características produzido para introduzir não linearidade na tarefa em questão. Neste artigo, foi usada a função de ativação ReLU (do inglês Rectified Linear Units, em português unidade linear retificada). É uma função que retorna zero para qualquer valor negativo, e o próprio número para números positivos.

Uma camada de *pooling* é comumente adicionada após uma camada de convolução para reduzir o tamanho do mapa de características e adicionar invariância posicional aos pixels com as respostas máximas às convoluções. Para esta pesquisa, o

valor máximo de pixel (*max pooling*) é escolhido dentro da janela e é então mapeado para uma única posição de pixel no mapa de características resultante.

Após uma série de convoluções e *pooling*, os mapas de características resultantes são então achatados (*flattened*) para produzir um vetor 1-D a partir dos mapas 2-D. O vetor 1-D resultante é a entrada de uma rede neural artificial (RNA) para classificação. As camadas totalmente/completamente conectadas são as camadas ocultas dentro da RNA. Para a camada de saída, foi escolhida a função de ativação Softmax, que retorna uma probabilidade de que um vetor de entrada pertença a uma determinada classe. A classe com a maior probabilidade é a considerada como resultado.

As CNN's foram desenvolvidas utilizando o Tensorflow, biblioteca da linguagem de programação Python. Foram desenvolvidos 9 modelos diferentes que possuem de duas a dez camadas de convolução, aumentando uma camada a cada modelo. O código desenvolvido pode ser encontrado no GitHub[3].

Os otimizadores, funcionam no intuito de buscar os melhores parâmetros, visando minimizar a função de perda usada. Como otimizador, foi escolhido o Adam, que é uma versão modificada e melhorada do SGD (do inglês Stochastic Gradient Descent), a descida do gradiente, ambos algoritmos tem o objetivo de encontrar as entradas que irão minimizar uma determinada função, esses algoritmos realizam essa busca de forma iterativa. Os demais parâmetros das funções de ativação, funções de perda e otimizadores foram mantidos nos padrões oferecidos pelo TensorFlow

3.3. Métricas

Para avaliar a eficiência dos modelos, foram usadas a matriz de confusão, e as métricas de acurácia e a medida F1. A matriz de confusão é uma matriz com as frequências de classificação de cada modelo. A Tabela 1 é um exemplo de matriz de confusão, onde estão indicadas 3 classes: A, B e C.

Tabela 1. Matriz de confusão da predição de três classes: A, B e C.

Matriz de Confusão 3x3		Classificada ou Estimada			
		A	B	C	
Real	A	30	50	20	100
	B	20	60	20	100
	C	10	10	80	100
		60	120	120	

Os valores representados na diagonal da matriz são as classificações realizadas corretamente, no entanto, os demais valores mostram as quantidades incorretas de classificações. Ainda, ao somar os valores das colunas obtém-se a quantidade de classificações realizadas pelo sistema para a classe em questão e ao somar os valores de uma linha obtém-se a quantidade total de representantes da classe original. No exemplo, há 100 (cem) elementos da Classe A, sendo que 30 foram classificados corretamente, mas 70 elementos A foram classificados incorretamente, 50 como B e 20 como C. Dos 100 elementos da Classe B, 60 foram classificados corretamente, 20 foram classificados

[3]https://github.com/duvrdx/ic_classificacao_rochas

incorretamente como A e 10 incorretamente como C. Dos 100 elementos da Classe C, 80 foram classificados corretamente, 10 classificados incorretamente como A e 10 classificados incorretamente como B.

A acurácia é a proporção entre as predições corretas e a soma das predições corretas e as predições erradas. Com a seguinte fórmula, sendo VP+VN (verdadeiro positivo + verdadeiro negativo), as predições corretas e FP+FN (falso positivo + falso negativo) as predições erradas [Labatut and Cherifi 2012]. No exemplo, o resultado é 56,6% (divisão de 170 por 300).

$$acuracia = \frac{vp + vn}{vp + vn + fp + fn}$$

A métrica de precisão pode ser vista como uma resposta à pergunta: "para uma determinada classe X, quantas instâncias foram previstas corretamente?". No exemplo, para a classe A seria 50% (resultado da divisão de 30 por 60), para a classe B seria 0,5 ou 50% (60/(60+60)), para a Classe C, 66,6% (80/(80+40)). A revocação é uma medida da integridade do classificador que vai medir a capacidade de encontrar corretamente todas as instâncias positivas, como uma resposta à pergunta: "para uma determinada classe X, quantos desta classe você encontra?". No exemplo, para a classe A seria 30% (resultado da divisão de 30 por 100), para a Classe B, 60% (60/(60+40)), e para a Classe C, 80% (80/(80+20)). A medida F1 é calculada como a média harmônica entre a precisão e a revocação.

$$F = 2\frac{precisao.revocacao}{precisao + revocacao}$$

3.4. Treinamento

Como a base de dados não possui uma grande quantidade de imagens, todos os modelos contam com uma fase de Data Augmentation, que serve para aumentar o número de amostras, criando variações das imagens que já existem. Foram usadas as mesmas operações do trabalho de [Pascual et al. 2019], apresentadas na Figura 3.

Para a validação cruzada, o dataset foi dividido em 70% para o treinamento dos modelos, e os outros 30% para os testes. A divisão do dataset foi a mesma utilizada no trabalho de [Pascual et al. 2021]. As configurações de treino foram iguais para todos os modelos, o *batch size* foi de 2, para aproveitar melhor as poucas imagens presentes no dataset. Foi utilizado o número de 1.000 épocas para que nossos modelos passassem mais tempos treinando. Diferente do trabalho de [Pascual et al. 2021], inicializamos todos os pesos dos modelos aleatoriamente, deixando com que o *framework* utilizado se encarregasse disso.

No treinamento foi possível notar a diferença entre as curvas dos gráficos de perda e acurácia. Onde os modelos entre duas e oito camadas tinham curvas similares, e os modelos de nove e dez camadas eram diferentes das demais. A Figura 4 apresenta o gráfico de acurácia e perda para a arquitetura com 5 camadas e a Figura 5, o gráfico para o modelo com 9 camadas.

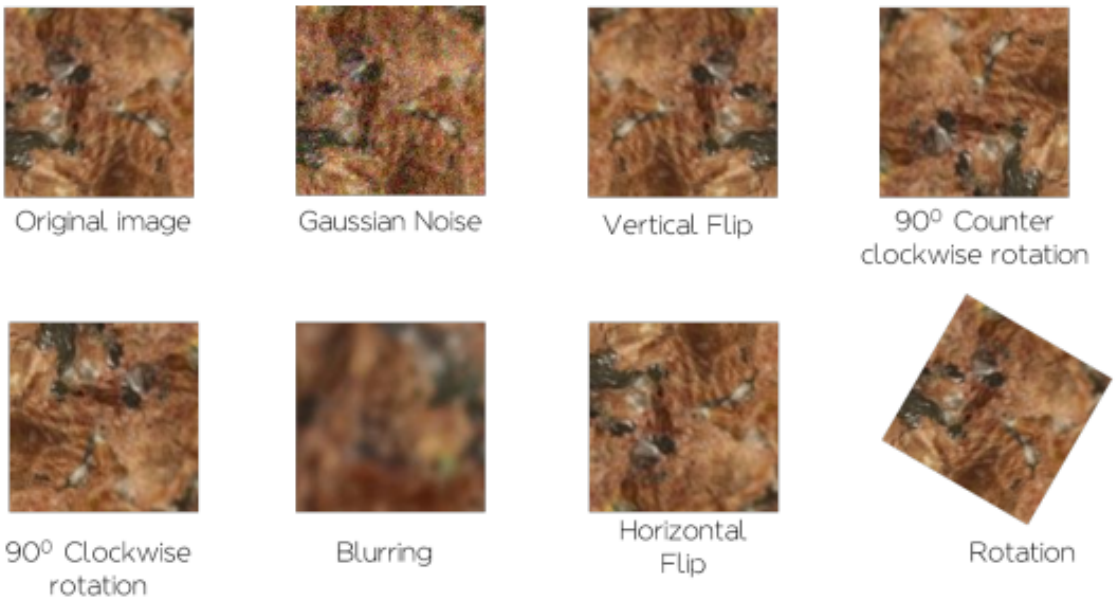

Figura 3. Operações de *data augmentation*.

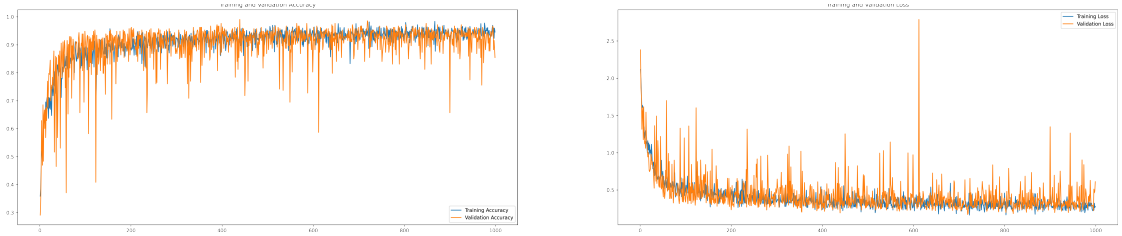

Figura 4. Gráfico de Acurácia e Perda do modelo com 5 camadas

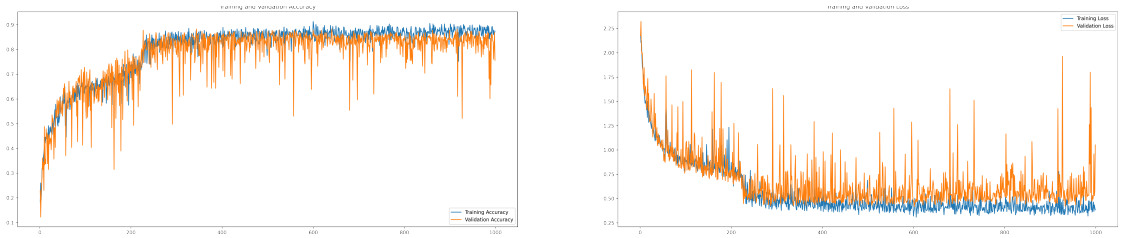

Figura 5. Gráfico de Acurácia e Perda do modelo com 9 camadas

4. Resultados e Discussão

Para a visualização dos resultados, foram elaborados dois gráficos (Figura 6), em que o eixo x é a quantidade de camadas convolucionais e no eixo y, são os valores de acurácia e de perda para cada modelo. O melhor resultado é quando há a menor perda possível, e a acurácia deve ser a maior possível. Os valores de acurácia dos modelos variam entre cerca de 0,88 e 0,99, e os valores de perda, entre cerca de 0,2 e 0,5. A maior acurácia é alcançada com 5 camadas de convolução (acurácia de 0,9906 e perda em 0,193), e o pior resultado é com 9 camadas.

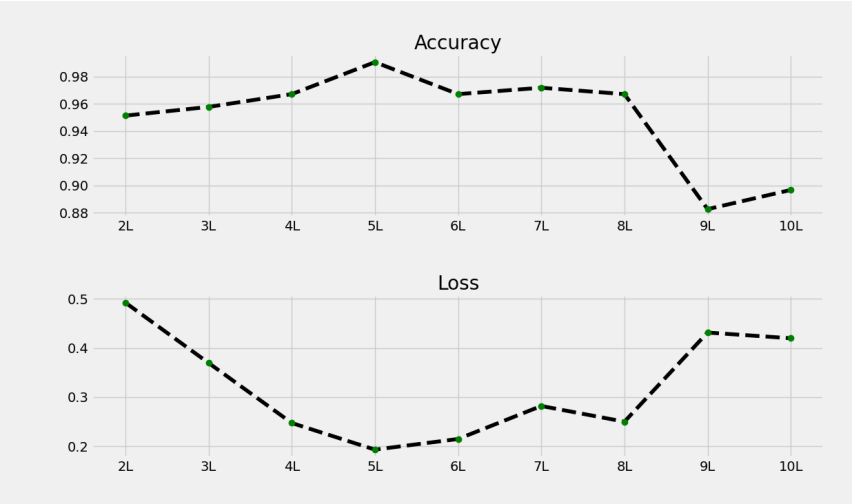

Figura 6. Acurácia e perda por quantidade de camadas

O relatório com métricas de precisão, revocação e Medida-F1 é apresentado na Figura 7.

Classification Report:

	precision	recall	f1-score	support
class 1	1.00	1.00	1.00	17
class 2	0.97	0.97	0.97	29
class 3	0.95	0.95	0.95	21
class 4	1.00	1.00	1.00	28
class 5	1.00	1.00	1.00	26
class 6	1.00	1.00	1.00	24
class 7	1.00	1.00	1.00	15
class 8	1.00	1.00	1.00	32
class 9	1.00	1.00	1.00	21
accuracy			0.99	213
macro avg	0.99	0.99	0.99	213
weighted avg	0.99	0.99	0.99	213

Figura 7. Resultados gerados

A matriz de confusão para o modelo com 5 camadas de convolução é apresentada na Figura 8. Analisando os dados da matriz de confusão, é perceptível que a maioria dos valores se concentraram na diagonal principal, isso quer dizer que o modelo acertou bem nos testes. O modelo realizou predições equivocadas apenas nas classes 2 e 3 (Figura 9 e Figura 10), assim, no relatório, 8 das 10 classes obtiveram 1.0 na medida F1.

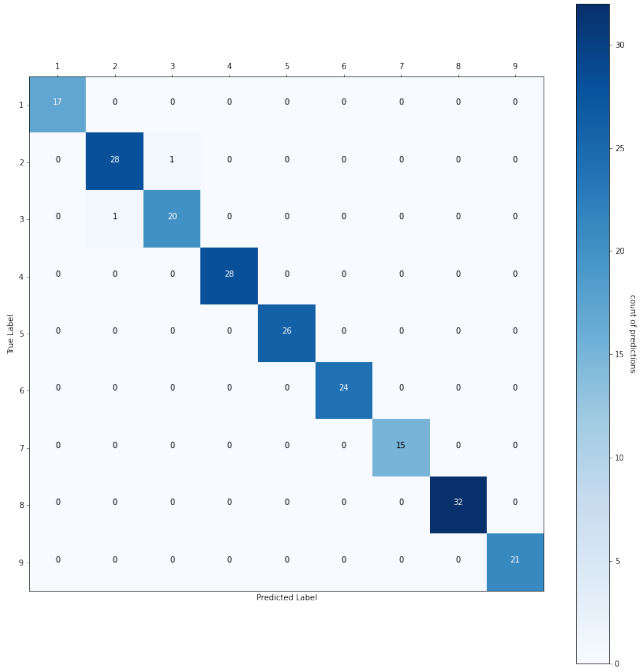

Figura 8. Matriz de confusão do modelo com 5 camadas

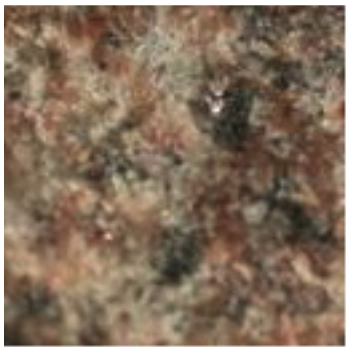

Figura 9. Imagem classificada incorretamente, classe correta: granite (2), mas foi classificada como red granite (3).

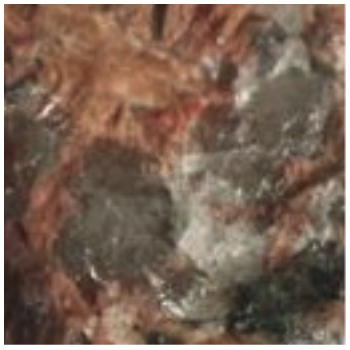

Figura 10. Imagem classificada incorretamente, classe correta: red granite (3), mas foi classificada como granite (2).

5. Conclusão

O objetivo de investigar variações de uma arquitetura de CNN, especificamente a quantidade de camadas de convolução, para o problema de classificação de granito foi alcançado. Foi possível gerar um gráfico com a acurácia pela quantidade de camadas. Ao aumentarmos o número de camadas (a partir de 2), podemos ver um aumento no valor da acurácia, até chegar ao valor máximo com 5 camadas. Segue-se com uma pequena queda e estabilização nos modelos com 6, 7 e 8 camadas, tendo uma queda muito significativa com 9 camadas, sendo até pior que o resultado com o modelo de 2 camadas.

Assim, constatamos que a discussão feita no artigo de [Ren et al. 2020], de que o *design* da arquitetura neural é crucial para a representação das características dos dados e o seu desempenho final. Como trabalhos futuros, pretendemos criar uma base de dados com mais amostras e mais classes e implementar técnicas de NAS (Neural Architecture Search) para encontrar a melhor arquitetura de CNN para o problema de classificação de granitos. E utilizarmos novas formas de avaliação como, *ablation studies*, que consiste na realização de testes removendo módulos, para avaliar o impacto de cada módulo no desempenho do sistema como um todo.

6. Agradecimentos

O primeiro autor agradece ao Ifes pela bolsa de iniciação científica. A profª Komati agradece ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pela Bolsa de Produtividade DT-2 (308432/2020-7) e à FAPES (Fundação de Amparo à Pesquisa e Inovação do Espírito Santo) pelo Auxílio Taxa de Pesquisa (nº 293/2021).

Referências

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., and Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, 10(20).

Bianconi, F., González, E., Fernández, A., and Saetta, S. A. (2012). Automatic classification of granite tiles through colour and texture features. *Expert Systems with Applications*, 39(12):11212–11218.

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53(8):5455–5516.

Labatut, V. and Cherifi, H. (2012). Accuracy measures for the comparison of classifiers. *arXiv preprint arXiv:1207.3790*.

Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Pascual, A. D., McIsaac, K. M., and Osinski, G. (2021). Deep learning of rock images for intelligent lithology identification. *Preprints*.

Pascual, A. D. P., Shu, L., Szoke-Sieswerda, J., McIsaac, K., and Osinski, G. (2019). Towards natural scene rock image classification with convolutional neural networks. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–4. IEEE.

Ran, X., Xue, L., Zhang, Y., Liu, Z., Sang, X., and He, J. (2019). Rock classification from field image patches analyzed using a deep convolutional neural network. *Mathematics*, 7(8):755.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., and Wang, X. (2020). A comprehensive survey of neural architecture search: Challenges and solutions. *arXiv preprint arXiv:2006.02903*.

Shu, L., McIsaac, K., Osinski, G. R., and Francis, R. (2017). Unsupervised feature learning for autonomous rock image classification. *Computers & Geosciences*, 106:10–17.

Xu, Z., Ma, W., Lin, P., and Hua, Y. (2022). Deep learning of rock microscopic images for intelligent lithology identification: Neural network comparison and selection. *Journal of Rock Mechanics and Geotechnical Engineering*, 14(4):1140–1152.