Exploiting Deep Metric Learning for Mable Quality Assessment with Small and Imbalanced Image Data

George K. Sidiropoulos
HUman-MAchines Interaction Laboratory (HUMAIN-Lab), Dept. of Computer Science International Hellenic University
Kavala, Greece
georsidi@cs.ihu.gr

Athanasios G. Ouzounis
HUman-MAchines Interaction Laboratory (HUMAIN-Lab), Dept. of Computer Science International Hellenic University
Kavala, Greece
athouzo@cs.ihu.gr

George A. Papakostas*
HUman-MAchines Interaction Laboratory (HUMAIN-Lab), Dept. of Computer Science International Hellenic University
Kavala, Greece
gpapak@cs.ihu.gr

Ilias T. Sarafis
Dept. of Chemistry International Hellenic University
Kavala, Greece
isarafis@chem.ihu.gr

Andreas Stamkos, Vassilis Kalpakis
Intermek A.B.E.E.
Kavala, Greece
{a.stamkos, v.kalpakis}@intermek.gr

George Solakis
Solakis Antonios Marble S.A.
Drama, Greece
george@solakismarble.gr

Abstract—**The classification of ornamental dolomitic marble stone tiles has been an issue in the past years, even more so according to their aesthetical criteria. Quality control and product classification during the final stage of a production line is the main problem of this step, which, when done right, can increase profitability. Machine Learning has been employed in many cases to improve and accelerate the decision and assessment process of this step. Due to the unique nature of the problem, the image datasets constructed can be heavily unbalanced, as there is no control over the number of marble tiles that are collected for each class. This paper examines the application of metric learning and more specifically Siamese networks, for the classification of dolomitic marble tiles, examining the performance of 7 convolutional neural networks as feature extractors. The results are then compared to the application of transfer learning techniques on the same convolutional networks. The experiments conducted revealed the high robustness of the metric learning approach, by providing very low standard deviation (stdev 0.53%) between the models' performance, compared to transfer learning where results per model vary (stdev 2.53%) to a higher degree.**

Keywords—machine vision, metric learning, deep learning, dolomite tile sorting

I. INTRODUCTION

Ornamental rocks have been extensively used as a decoration, as well as a building material for centuries, with Greece being a main source of production all over the world. An important aspect of ornamental dolomitic marble stone tiles (marble tiles), is their aesthetic factor, which plays a huge role in the profitability of the production industry. Along with the endurance of this material, materials that come from the earth's crust are being used as decoration material until today, despite the modern alternatives like glass, metal, or concrete. Marble tiles are quarried from the earth as blocks, which are then cut into slabs and manufactured. The last and important step in the production line is the classification of the tiles, which is still being done by expert geologists. The classification relies on two factors, which are heavily considered during the process: 1) the grouping of similar ornaments, an important factor when tiling a wall or floor and 2) the grouping of tiles based on the number of impurities and visible cracks, which can change the overall look of a natural rock. Naturally, the fewer cracks and impurities a tile has, the higher the value the tile can reach in the market. Therefore, this production step can become time consuming and sometimes is very subjective, with the misclassification of the final product becoming very costly. As a result, the application of machine learning (ML) and computer/machine vision (CV/MV) has been considered by many in this field, automating the quality control and assessment process, as well as the classification of the tiles, reducing the production cost by a high degree.

Automatic marble slab classification has been an issue, with Hernandez et al. [1] making use of the Multi-Layer Perceptron, combined with backpropagation. The issue has seen many different approaches, such as Learning Vector Quantization to cluster and classify marble slabs, taking into account their texture information. A similar approach to [1] was followed later in 2005, achieving a classification rate of 98.9% for three classes of "Crema Marfil Sierra de la Puerta" types of marble slabs [2]. Convolutional Neural Networks (CNNs) were applied in 2017 were applied to classify granite tiles, while also applying augmentation and majority voting techniques [3]. Two years later, CNNs were used again for the classification of travertine image tiles, classifying them into two classes [4], while in 2020, the VISUAL Geometry Group 16 (VGG16) network was used for the identification of peridotite, basalt, marble, gneiss, conglomerate, limestone, granite and magnetite quartzite, reaching recognition rates of 96% [5]. VGG16, Residual Network (ResNet) and LeNet were also used to classify marble tiles into 28 classes, comparing their performance, with VGG16 achieving a 97% accuracy rate [6]. In 2021, machine learning techniques [7] were applied to images constructed from the extraction of texture descriptors, on the same dataset used in the current study. The performance of different ML models was compared for each of the 18 different texture descriptors, on a dataset constructed and provided by Solakis Marble S.A. This research was then extended [8] by examining the performance of transfer learning (TL) techniques of 15 pre-trained Convolutional Neural Networks (CNNs), highlighting the interpretability those models can offer through activation mapping. To extend the research conducted in the previous studies [7], [8], as well as to tackle the existing issue of the imbalanced datasets in the literature, we examine the performance of 7 of the CNNs applied in the previous study trained using the metric learning approach. More specifically, we take advantage of the inherent capabilities of Siamese networks, namely the good generalization in cases of small and imbalanced datasets and examine their performance on the same dataset in the previous study, as well as using a smaller balanced dataset.

The main contributions of this study are the following:

1. Comparison of 7 pre-trained CNNs on real world data originating from the production line of natural stone tile production.
2. Improvement upon the classical TL approach.

This paper is organized as follows: Section II described the dataset and methodology employed in the study, Section III presents the experiments conducted, along with the results obtained and, finally, Section IV presents the discussion of the results.

II. Materials and Methods

A. Dataset

The dataset is compiled from 30x60cm marble tiles as shown in Figure 1, delivered by Solakis Antonios Marble S.A. The slabs are exclusively quarried in the village of Kokkinoghia of Drama, north-east of Greece, which are then cut and the decorative material is extracted. The Kokkinoghia Grey stones, taking their name from the village where they are extracted and most commonly referred to as Grey Lais, are carbonate metamorphic rocks. Carbonate metamorphic rocks, commonly known as dolostone or dolomites, are chemically consisted of 94% mineral dolomite and 6% of mineral calcite. The tiles' textures can vary a lot, from uniformly distributed straight lines to randomly curved ones, with their color ranging from light to dark grey colors.

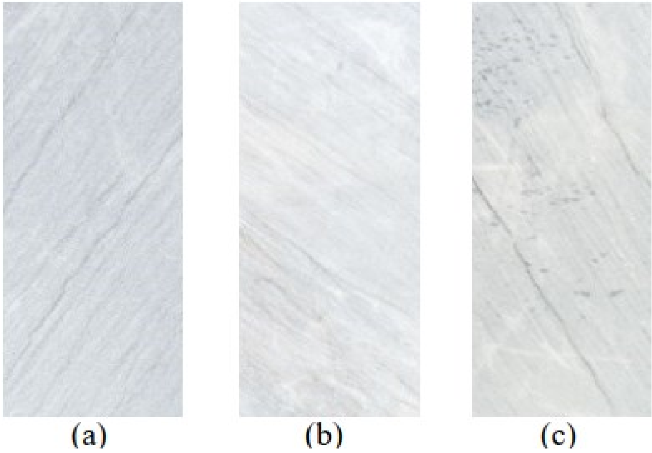

Figure 1 Representative tiles of the three classes: a) Lais G Extrav (A), b) Lais GA (B) and c) Lais GM (C).

The image acquisition process was performed using a low-cost experimental setup inside an industrial environment, collecting a total of 986 images of the polished side of the tiles. Each image had a resolution of 1500x725 pixels, compressed using the jpg image format. After the collection process, specialized workers classified the samples into three quality classes, based on their decorations: Class A, B and C, with each class being consisted of 697, 133 and 156 samples respectively. Due to the obvious sample imbalance, the first class was downsampled by selecting 200 random samples, with the dataset's size resulting in a total of 489 images.

B. Methodology

The study's pipeline can be seen in Figure 1, with each step being described in this section.

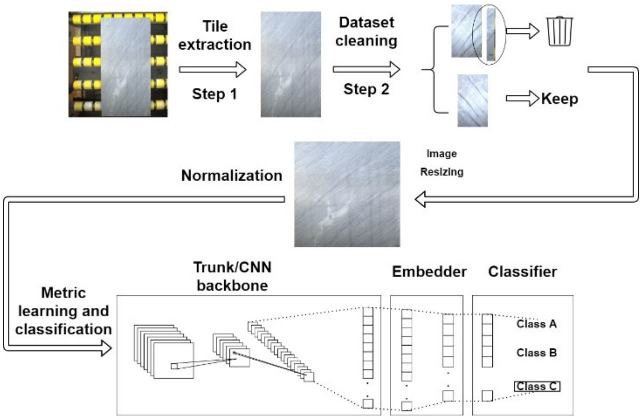

Fig. 2 Pipeline of the proposed methodology.

1) Dataset preparation

Before training the models with the image dataset, some preprocessing steps were followed in order to detect, extract and process the RGB tile image. In the first step, it is crucial to remove the entire surrounding environment (noise) from the image and extract the marble tile. The extraction process included the following steps: 1) Convert the image from RGB to HSV color space, 2) apply a Gaussian blur, 3) threshold the image using specific values, 4) detect contours, 5) detect horizontal and vertical lines, 6) determine the four corners of the tile's rectangle and 7) apply a perspective transformation to align the tile to a 400x700 pixel image.

After examining the resulting dataset, there were cases of marble tile images where the surrounding environment was still visible (noise), due to lack of precision during the extraction. Due to the nature of the metric learning process, which will be discussed in the following subsection, the dataset was reduced, even more, removing all the images where there was visible noise. Thus, the resulting dataset had the following samples per class: Class A: 66, Class B: 67 and Class C: 96 samples. It is obvious that the dataset has been reduced by a large degree.

After the extraction process, all the images were resized to 224x224, the original size of the images that were used to train the CNNs. Lastly, during the normalization step, the image's values were normalized between the values of [0, 1]. The last step was performed using the PyTorch Library [9], using the recommended mean and std values for the normalization.

2) Metric Learning

Similarity learning is a sub-field of machine learning, which aims to learn the distance between two data points. Similarity measures have been present for many years and aim to describe how similar are two data points, with similar data points having a high similarity value. A very popular example of such similarity-dissimilarity based learning is the nearest neighbors classification process, which uses the standard Euclidean distance. Using a standard distance like this, can omit important properties that exist in the dataset [10], which, as a result, created the need for the application of a similarity that brings similar data as close as possible. Due to its nature, it has become a very viable alternative to the standard training approach in small datasets [11].

Metric learning is closely related to distance/similarity learning, with its purpose being to learn a distance (or similarity) function between two objects. In other words, it extracts a high-dimensional embedding vector, aiming to maximize the semantic similarity of the embeddings of samples from the same class. Another advantage of this type of learning is that it can generalize to classes that the model has not been trained before, as it was trained to simply learn the general concept of similarity, in comparison to the traditional class-based learning which learns class-specific features [12]. When deep neural networks (DNNs) are employed to extract such embedding vectors, the training approach is called Deep Distance Metric Learning (DDML), which is also the case in our study.

During the last step of the pipeline, consisted of the DDML, we incorporated 7 pre-trained CNNs using the ImageNet database, provided by the PyTorch library, which were used as the trunk of the DDML network. Specifically, the following CNNs were used: DenseNet121, DenseNet201 [13], ResNet50, ResNet152 [14], VGG16, VGG19 [15] and MobileNetV2 [16].

For the evaluation of the models, we performed a stratified 10-fold cross validation technique, which splits the dataset in 10 equal parts, training the model on the 9 and testing on the remaining 1 part. It is worth noting that a stratified splitting maintains the percentage of samples for each class in each fold, helping against overfitting to a certain degree in cases of unbalanced datasets. Moreover, we used the accuracy, precision, recall and f1-score metrics to evaluate the models' performance. Lastly, the experiments were conducted using the Python programming language, using the PyTorch Library for the models, along with the Pytorch Metric Learning Library [17] for the metric learning process. As metric learning requires the feeding of combinations of images to the model during the training process, we employed the MPerClassSampler from the latter library, generating batches of 16 samples per class (a total of 48 images per batch).

The training approach required the generation of triplets of images, the anchor, the positive and the negative image, with the positive image being the same class as the anchor and the negative belonging to a class other than the anchor image. By using three images, during the training process, the model aims to minimize the distance between the anchor and the positive image, while simultaneously maximizing the distance between the anchor and the negative image. The loss used in our study is the Triplet Margin Loss (or just Triplet Loss) described in Eq.(1):

$$L_{triplet} = \max(d_{ap} - d_{an} + m, 0) \tag{1}$$

With d_{ap} being the distance between the anchor and the positive sample, d_{an} the distance between the anchor and the negative sample and m a constant value denoting the margin between the pairs. The max function is used to maintain its values ≥ 0.

To further describe the model construction process we added additional layers to the aforementioned models, which were divided into two different models, the "embedder" and the "classifier". The embedder model, converted the output of the last convolutional layer of the CNN to an embedding vector of 256 values, while the classifier output the predicted class of the input image. It should be noted that the weights of all the layers of the original CNN models were being updated during the training process. It is obvious that the model now resembles that of a classifier, instead of a feature extractor/similarity learning model.

The models were trained for 100 epochs, with 208 combinations of images being fed to the model during each epoch. For the loss function, the triplet margin loss was used for the trunk and embedder parts of the model, with the cosine similarity as the distance between data points. Moreover, threshold reduction, to reduce the many loss values to one, with a zero-mean regularizer applied to the weights and embeddings, while the cross-entropy loss was used for the classification loss and the Adam optimizer as the optimization algorithm of all the models (trunk, embedder and classifier).

The reasoning behind the final model structure is so that the model can later be used to classify new incoming marble tile images, instead of following an image retrieval kind of process, where we would have to compare the new incoming samples with some ground truth images from each class from a database. As mentioned before, DDML does not learn class specific features and simply compares the two images and having in mind that marble tile classification can be very subjective when it comes to market preference, the classification process would become more tedious. This would require a large number of images for each class to be stored, having to compare the new sample with all the images, requiring the application of a decision-making algorithm given the similarity scores received for each image.

Furthermore, during our experiments, we compared the performance of DDML with the TL approach of the same models, following the same process described in our previous study [8].

III. Experiments

A. Results

Table 1 and Table 2 show the results we obtained from the DDML and TL process respectively, where the best model (DensetNet201) performed the best following the TL approach at 80.30% F1 score. In general, the results obtained from both approaches are satisfactory, where almost all models perform above 75% F1 score and above 77% Accuracy. It is worth noting that DDML performs, on average, better than TL, with a low deviation. The best results obtained by following the DDML approach were from the DenseNet201 model too, with 79.78% Accuracy, 80.78% Precision, 79.30% Recall and 79.05% F1 score. The model that performed the worst is different in both cases, with VGG16 in DDML performing the worst in general.

Table 1 Results obtained from the DDML approach.

	Accuracy (%)	**Precision (%)**	**Recall (%)**	**F1 score (%)**
DenseNet121	78.29	79.62	78.01	77.71
DenseNet201	**79.78**	**80.78**	**79.30**	**79.05**
MobileNetV2	78.54	78.86	78.09	77.82
ResNet152	78.90	79.91	78.83	78.48
ResNet50	79.19	80.90	78.66	78.47
VGG16	79.54	79.19	78.82	77.50
VGG19	79.19	79.47	78.63	77.46
Average	79.06	79.82	78.62	78.07
Stdev	0.53	0.77	0.45	0.60

TABLE 2 RESULTS OBTAINED FROM THE TRANSFER LEARNING APPROACH.

	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
DenseNet121	75.08	77.09	73.99	74.06
DenseNet201	**80.77**	**83.34**	**79.78**	**80.30**
MobileNetV2	73.77	74.31	73.40	72.64
ResNet152	79.86	83.05	79.38	79.33
ResNet50	78.97	81.16	78.70	78.78
VGG16	77.25	78.58	76.73	76.09
VGG19	77.27	78.12	76.55	76.16
Average	77.57	79.38	76.938	76.768
Stdev	2.53	3.31	2.54	2.84

IV. CONCLUSIONS

Dolomitic marble tile classification using Deep Distance Metric Learning was analyzed extensively, using 7 well-known deep CNNs, as well as comparing the aforementioned approach with transfer learning using the same models.

The experimental study revealed no significant improvement, despite the well-established improvement of DDML in cases of small datasets. Despite that, a more stable performance across all models was performed, with very low standard deviation between the models, compared to transfer learning where results per model vary to a higher degree. This outcome is very important when dealing with a real-time application where an accurate, as well as a lightweight deep learning model needs to be used. In this context, the proposed DDML approach allows the use of lower complexity models without sacrificing accuracy.

ACKNOWLEDGMENT

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: T1EDK-00706).

REFERENCES

[1] V. G. Hernandez, P. C. Perez, L. G. G. Perez, L. M. T. Balibrea, and H. Puyosa Pina, “Traditional and neural networks algorithms: applications to the inspection of marble slab,” in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, Vancouver, BC, Canada, 1995, vol. 5, pp. 3960–3965. doi: 10.1109/ICSMC.1995.538408.

[2] J. Martinez-Alajarin, J. D. Luis-Delgado, and L. M. Tomas-Balibrea, “Automatic System for Quality-Based Classification of Marble Textures,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 35, no. 4, pp. 488–497, Nov. 2005, doi: 10.1109/TSMCC.2004.843236.

[3] A. Ferreira and G. Giraldi, “Convolutional Neural Network approaches to granite tiles classification,” *Expert Syst. Appl.*, vol. 84, pp. 1–11, Oct. 2017, doi: 10.1016/j.eswa.2017.04.053.

[4] I. Pence and M. Ş. Çeşmeli, “Deep Learning in Marble Slabs Classification,” 2019.

[5] X. Liu, H. Wang, H. Jing, A. Shao, and L. Wang, “Research on Intelligent Identification of Rock Types Based on Faster R-CNN Method,” *IEEE Access*, vol. 8, pp. 21804–21812, 2020, doi: 10.1109/ACCESS.2020.2968515.

[6] M. Canayaz and F. Uludağ, “MARBLE CLASSIFICATION USING DEEP NEURAL NETWORKS,” *Eur. J. Tech.*, pp. 52–63, Jun. 2020, doi: 10.36222/ejt.671527.

[7] G. K. Sidiropoulos, A. G. Ouzounis, G. A. Papakostas, I. T. Sarafis, A. Stamkos, and G. Solakis, “Texture Analysis for Machine Learning Based Marble Tiles Sorting,” in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, NV, USA, Jan. 2021, pp. 0045–0051. doi: 10.1109/CCWC51732.2021.9376086.

[8] A. Ouzounis, G. Sidiropoulos, G. Papakostas, I. Sarafis, A. Stamkos, and G. Solakis, “Interpretable Deep Learning for Marble Tiles Sorting:,” in *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications*, Online Streaming, --- Select a Country ---, 2021, pp. 101–108. doi: 10.5220/0010517001010108.

[9] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *ArXiv191201703 Cs Stat*, Dec. 2019, Accessed: Aug. 11, 2021. [Online]. Available: http://arxiv.org/abs/1912.01703

[10] J. L. Suárez-Díaz, S. García, and F. Herrera, “A Tutorial on Distance Metric Learning: Mathematical Foundations, Algorithms, Experimental Analysis, Prospects and Challenges (with Appendices on Mathematical Background and Detailed Algorithms Explanation),” *ArXiv181205944 Cs Stat*, Aug. 2020, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/1812.05944

[11] Y. Du, C. Liu, and B. Zhang, “Detection of GH Pituitary Tumors Based on MNF,” in *2019 Chinese Control And Decision Conference (CCDC)*, Nanchang, China, Jun. 2019, pp. 635–639. doi: 10.1109/CCDC.2019.8832789.

[12] M. Karpusha, S. Yun, and I. Fehervari, “Calibrated neighborhood aware confidence measure for deep metric learning,” *ArXiv200604935 Cs Stat*, Jun. 2020, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/2006.04935

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *ArXiv160806993 Cs*, Jan. 2018, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/1608.06993

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385

[15] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Apr. 2015, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/1409.1556

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *ArXiv180104381 Cs*, Mar. 2019, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/1801.04381

[17] K. Musgrave, S. Belongie, and S.-N. Lim, “PyTorch Metric Learning,” *ArXiv200809164 Cs*, Aug. 2020, Accessed: Aug. 11, 2021. [Online]. Available: http://arxiv.org/abs/2008.09164