Multiscale Context Features for Geological Image Classification

Matheus V. Todescato[a], Luan F. Garcia[b], Dennis G. Balreira[c] and Joel L. Carbonera[d]
Institute of Informatics, UFRGS, Porto Alegre, Brazil

Keywords: Image Classification, Transfer Learning, Deep Learning, Geology.

Abstract: Dealing with image retrieval in corporate systems becomes challenging when the dataset is small and the images present features in multiple scales. In this paper, we propose the notion of multiscale context features, in order to decrease information loss and improve the classification of images in such scenarios. We propose a preprocessing approach that splits the image into a set of patches, computes their features using a pre-trained model, and computes the context feature representing the whole image as an aggregation of the features extracted from individual patches. Besides that, we apply this approach in different scales of the image, generating context features of different scales, and we aggregate them to generate a multiscale representation of the image, which is used as the classifier input. We evaluated our method in a geological images dataset and in a publicly available dataset. We evaluate our approach with three efficient pre-trained models as feature extractors. The experiments show that our approach achieves better results than the conventional approaches for this task.

1 INTRODUCTION

This work is part of a project whose goal is to develop an *image retrieval system* for the petroleum industry. Due to the enormous amount of data in current corporate databases, managing and retrieving the relevant data for supporting the tasks of interest becomes a challenge. In this context, dealing with images is an even more challenging task. The main reason is the absence of explicit meaning associated with images, hindering the retrieval of this kind of data through conventional search queries.

A common approach for dealing with this scenario involves annotating images with semantic tags to allow searching and retrieving them (Hollink et al., 2003). However, manual annotation is laborious and not feasible in big data. Thus, automatic approaches for image classification could be of great value for automatically labeling images in this scenario (Wong and Leung, 2008; Zhang et al., 2012), making possible a subsequent retrieval of this data through conventional queries. Deep learning techniques are natural candidates for automatically labeling large databases of images.

In the last decade, Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012; Szegedy et al., 2015; Tan and Le, 2019) and, most recently, Vision Transformers (ViTs) (Dosovitskiy et al., 2020) significantly improved the performance on image classification tasks. Researchers have been using these techniques in several distinct domains (Sladojevic et al., 2016; Dung et al., 2019; Abbas et al., 2021; Hong et al., 2020). Despite these great results, sophisticated neural network architectures generally demand significant amounts of training data to achieve great performances (Zhu et al., 2021). Transfer learning (Torrey and Shavlik, 2010) has emerged in this context as a promising approach to deal with this problem (Liang and Zheng, 2020; Horry et al., 2020), since these approaches take advantage of the knowledge learned from bigger amounts of data for dealing with tasks in which only small data is available.

In general, when transfer learning is applied, images used for feeding the pre-trained neural networks are standardized to match the input requirements of the architectures (LeCun et al., 1998; Krizhevsky et al., 2009). However, datasets of images with different sizes and aspect ratios are very common in real-world settings. Thus, this standardization process can discard critical parts of the image or change its aspect ratio, causing information loss or introducing noise. Some approaches address these issues by applying,

[a] https://orcid.org/0000-0001-7568-8784
[b] https://orcid.org/0000-0001-9328-9007
[c] https://orcid.org/0000-0002-0801-9393
[d] https://orcid.org/0000-0002-4499-3601

Todescato, M., Garcia, L., Balreira, D. and Carbonera, J.
Multiscale Context Features for Geological Image Classification.
DOI: 10.5220/0011981100003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 1*, pages 407-418
ISBN: 978-989-758-648-4; ISSN: 2184-4992

for instance, patch-based architectures (Anthimopoulos et al., 2016; Araújo et al., 2017) or *Fully Convolutional Network* (FCN) (Wang et al., 2021; Zhuang et al., 2021). In this work, we investigate the hypothesis that the aggregation of features extracted from image patches by pre-trained models (called *context features*) can better represent the global information of the images. We also analyze whether combining context features of different scales can improve the classification quality in image datasets containing images of varying sizes representing relevant features in different scales.

In the Petroleum Geology domain, which is the focus of this work, the datasets are generally reasonably small, and the images commonly have heterogeneous sizes and visual features that are apparent in different scales. Thus, this paper focuses on presenting an approach based on *multiscale context features* for geological image classification that deals with such challenges. Our approach involves three steps: (i) firstly, we extract *context features* in different scales of each image using transfer learning from pre-trained models; (ii) next, we aggregate the context features of each scale to generate a *multiscale* representation of the image information; and (iii) finally, we use the resulting *multiscale context features* that represents each image as input for training a classifier.

We evaluated our method in a corporate dataset of geological images, which is the main focus of this work. Additionally, we also evaluated our approach in a publicly available dataset, in order to promote the reproducibility of our results and demonstrate our approach's capabilities to deal with images of other domains. Both datasets include highly varied images that challenge traditional approaches, including images with heterogeneous sizes and that present some features in different scales. We measure our algorithm's performance using three different pre-trained models as feature extractors: DenseNet (Huang et al., 2017), ResNeXt (Xie et al., 2017)) and CLIP (Radford et al., 2021). We compare our method with two well-established approaches for dealing with images of different sizes: (i) normalizing the image to the architecture input requirements by resizing and performing a center crop and (ii) applying a pre-trained model for extracting the features of the whole image without discarding information. Our experiments suggest that the proposed approach outperforms both considered alternatives. Furthermore, our work demonstrates that our approach achieves better performance by using the CLIP pre-trained model as a feature extractor. We also confirmed our results with a public dataset (Krause et al., 2013), obtaining similar results.

The remainder of this paper is structured as the following. Section 2 presents the related work. We detail our proposed approach in Section 3. Section 4 presents our experiments. Finally, Section 5 presents the conclusions.

2 RELATED WORK

In general, sophisticated deep learning models have a considerable number of trainable parameters, which demand enormous amounts of training data to achieve a good performance (Zhu et al., 2021). In contexts focused on small datasets, it is a common approach nowadays to apply Transfer Learning (TL) in several tasks and domains (Torrey and Shavlik, 2010; Zhuang et al., 2020), being widely used in image classification (Kim et al., 2022; Celik et al., 2020). Convolutional Neural Networks (CNN) such as ResNeXt (Xie et al., 2017) and DenseNet (Huang et al., 2017) have been achieving great performances when used for transfer learning, both as being used as pre-trained models for *feature extraction*, as well as for end-to-end classification with *fine-tuning* (Abou Baker et al., 2022; Varshni et al., 2019). In (Kieffer et al., 2017; Mormont et al., 2018), the authors compare the performance achieved with feature extraction and fine-tuning for classifying histopathology images. The experiments demonstrate that fine-tuning achieves higher performance while feature extraction achieves a good performance requiring fewer computational resources for training. Our approach benefits from using transfer learning for feature extraction, allowing our approach to achieve better classification performance in small datasets and to reduce the training time.

Usually, in image classification tasks, images are standardized to a homogeneous fixed size (LeCun et al., 1998; Krizhevsky et al., 2009) for matching the input requirements of the adopted neural network architectures. For example, a common approach involves resizing the smallest dimension of each image to match the input dimensions of the model and, after that, performing a center crop of the image, discarding information. This process can result in the loss of relevant information from the images or insertion of noise in the dataset (Tayal et al., 2021; Han et al., 2020; Li et al., 2021). On the other hand, in contrast to conventional CNN, some works (Wang et al., 2021; Zhuang et al., 2021) adopt *fully convolutional networks* (FCN), which do not require the input images to have a homogeneous size. As we will show later, the datasets used in this work are highly heterogeneous in size. For this reason, we designed our approach to preserve as much information as possible from the original image.

Some studies consider different spatial scales in order to classify images. In (Mohan and Venkatesan, 2020), for example, the authors propose using a multiscale spatio-spectral feature-based hybrid CNN model for hyperspectral image classification, which adopts different window sizes in 3D convolution filters. In general, in these works, different scales are related to distinct, independent convolution layers with various kernels later combined. In general, tasks involving small training datasets can be challenging for this kind of strategy.

In the medical field, where images tend to be large and with scattered details (X-ray, CT images), several works use patch-based approaches to consider all the image information. These approaches split the images, so it is possible to analyze them by considering their different patches (Anthimopoulos et al., 2016; Araújo et al., 2017). To deal with CT images for classifying cases of COVID-19 and pneumonia, for example, (Xu et al., 2020) use attention techniques to select areas and use them as patches to perform classification only in essential areas. In (Barstugan et al., 2020), the authors present an approach that preprocesses the image (losing information) and splits it into patches of different sizes, training just one classifier with these different scales. An example presented in (Chen et al., 2021) showed remarkable results using multiscale patch sizes to train a vision transformer, performing a fusion of features extracted at different scales. Our approach differs from existing works because, in order to create the context features, we use overlapping patches with a small stride while keeping the image's original aspect ratio, minimizing information loss. Besides that, once each patch is extracted, we apply transfer learning to extract features of each patch and aggregate the resulting features for representing the whole image. This strategy makes our approach suitable for dealing with small datasets.

3 PROPOSED APPROACH

This section presents our approach. Firstly, we discuss our method for extracting the image's *context features* in detail. After, we discuss our technique for generating a multiscale representation of images using *multiscale context features*.

3.1 Extracting Context Features

A challenging task when dealing with images of varying sizes in deep learning models is to preserve the image information while avoiding the inclusion of distortions and simultaneously matching the input requirements of the neural network architectures adopted in the task. Our approach overcomes these challenges by creating *context features* that represent the global image information by aggregating features extracted with pre-trained models of different parts (patches) of the image. We represent this process in Algorithm 1. Our algorithm takes as input the target image I, a pre-trained model FE used as a *feature extractor*, the value $N \in \mathbb{Z}^*$ that controls the scale of analysis, and the stride $S \in (0,1]$ of the sliding window, which is a percentage of the input size of FE. The first step of the algorithm is resizing image I according to the parameter N. We resize the image's smallest dimension to N times the input size of FE, and we resize the other image dimension accordingly so that the resulting image does not lose its original aspect ratio. For example, if the input size of FE is 224×224, by adopting $N = 1$, the image's smallest dimension would be 224 pixels after this resizing process.

Algorithm 1: Extraction of context features.

```
Input: Image as $I$, pre-trained model as $FE$, the approach parameter as $N$
       and the stride of the sliding window as $S$;
Output: A context features vector $V$;
begin
    $W \leftarrow Width(I)$;
    $H \leftarrow Height(I)$;
    $HJ \leftarrow$ height input size of FE;
    $WJ \leftarrow$ width input size of FE;
    if $H \leq W$ then
        $aspect \leftarrow W/H$;
        $NewH \leftarrow HJ * N$;
        $NewW \leftarrow NewH * aspect$;
    else
        $aspect \leftarrow H/W$;
        $NewW \leftarrow WJ * N$;
        $NewH \leftarrow NewW * aspect$;
    $I \leftarrow I$ resized to $NewH$x$NewW$;
    $P \leftarrow$ patches extracted from $I$ using S;
    $V \leftarrow$ vector of $Z$ positions initialized with zeros;
    for $P_j$ in $P$ do
        $F \leftarrow$ features of $P_j$, from $FE$;
        $V \leftarrow V + F$;
    $V \leftarrow V/|P|$;
    return $V$;
```

After resizing the image, we use a sliding window to extract the image's patches. The window (whose dimensions match the FE dimensions) slides through the entire image according to the stride S. The extracted patches can overlap each other according to the sliding window stride. For example, small values of S imply larger overlapping between patches, reinforcing the image's most prevalent features. For each patch generated in the last step, the algorithm extracts its features with FE, taking advantage of *transfer learning*. The algorithm feeds FE with a given patch and uses the vector F with Z features generated by FE as its representation. Once we extract the features of all image patches, we calculate their average to generate a single vector V of features (the context features) that represents the information of the entire image. In the end, the algorithm outputs the vector V of size Z, which is the output size of FE.

Notice that by keeping the original image's aspect ratio, the algorithm avoids the distortion of essential domain features. Also, our approach is agnostic to the feature extractor (FE) adopted in the process. Thus, different pre-trained models can be used for different contexts. Besides, our approach extracts patches from the image as an intermediate step for obtaining the *local features* related to specific parts of the image. In the end, the algorithm outputs a single vector of context features that may represent the whole image. Since generating context features for representing images is performed as a preprocessing phase, our approach consumes less computational resources in the training phase than approaches that use all individual patches as inputs for training the model. Finally, since we generate context features by *transfer learning* from models that usually were pre-trained on big datasets, our strategy can be a promising approach for training classifiers with relatively small datasets.

3.2 Generating Multiscale Context Features

In the second step, we use the Algorithm 1 to generate context features of a given image I at multiple scales by varying the values of N. Next, context features of different scales are aggregated into a single vector by concatenation. Our approach assumes that the resulting feature vector (*multiscale context features*) represents the global information of a given image at multiple scales. The multiscale context features extracted from a given image can then be used as the input data for training a given arbitrary classifier.

We illustrate our overall approach in figure 1. In this example, our approach generates multiscale context features that integrate context features generated by Algorithm 1 with N=1, N=2, and N=3. Notice that these scales were considered just for the sake of the example since our approach is flexible, allowing the integration of more or fewer scales as needed.

It is crucial to notice that by changing the value of N, the resulting patches focus on areas with different sizes relative to the image size. The larger the N, the smaller the area of the image that a patch contains. Then, given an image I, different values of N can highlight different image patterns or the same pattern at different scales in I. Thus, this strategy can be a promising approach for dealing with sets of images of very different sizes representing the same pattern at different scales.

4 EXPERIMENTS

In this section, we discuss the experiments[1] to evaluate our approach. Firstly, we describe the datasets used in our experiments. Next, we explain the methodology that we follow for performing the experiments. Finally, we discuss our experimental results.

4.1 Datasets

As stated earlier this work is part of a project whose goal is to develop an *image retrieval system* for the petroleum industry. Due to this, we evaluate our approach with a dataset of *geological images*. It is important to notice that the dataset used in this project was developed in cooperation with companies and cannot be shared due to copyright issues. In order to promote the reproducibility of our results and demonstrate it can be applied to other domains we also evaluated the proposed approach with the Stanford Cars dataset (Krause et al., 2013), which is publicly available.

In (Abel et al., 2019), the authors present an ontology for image classification in Petroleum Geology containing 175 classes, including, for instance, ternary diagram, satellite image, geological map, profile, and geological cross-sections. Figure 2 presents some examples of images from these classes. After, a set of geological images was collected from geological reports and labeled by experts for some of the classes specified by the ontology. The dataset contains 25725 images distributed in 45 classes and is unbalanced, with classes with only 36 images and classes with 8450 images, with an average of 571.6 images per class and a standard deviation of 1290.9. Figure 3 represents the distribution of images in each class, emphasizing its unbalanced distribution. Besides, the images in the dataset are significantly heterogeneous in size and visual features. The average image area in the dataset is 1072336 pixels with a standard deviation of 1861876. There are classes with an average area smaller than half of the others, such as, for example, the class *aerial photograph*, which has 495539 pixels of average area, and the class *reference map*, which has 1095760 pixels of average area. Also, for some images, one of the dimensions is much larger than the other. We emphasize this high variation in Figure 4 and in Figure 5, which presents some statistical properties regarding the image sizes and aspect ratios in this dataset. These aspects make this dataset challenging for image classification.

[1] Our code is available at https://github.com/BDI-UFRGS/ContextFeatures

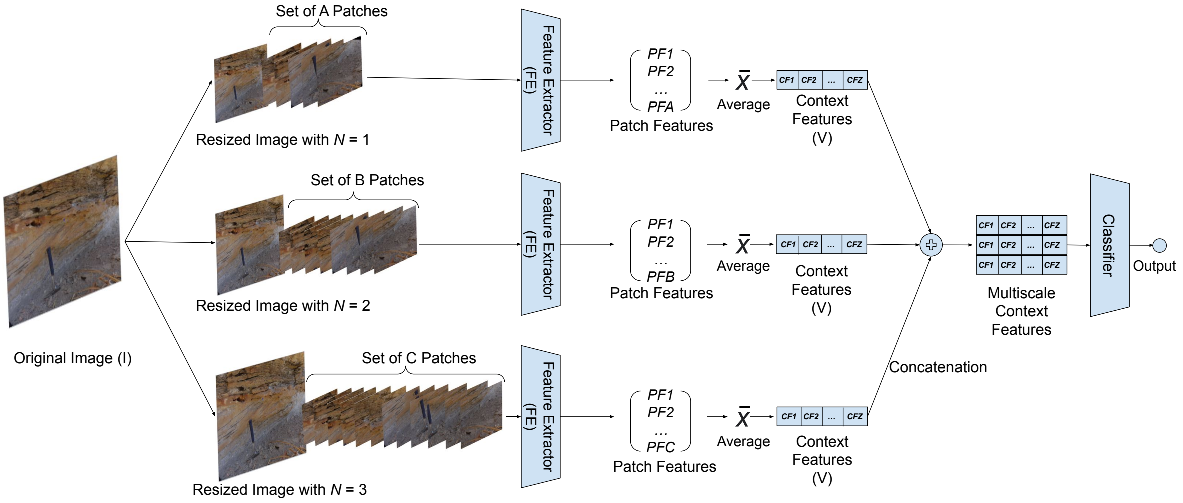

Figure 1: Example of our methodology for image classification with multiscale context features. Given an image, we apply Algorithm 1 with different values of *N* (1,2 and 3, in this example) for generating context features. After, we concatenate the context features of different scales into a single vector of multiscale context features. Finally, we use these features as input for a classifier (in both the training and inference phases).

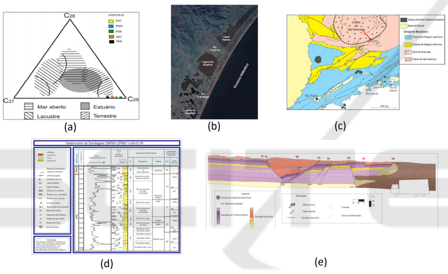

Figure 2: Examples of images illustrating some classes of the ontology: (a)ternary diagram (Missio Júnior, 2014), (b) satellite image (Ivanoff, 2013), (c) geological map (Arruzzo, 2016), (d) profile (Paiva, 2018) and (e) geological cross-section (Silva, 2015).

The Stanford Cars dataset (Krause et al., 2013), on the other hand, has a total of 16185 images distributed in 196 classes. The average number of samples in each class is 84 with a standard deviation of 6.28. The samples range from 61 in the smallest class to 110 in the biggest. Unlike the geological dataset, this one is balanced but also heterogeneous regarding the size and aspect ratio, as shown in Figure 6.

4.2 Methodology

In our experiments, we evaluate our approach for classifying images in both the datasets previously presented. Since our approach is agnostic regarding the pre-trained model used as feature extractor, we applied it with three different pre-trained models[2]:

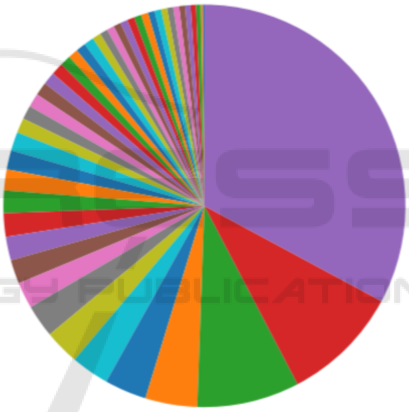

Figure 3: Representation of the number of instances for each class of the dataset.

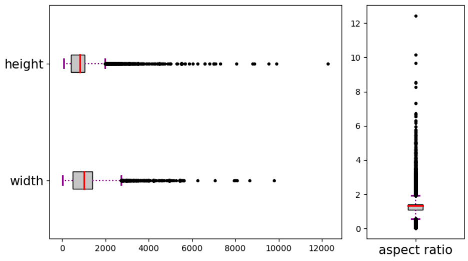

Figure 4: Boxplots representing the statistical properties of the images in the dataset of Geological images, focusing on their height, width, and aspect ratio (as Width/Height).

- ResNeXt-101 pre-trained on ImageNet. ResNeXt (Xie et al., 2017) presents a remarkable performance with fine-tuning (Abou Baker et al., 2022).

[2] We obtained the resNext-101 and DenseNet-121 pre-trained models from https://pytorch.org/vision/stable/models.html, and the CLIP Vit-B/32 from https://github.com/openai/CLIP

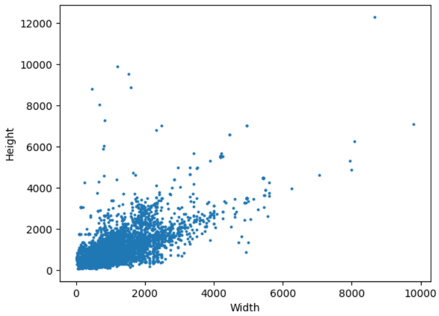

Figure 5: Distribution images according to their dimensions.

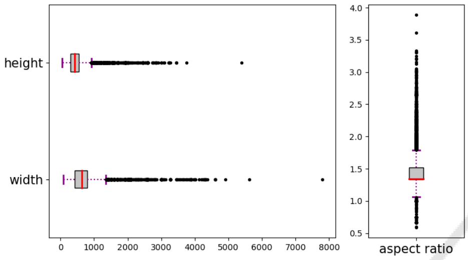

Figure 6: Representation of the statistical properties of the images in the Stanford Cars dataset through boxplots, focusing on their height, width, and aspect ratio (as Width/Height).

- DenseNet-121 pre-trained on ImageNet. DenseNet (Huang et al., 2017) presents a good performance as a feature extractor (Varshni et al., 2019).
- CLIP ViT-B/32 pre-trained in a dataset with 400 million images. CLIP (Radford et al., 2021) is a transformer-based model that presents impressive results in recent studies (Zhai et al., 2022).

We evaluated our approach's capability to represent important features of the target images in different basic scales and in multiscale settings. We applied Algorithm 1 for extracting context features (CF) of every image in the target dataset, using the three selected models as feature extractors in three different basic scales, controlled by the parameter N. Thus, we adopted $N = 1$, $N = 2$, and $N = 3$. By concatenating all possible combinations of the basic three scales, we also transformed the images into their respective representation in terms of *multiscale context features* (MCF). In order to simplify the explanation, we will adopt a notation in which we call N1 the context features generated with $N = 1$, and N1+N2+N3 the multiscale context features constituted by the concatenation of the context features generated with $N = 1$, $N = 2$, and $N = 3$. The notations of other combinations of scales follow this same pattern.

In our experiments, we adopted a *5-fold cross-validation procedure*. In each iteration, we consider one fold as test set, and from the data of the remaining folds, we use 10% as validation data and 90% as training data. We adopt *Adam optimizer* with a *learning rate* of 0.001 and a limit of 100 epochs, applying *early stopping* considering 5 epochs without a minimal improvement of 0.001 in the *cross-entropy loss* of the validation set. Notice also that the feature extractor is not fine-tuned in this process.

We use the following metrics to evaluate our approach: Top-1 Accuracy, Macro Precision, Weighted Precision, Macro Recall, Weighted Recall, Macro F1, and Weighted F1. These metrics provide a good evaluation of the results since they cover several evaluation aspects in a multiclass classification setting. The results reported in Tables 1, 2, and 3 are averages obtained from the test set in the cross-validation procedure. We performed the experiments on a desktop with an Intel i7-10700 CPU and an NVIDIA RTX 3060 GPU. The code was implemented in python, using mainly the PyTorch library[3].

In our first experiment, we tested different values for the stride, which, as described in the Algorithm 1, defines how much the window slides in pixels as a percentage of the feature extractor's input size. We tested four stride values: 100%, 50%, 10%, and 1%. For evaluating our approach, we used the resulting CF to train a simple classifier consisting of only an input layer and a fully-connected layer (Dense layer) with a linear activation function layer as output. Notice that the images in the test set are also transformed into context features or multiscale context features and evaluated by the trained classifier. This experiment was carried out on the dataset of geological images, focusing only on the three basic scales and using just CLIP as a feature extractor. We opted to use CLIP due to the amount of data used in its training, and its reported performance.

Table 1 presents the results of the first experiment. We achieved the best overall performance by adopting 1% of stride S. In general, lower stride values imply higher macro and weighted F1 performance. This result suggests that by using lower values of stride, the resulting context features emphasize the most important features that represent the overall information of the image.

We performed a second experiment in the geological dataset (Table 2) and the Stanford Cars dataset (Table 3), comparing our approach with three other different families of approaches. In the first one, we only use the feature extraction layers of the pre-trained DenseNet, which contains only convolution and pooling operations and can take input images of

[3] https://pytorch.org/

Table 1: Experimental results using different stride values for different scales. CLIP (Radford et al., 2021) was used as a feature extractor and applied to the geological dataset. We group the results according to the scale size (parameter *N*) and for each scale.

Scale	Stride	Top-1 Accuracy		Macro			Weighted	
			Precision	Recall	F1	Precision	Recall	F1
N1	100%	92.44%	85.24%	82.42%	82.57%	92.32%	92.44%	92.33%
N1	50%	92.29%	86.54%	80.72%	82.43%	92.28%	92.39%	92.15%
N1	10%	92.71%	86.96%	81.75%	83.48%	92.62%	92.71%	92.52%
N1	**1%**	**92.93%**	**86.07%**	**83.47%**	**84.6%**	**92.84%**	**92.93%**	**92.85%**
N2	100%	91.92%	84.17%	80.84%	82.20%	91.84%	91.92%	91.81%
N2	50%	91.68%	85.40%	78.04%	80.13%	91.59%	91.69%	91.36%
N2	10%	91.96%	85.76%	78.66%	80.73%	91.88%	91.96%	91.68%
N2	**1%**	**92.57%**	**85.6%**	**82.39%**	**83.73%**	**92.51%**	**92.57%**	**92.48%**
N3	100%	90.23%	80.96%	76.05%	77.92%	90.08%	90.23%	90.03%
N3	50%	90.86%	82.06%	77.39%	79.19%	90.74%	90.86%	90.69%
N3	10%	90.91%	82.15%	77.61%	79.33%	90.78%	90.01%	90.74%
N3	**1%**	**90.94%**	**81.96%**	**78.07%**	**79.6%**	**90.83%**	**90.94%**	**90.79%**

varying sizes. The only constraint of this model is that the smallest image dimension should be greater or equal to 224 pixels. Thus, in this case, the only preprocessing involved was resizing (preserving the aspect ratio) images that were smaller than the input requirements of this model. We used the preprocessed images as input for this model. Then we used the features produced as the output of this model for training the same simple classifier described in the first experiment for evaluating our approach. Notice that in this approach, we do not discard image information. Due to this, we refer to this approach as *lossless* in Table 2 and Table 3. We also perform feature extraction with this model over the images in the test set, and then, the trained model evaluates the features.

For the other two families of approaches, we take the Raw Image (RI), resize the smaller edge of the image to the size of the architecture input, and perform a center crop. The first of these two approaches uses the original image's resulting fragment as input for a pre-trained model used as a feature extractor. We use the resulting features (*simple features* - SF) for representing the image. In this case, we used the same three pre-trained models adopted for generating the context features in our approach. Notice that this approach also uses transfer learning from the same pre-trained models, but uses only the information of an arbitrary (central) part of the original image. We used the same simple classifier of the first experiment to evaluate the approaches that use transfer learning, taking the features extracted by the pre-trained models as input. The images in the test set are processed the same way as the training images and then evaluated by the trained model.

Finally, we also trained an end-to-end DenseNet architecture without pre-training, using as input only the fragments of the image resulting from resizing and cropping (as previously discussed). We include this method in our experiments to compare our approach with the performance of a sophisticated architecture without using transfer learning. In this method, the model input is the *raw image* information, contrasting with the other approaches that use input features provided directly by a pre-trained feature extractor, context features, or multiscale context features.

We refer to these last two families of approaches that discard some image information as *lossy* in Table 2 and Table 3.

4.3 Results

We present our results in the dataset of geological images in Table 2 and the results obtained in the Cars dataset in Table 3. Each experiment is organized in a separate row, corresponding to a specific approach discussed in our methodology section. We grouped our results according to the input information of each compared approach. In order to improve readability, we also specify in the tables if the approach uses transfer learning (TL column) and the metrics mentioned in the previous subsection. In the following paragraphs, we will discuss the results of Table 2, related to the dataset of geological images. Macro F1 is the main metric used in the comparisons as it best represents the effectiveness of the approaches in unbalanced datasets.

Line 1 presents the performance of the DenseNet model trained using only parts of the raw images (lossy information) as input, without taking advantage of transfer learning. This approach presents the worst performance amongst the compared approaches, demonstrating that using transfer learning significantly improved the results in this dataset. This result was expected since the dataset is reasonably small and contains a high variability regarding the visual features of each class.

In Line 2, we show the results of using features extracted by DenseNet, taking the whole image as input (lossless information). The results obtained by this approach are very similar to those of Line 4, which also uses DenseNet. Notice that the approach of Line 4 discards some image information by cropping the images after resizing them. Thus, according to the results, resizing and cropping images produce a reasonable representation of the information contained in the whole image for this model. In Line 4, only macro precision and Macro F1 are slightly superior compared to the results of the approach in Line 2.

In Lines 3-5, CLIP results outperformed those achieved using DenseNet and Resnext. These results suggest that CLIP can produce more representative features from the lossy information of the images obtained after resizing and cropping operations.

Table 2: Comparison of the metrics achieved by each experiment configuration in the geological dataset. Each row contains a different configuration. We divided these setups in the table according to the input information (Column 4) and the use of transfer learning (Column 3). In Column 4 *RI (lossy)* indicates the approach using the **Raw Image** as input with resize and crop to 224x224 input size, and *SF* is the approach that uses the generated **Simple Features** as input. *lossless* indicates the use of the original image information, with resizing only images with the smallest dimension small than 224 pixels, whereas *lossy* indicates the use of the resizing and cropping in all images to the standard input size. Also, *CF* denotes **Context Features** and *MCF* indicates **Multiscale Context Features**. The operator + indicates the concatenation of the features of different scales.

Row	Model	TL	Input Information	Top-1 Accuracy		Macro			Weighted	
					Precision	Recall	F1	Precision	Recall	F1
1	DenseNet	×	RI(lossy)	85.53%	72.44%	70.87%	71.32%	85.54%	85.53%	85.44%
2	DenseNet	√	SF(lossless)	90.23%	81.16%	78.08%	79.39%	90.11%	90.23%	90.1%
3	ResNeXt	√	SF(lossy)	87.77%	76.66%	72.77%	74.37%	87.45%	87.77%	87.53%
4	DenseNet	√	SF(lossy)	89.87%	81.73%	77.91%	79.45%	89.65%	89.87%	89.68%
5	**CLIP**	√	**SF(lossy)**	**92.86%**	**86.33%**	**83.16%**	**84.45%**	**92.74%**	**92.86%**	**92.74%**
6	ResNeXt	√	CF(N1)	87.93%	76.68%	73.17%	74.61%	87.65%	87.93%	87.71%
7	DenseNet	√	CF(N1)	89.85%	80.88%	77.44%	78.86%	89.63%	89.85%	89.67%
8	**CLIP**	√	**CF(N1)**	**92.93%**	**86.07%**	**83.47%**	**84.6%**	**92.84%**	**92.93%**	**92.85%**
9	ResNeXt	√	CF(N2)	88.72%	78.75%	74.06%	75.9%	88.46%	88.72%	88.5%
10	DenseNet	√	CF(N2)	90.06%	81.69%	77.2%	79.03%	89.91%	90.06%	89.9%
11	**CLIP**	√	**CF(N2)**	**92.57%**	**85.6%**	**82.39%**	**83.73%**	**92.51%**	**92.57%**	**92.48%**
12	ResNeXt	√	CF(N3)	88.04%	77.62%	72.74%	74.66%	87.78%	88.04%	87.80%
13	DenseNet	√	CF(N3)	89.2%	79.85%	74.81%	76.81%	88.98%	89.2%	88.99%
14	**CLIP**	√	**CF(N3)**	**90.94%**	**81.96%**	**78.07%**	**79.6%**	**90.83%**	**90.94%**	**90.79%**
15	ResNeXt	√	MCF(N1+N2)	89.68%	80.14%	76.33%	77.9%	89.43%	89.68%	89.49%
16	DenseNet	√	MCF(N1+N2)	91.08%	83.35%	79.65%	81.16%	90.92%	91.08%	90.94%
17	**CLIP**	√	**MCF(N1+N2)**	**93.69%**	**87.76%**	**85.01%**	**86.17%**	**93.64%**	**93.69%**	**93.62%**
18	ResNeXt	√	MCF(N1+N3)	89.84%	80.07%	76.93%	78.28%	89.64%	89.84%	89.7%
19	DenseNet	√	MCF(N1+N3)	91.2%	82.94%	79.75%	81.12%	91.05%	91.2%	91.07%
20	**CLIP**	√	**MCF(N1+N3)**	**93.62%**	**87.36%**	**84.7%**	**85.85%**	**94.55%**	**93.62%**	**93.56%**
21	ResNeXt	√	MCF(N2+N3)	89.31%	79.27%	75.66%	77.18%	89.10%	89.31%	89.14%
22	DenseNet	√	MCF(N2+N3)	90.43%	82,00%	77.86%	79.57%	90.26%	90.43%	90.28%
23	**CLIP**	√	**MCF(N2+N3)**	**92.57%**	**85.11%**	**82.47%**	**83.61%**	**92.5%**	**92.57%**	**92.49%**
24	ResNeXt	√	MCF(N1+N2+N3)	90.10%	80.2%	77.28%	78.52%	89.95%	90.10%	89.98%
25	DenseNet	√	MCF(N1+N2+N3)	91.36%	83.83%	79.98%	81.57%	91.23%	91.36%	91.23%
26	**CLIP**	√	**MCF(N1+N2+N3)**	**93.74%**	**88.07%**	**85.25%**	**86.41%**	**93.73%**	**93.74%**	**93.69%**

In the following discussion, we focus on analyzing the performance of our approach. In this context, we notice that context features generated in different basic scales (in Lines 6-14) present promising results. In general, for all pre-trained models considered in the experiments, the results obtained from the scale with $N = 1$ provide slightly better results in most of the considered metrics when compared with the results in Lines 3-5. An exception to this pattern is the case of DenseNet, whose results are similar but generally worse when using context features.

By comparing the results obtained in the three basic scales, we can notice that when adopting CLIP as the feature extractor, in general, as the scale increases, the performance decreases in most of the metrics. However, this is not true in the case of Resnext and DenseNet, which achieves better results at scale with $N = 2$. We hypothesize that this difference results from the differences in the training datasets used by CLIP, DenseNet, and Resnext.

In Lines 15-23, we present the results using multiscale context features (MCF) generated from the concatenation of two basic scales. We can notice that the approaches using MCF that combine the scales resulting from $N = 1$ and $N = 2$ (Lines 15-17) achieve better results in all metrics when compared to all the previously discussed approaches. Notice that the results achieved by adopting the CLIP model are the best ones in this context. The results in Line 17, obtained using CLIP, surpass the other approaches presented in Lines 15-23. Besides that, by using Resnext and DenseNet as feature extractors, we achieve the best results considering scales of $N = 1$ and $N = 3$. These results are intriguing since, by considering only the context features of basic scales for these models, we obtain the best results with the scale of $N = 2$.The underlying reasons for this result should be investigated in future works. Finally, for this dataset, the approaches using MCF obtained from combining the scales of $N = 2$ and $N = 3$ achieved the worst results. For this dataset, the scale with $N = 1$ produces the most informative features, complemented by context features of the scales with $N = 2$ and $N = 3$ (depending on the pre-trained model used as feature extractor).

Finally, Lines 24-26 present the results obtained from multiscale context features generated from the concatenation of the three basic scales. For all considered pre-trained models, the performance obtained by these approaches is superior to the performances of all the other approaches using context features and multiscale context features. We can see this effect when we compare each approach using a specific model along the different scales and combinations. Also, the ap-

Table 3: Achieved results in the Stanford Cars dataset. This table adopts the same notation adopted in Table 2.

Row	Model	TL	Input Information	Top-1 Accuracy		Macro			Weighted	
					Precision	Recall	F1	Precision	Recall	F1
1	ResNeXt	×	RI(lossy)	23.85%	25.44%	23.92%	23.93%	25.38%	23.85%	23.87%
2	DenseNet	√	SF(lossless)	57.94%	59.30%	57.90%	58.21%	59.08%	57.94%	58.12%
3	ResNeXt	√	SF(lossy)	47.61%	47.56%	47.67%	47.41%	47.37%	47.61%	47.29%
4	DenseNet	√	SF(lossy)	57.33%	57.59%	57.34%	57.25%	57.39%	57.33%	57.15%
5	**CLIP**	√	**SF(lossy)**	**83.29%**	**83.44%**	**83.20%**	**83.24%**	**83.44%**	**83.29%**	**83.29%**
6	ResNeXt	√	CF(N1)	50.79%	50.72%	50.81%	50.57%	50.54%	50.79%	50.47%
7	DenseNet	√	CF(N1)	60.72%	60.90%	60.76%	60.62%	60.69%	60.72%	60.51%
8	**CLIP**	√	**CF(N1)**	**86.52%**	**86.66%**	**86.42%**	**86.48%**	**86.66%**	**86.52%**	**86.54%**
9	ResNeXt	√	CF(N2)	50.43%	50.18%	50.36%	49.77%	50.02%	50.43%	49.73%
10	DenseNet	√	CF(N2)	59.21%	59.10%	59.15%	58.68%	59.02%	59.21%	58.68%
11	**CLIP**	√	**CF(N2)**	**70.03%**	**70.33%**	**69.93%**	**69.80%**	**70.23%**	**70.03%**	**69.80%**
12	ResNeXt	√	CF(N3)	40.68%	39.88%	40.67%	39.28%	39.74%	40.68%	39.22%
13	DenseNet	√	CF(N3)	46.97%	46.57%	46.90%	45.93%	46.82%	46.97%	45.91%
14	**CLIP**	√	**CF(N3)**	**43.74%**	**43.60%**	**43.69%**	**42.37%**	**43.44%**	**43.74%**	**42.30%**
15	ResNeXt	√	MCF(N1+N2)	60.28%	60.72%	60.26%	60.33%	60.53%	60.28%	60.25%
16	DenseNet	√	MCF(N1+N2)	70.23%	70.75%	70.19%	70.34%	70.59%	70.23%	70.29%
17	**CLIP**	√	**MCF(N1+N2)**	**87.27%**	**87.44%**	**87.18%**	**87.26%**	**87.45%**	**87.27%**	**87.31%**
18	ResNeXt	√	MCF(N1+N3)	59.95%	59.27%	58.87%	58.88%	59.11%	58.95%	58.85%
19	DenseNet	√	MCF(N1+N3)	68.74%	69.27%	68.66%	68.82%	69.11%	68.74%	68.79%
20	**CLIP**	√	**MCF(N1+N3)**	**86.43%**	**86.58%**	**86.34%**	**86.40%**	**86.59%**	**86.43%**	**86.45%**
21	ResNeXt	√	MCF(N2+N3)	56.56%	56.76%	56.49%	56.25%	86.62%	56.56%	56.22%
22	DenseNet	√	MCF(N2+N3)	64.95%	65.04%	64.88%	64.72%	64.98%	64.95%	64.73%
23	**CLIP**	√	**MCF(N2+N3)**	**70.92%**	**71.21%**	**70.82%**	**70.80%**	**71.13%**	**70.92%**	**70.81%**
24	ResNeXt	√	MCF(N1+N2+N3)	63.43%	63.95%	63.36%	63.47%	63.81%	63.43%	63.44%
25	DenseNet	√	MCF(N1+N2+N3)	72.51%	73.07%	72.41%	72.60%	72.95%	72.51%	72.60%
26	**CLIP**	√	**MCF(N1+N2+N3)**	**86.71%**	**86.90%**	**86.61%**	**86.69%**	**86.90%**	**86.71%**	**86.74%**

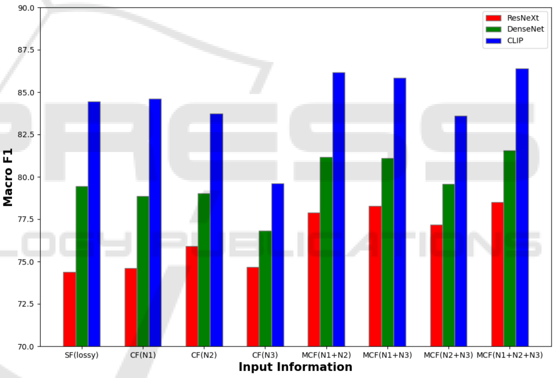

Figure 7: Performance (Macro F1) obtained on the geological dataset using simple features extraction (SF) and the proposed context features approach (CF and MCF).

proach in Line 24, using Resnext, provides worse results when compared to those achieved by DenseNet using lossless image information as input (Line 2). However, this is different for the approaches in Lines 25-26, using DenseNet and CLIP, which achieve better results. Finally, the results in Line 26, using pre-trained CLIP as the feature extractor, achieve the best results in the experiments.

In general, the pre-trained CLIP model allows our approach to achieve the best results when compared with the performance achieved by using DenseNet and Resnext. Furthermore, by using context features and multiscale context features, in general, we achieve better results than conventional approaches that use transfer learning (Lines 2-5). Finally, the multiscale context features generated by considering the three basic scales, in general, achieve better results, suggesting that multiscale context features can represent the information of the whole image at different scales, emphasizing different patterns or the same pattern at different scales. Thus, these features provide different perspectives that can be exploited during training, resulting in robust classifiers.

Next, we will discuss the results regarding the Stanford Cars dataset, reported in Table 3. Generally, the results obtained in this dataset follow the patterns observed in the dataset of geological images. The performance achieved with scale $N = 1$ (Lines 6-8) is better than the performance obtained with simple features (Lines 2-5) and the ResNeXt without transfer learning (Line 1). The other scales do not have the same performance, but, as in the geological dataset, the concatenation of multiple scales improves the results (Lines 15-26). Also, we can notice that CLIP achieves the best overall performance in this dataset. However, a significant difference is that in this dataset, the best performance was achieved using only two scales (Line 17). This difference only appears when adopting CLIP, as both ResNeXt and DenseNet have their best performance using three scales (Lines 24 and 25).

The improvements obtained using our approach in the Stanford cars dataset are more pronounced than those observed in the geological dataset. By using ResNeXt, our approach achieves an improvement of 16.06% of Macro F1, since the performance increases from 47.41%, when using simple features, to 63.47%, achieved by considering multiscale context features with three scales. With DenseNet, our approach achieves an improvement of 15.35% of Macro F1, since that by using simple features we achieve 57.25%, and by using the three-scale MCF we achieve 72.60%. By adopting CLIP as a feature extractor, our approach achieves an improvement of 4.02% of Macro F1, increasing from 83.24%, when adopting simple features, to 87.26%, when adopting MCF with $N = 1$ and $N = 2$.

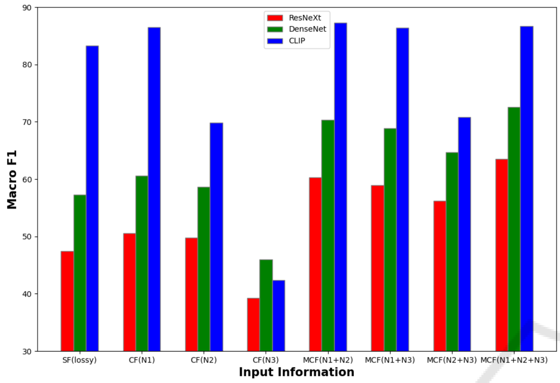

Figure 8: Performance (Macro F1) obtained on the Stanford Cars dataset using simple features extraction (SF) and the proposed context features approach (CF and MCF).

Figures 7 (geological dataset) and 8 (Stanford Cars dataset) represent a visual comparison of the Macro F1 achieved by the classifier, by using simple features and by using context features and multiscale context features. In both figures, it is possible to notice that context features of single scales provide interesting results, but by considering multiple feature scales (MCF) the performance increases in a significant way. The results reinforce that our approach is effective in both datasets.

5 CONCLUSION

This work proposes an approach for geological image classification based on context features. Our goal was to improve the classification of these geological images into an image retrieval system for the petroleum industry. Our methodology splits images into patches of multiple scales and uses transfer learning to generate context features and multiscale context features. Our approach's goal is to represent the information of the whole image, avoiding discarding crucial information. Our procedure represents a preprocessing step for representing the images in the target dataset. Even though fine-tuning is a powerful approach, our methodology based on feature extraction presents a satisfactory performance and promotes a light consumption of computational resources during training time, with the advantage of needing retraining only on the classifier layers. It is also important to notice that our methodology is agnostic regarding image sizes, making it suitable for dealing with datasets that include images of varying sizes.

We evaluated our approach in a geological dataset, which is the target of this project, and also in a public dataset. We also compared our approach with conventional approaches for image classification. Our results show that the proposed approach can generate representative features of the target images. Besides, we concluded that our approach achieves the best results using multiscale context features, surpassing the performance of the alternative approaches considered in our experiments. The results also show that our approach is suitable for dealing with relatively small datasets by taking advantage of transfer learning from pre-trained models (used as feature extractors). Finally, by using the pre-trained CLIP model for generating multiscale context features, our approach achieved the best overall results, outperforming DenseNet and ResNeXt by a reasonable margin, suggesting that this model can generate more representative features in this experimental setting.

We plan to investigate how to improve the proposed approach in future works. For example, we hypothesize that by considering the informativeness of each image patch into account, our approach can generate even better context features. We also intend to explore the relationships among the scales of analysis and their combinations; the pre-trained models used as feature extractors, and the statistical properties of the target dataset. This analysis can provide insights into why features produced by a given pre-trained model produce better results when considering the combination of specific scales of analysis in a given dataset.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) and Petrobras for the financial support of this work.

REFERENCES

Abbas, A., Abdelsamea, M. M., and Gaber, M. M. (2021). Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *Applied Intelligence*, 51(2):854–864.

Abel, M., Gastal, E. S. L., Michelin, C. R. L., Maggi, L. G., Firnkes, B. E., Pachas, F. E. H., and dos Santos Alvarenga, R. (2019). A knowledge organization system for image classification and retrieval in petroleum exploration domain. In *ONTOBRAS*.

Abou Baker, N., Zengeler, N., and Handmann, U. (2022). A transfer learning evaluation of deep neural networks for image classification. *Machine Learning and Knowledge Extraction*, 4(1):22–41.

Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., and Mougiakakou, S. (2016). Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE transactions on medical imaging*, 35(5):1207–1216.

Araújo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., Polónia, A., and Campilho, A. (2017). Classification of breast cancer histology images using convolutional neural networks. *PloS one*, 12(6):e0177544.

Arruzzo, F. (2016). Mapeamento geológico estrutural da formação tiradentes, mesoproterozóico, são joão del rei.

Barstugan, M., Ozkaya, U., and Ozturk, S. (2020). Coronavirus (covid-19) classification using ct images by machine learning methods. *arXiv preprint arXiv:2003.09424*.

Celik, Y., Talo, M., Yildirim, O., Karabatak, M., and Acharya, U. R. (2020). Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images. *Pattern Recognition Letters*, 133:232–239.

Chen, C.-F. R., Fan, Q., and Panda, R. (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dung, C. V. et al. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99:52–58.

Han, F., Yao, J., Zhu, H., and Wang, C. (2020). Underwater image processing and object detection based on deep cnn method. *Journal of Sensors*, 2020.

Hollink, L., Schreiber, G., Wielemaker, J., Wielinga, B., et al. (2003). Semantic annotation of image collections. In *Knowledge capture*, volume 2.

Hong, D., Gao, L., Yokoya, N., Yao, J., Chanussot, J., Du, Q., and Zhang, B. (2020). More diverse means better: Multimodal deep learning meets remote-sensing imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4340–4354.

Horry, M. J., Chakraborty, S., Paul, M., Ulhaq, A., Pradhan, B., Saha, M., and Shukla, N. (2020). Covid-19 detection through transfer learning using multimodal imaging data. *Ieee Access*, 8:149808–149824.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Ivanoff, M. (2013). Sedimentação da lagoa de itapeva, rs - brasil. Master's thesis, Instituto de Geociências, UFRGS.

Kieffer, B., Babaie, M., Kalra, S., and Tizhoosh, H. R. (2017). Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE.

Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., and Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):1–13.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, Y., Zhao, H., Qi, X., Wang, L., Li, Z., Sun, J., and Jia, J. (2021). Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 214–223.

Liang, G. and Zheng, L. (2020). A transfer learning method with deep residual network for pediatric pneumonia diagnosis. *Computer methods and programs in biomedicine*, 187:104964.

Missio Júnior, V. (2014). Geoquímica orgânica das turfeiras das praias de hermenegildo e maravilhas. Master's thesis, Instituto de Química, UFRGS.

Mohan, A. and Venkatesan, M. (2020). Hybridcnn based hyperspectral image classification using multiscale spatiospectral features. *Infrared Physics & Technology*, 108:103326.

Mormont, R., Geurts, P., and Marée, R. (2018). Comparison of deep transfer learning strategies for digital pathology. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2262–2271.

Paiva, R, G. (2018). Estratigrafia de sequências aplicada à formação poti, viseano da bacia do parnaíba. Master's thesis, Instituto de Geociências, UnB.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International*

Conference on Machine Learning, pages 8748–8763. PMLR.

Silva, C. (2015). Análise da deformação tectônica em afloramento da formação pindamonhangaba (bacia de taubaté, rift continental do sudeste do brasil).

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, 2016.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Tayal, A., Gupta, J., Solanki, A., Bisht, K., Nayyar, A., and Masud, M. (2021). Dl-cnn-based approach with image processing techniques for diagnosis of retinal diseases. *Multimedia Systems*, pages 1–22.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R., and Mittal, A. (2019). Pneumonia detection using cnn based feature extraction. In *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*, pages 1–7. IEEE.

Wang, J., Song, L., Li, Z., Sun, H., Sun, J., and Zheng, N. (2021). End-to-end object detection with fully convolutional network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15849–15858.

Wong, R. C. and Leung, C. H. (2008). Automatic semantic annotation of real-world web images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1933–1944.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., Yu, L., Ni, Q., Chen, Y., Su, J., et al. (2020). A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*, 6(10):1122–1129.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113.

Zhang, D., Islam, M. M., and Lu, G. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362.

Zhu, W., Braun, B., Chiang, L. H., and Romagnoli, J. A. (2021). Investigation of transfer learning for image classification and impact on training sample size. *Chemometrics and Intelligent Laboratory Systems*, 211:104269.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

Zhuang, P., Li, H., Tan, S., Li, B., and Huang, J. (2021). Image tampering localization using a dense fully convolutional network. *IEEE Transactions on Information Forensics and Security*, 16:2986–2999.