Cultural heritage image classification using transfer learning for feature extraction: a comparison

Radmila Janković Babić

Mathematical Institute of the Serbian Academy of Sciences and Arts, Kneza Mihaila 36, Belgrade, Serbia

Abstract

Image classification in the domain of cultural heritage becomes extremely important with the development of digitisation practices. This study aims to analyze how classification performance on the small dataset representing cultural heritage changes depending on the feature extraction method. The dataset comprised of 150 images belonging to three classes: (i) archaeological sites, (ii) frescoes, and (iii) monasteries. Five transfer learning architectures were used to extract the features from images, while classification was per-formed using four traditional machine learning algorithms, mainly Random forest, Naïve Bayes, Decision tree, and Multilayer perceptron classifier. The results suggest that Random forest and Multilayer perceptron are the most suitable algorithms for classification of cultural heritage images, especially when used in combination with the DenseNet121 pre-trained architecture. Naïve Bayes also performed well, with a maximum accuracy of 100% obtained when features are extracted using EfficientNetB0. However, the Decision tree algorithm reached only moderate performance.

Keywords

Cultural heritage, classification, transfer learning

1. Introduction

Preservation of cultural heritage remains one of the most important tasks in the era of digitisation. In general, cultural heritage can be classified into tangible or physical (such as buildings, monuments, archaeological remains, art works, artifacts) and intangible (such as traditions, language, rituals, skills, folklore). As cultural heritage represents values, traditions, and beliefs of national identity, it shapes future generations and creates a strong bond to their history and surroundings. Since cultural heritage can easily be damaged and destroyed, it is essential to find adequate ways to restore and preserve it.

Digital technologies play a vital role for the preservation and restauration of cultural heritage. Recently, the use of Machine Learning (ML) techniques has proven to be an appropriate way to deal with preservation of cultural heritage. However, such use does not come without barriers. Major problems in this domain are the quality and size of datasets [1]. To tackle the dataset size problem, transfer learning architectures can be utilized where deep convolutional neural networks (CNNs) are trained on very large image datasets, and then applied on smaller data, while ML-based approaches are frequently used to enhance heritage objects using image reconstruction approaches [2, 3, 4].

Recent contributions of classification techniques in the domain of cultural heritage include the use of the multilayer perceptron (MLP), averaged one dependence estimators, forest by penalizing attributes, and k-nearest neighbor rough sets and analogy-based reasoning for classification of altar, gargoyle, dome, column, and vault images [5], and the performance was compared to the CNN. Feature extraction using VGG16 and classification using Random forest (RF) were performed in [6] with the aim to classify Batik types. Multiple linear regression and fuzzy inference models were used to predict

VIPERC2022: 1st International Virtual Conference on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding, 12 September 2022

EMAIL: rjankovic@mi.sanu.ac.rs
ORCID: 0000-0003-3424-134X

the life of built cultural heritage in [7], while logistic regression approach was compared to maximal entropy for predictive modeling of archaeological site locations in [8]. Chronological classification of ancient painting was performed in [9] using the Support Vector Machine (SVM) classifier.

Although traditional ML approaches proved to be accurate in classification of cultural heritage, more recent approaches are focused on deep learning, and specifically transfer learning. One of the major advantages of these approaches lies in the fact that extraction of features from the images is performed automatically, however, the selection of network configuration is more complex. In addition, deep learning approaches usually require large datasets to learn from, so transfer learning approaches were developed to reduce the computational efforts. Some of the recent contributions of deep learning and transfer learning approaches for cultural heritage include classification of Indian heritage sites using MobileNetV2 architecture [10], multimodal classification of cultural datasets [11], classification of four cultural heritage sites, mainly Baalshamin, Temple of Bel, Tetrapylon, and Roman theatre at Palmyra, using Support Vector Machine (SVM) algorithm, transfer learning based on AlexNet architecture, cloud vision approach, and full CNN [12]. Classification of architectural heritage has been performed in [13] using deep learning networks AlexNet, InceptionV3, ResNet, and Inception-ResNet-v2, while in [14] a CNN model from scratch was trained on the same dataset with good performance. The comparison of pre-trained CNN networks for cultural heritage image classification has been done in [15, 16], while in [17] the authors compared the performance of ResNet-18 and custom CNN with SVM and RF used as classifiers of Iberian Ceramics. However, to the author's knowledge, no studies aimed to compare the performance of several ML algorithms when different pre-trained architectures are used for feature extraction.

Hence, the aim of this paper is to compare the performance of four traditional ML algorithms when feature extraction is performed using five pre-trained architectures: (i) MobileNet, (ii) InceptionV3, (iii) Xception, (iv) EfficientNetB0, and (v) DenseNet121. When discussing performance, the focus of this study will be on small sets of data, which are by nature harder to classify correctly, as complex models usually require more data to accurately learn from it.

This paper is structured as follows. Section 2 presents the data and describes the methodology, while Section 3 discusses the obtained results. Section 4 presents the conclusions.

2. Data and methodology

2.1. Data

The dataset used in this study was first introduced in [18] where the aim was to compare the performances of decision tree classifiers. Here, however, the aim is to observe to what extent the classification performance changes depending on the pre-trained architectures that are used to extract features from images.

The dataset consists of 150 color images obtained from Google Images and Flickr, belonging to three classes: (i) archaeological sites, (ii) frescoes, and (iii) monasteries. All images are of size 150x150 pixels. Samples of images from each class are shown in Fig. 1. The dataset was divided into training and test sets, where 35 images per class were used for training and 15 images per class were used for testing the models.

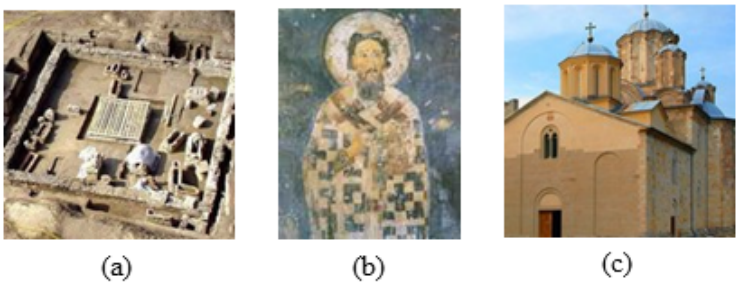


Figure 1: Example of images from the dataset – (a) archaeological site, (b) fresco, (c) monastery

2.2. Methodology

The methodology used in this study consists of several steps. First, pre-trained architectures are used to extract features from the images. The features are then normalized, after which the classification is performed using traditional ML algorithms. Finally, the performance of each model is evaluated.

2.2.1. Pre-trained architectures

In this study, five transfer learning architectures were used for feature extraction, namely MobileNet, DenseNet121, Xception, InceptionV3, and EfficientNetB0. The loaded weights were pre-trained on ImageNet.

The main strategy behind MobileNet is that it is built using depthwise separable convolutions. It consists of 28 layers, where all layers are followed by batch normalization and use the ReLU function, except for the last fully connected layer which uses a softmax function [19].

DenseNet121 is a pre-trained architecture consisting of 120 convolutional layers, four average pooling layers, and one fully connected layer. DenseNet is very similar to ResNet, with the main difference being the concatenation of the output feature maps with the inputs [20].

InceptionV3 is a deep learning architecture that applies convolutional transformations and max-pooling to each layer, and then concatenates these results into an output. InceptionV3 consists of 42 layers.

Xception is a deep convolutional neural network architecture that is based on depthwise separable convolution layers [21]. It is an extension of the Inception architecture, but instead of the Inception modules, it uses depthwise separable convolutions. The Xception architecture is 71 layers deep and consist of 36 convolutional layers.

EfficientNetB0 is a convolutional deep neural network that was developed in an attempt to answer the question if there is a way to scale up ConvNets in order to obtain better accuracy and efficiency [22]. This was possible by increasing the network depth, channel width, and image resolution, as the authors proposed [22]. EfficientNetB0 is 237 layers deep.

2.2.2. ML algorithms

Classification was performed using four ML algorithms, in particular RF, Multilayer Perceptron classifier (MLPC), Naïve Bayes, and Decision tree.

RF is an ensemble machine learning algorithm that is based on bootstrap aggregation. This algorithm constructs a number of decision trees that work as an ensemble where each tree predicts a class, and the prediction outcome will be the class with most votes [23]. The working process of RF classification starts by randomly selecting samples from the training set, constructing decision trees for each sample, generating an output for each decision tree, and finally selecting the most voted outcome.

MLP is a feedforward neural network-based algorithm that consists of an input layer, one or more hidden layers, and an output layer. The algorithm works by assigning weights to the inputs in a neuron, summing those weights and passing them through an activation function, and then propagate the results to the next layer, until it reaches the output layer. The error between the expected and real output is then calculated and backpropagated through the network, with the aim to minimize the cost function.

Naïve Bayes classifier is based on Bayes' theorem, and it assumes strong independence, meaning that each feature is not affected by other features. The algorithm works by first calculating prior probabilities for each class, then calculating the likelihood probability, and finally calculating the posterior probabilities for each class. The prediction outcome will be the class with the highest conditional probability.

Decision tree algorithm is a supervised learning approach that uses a tree-structured classifier to perform classification. Decision trees consist of nodes and branches, mainly the root node which represents the dataset, branches which represent the decision rules, and leaf nodes that represent the outcome.

Model configuration is the same for each algorithm. RF model consisted of 70 trees in the forest, and the quality of the split was measured using Gini impurity. Naïve Bayes and Decision tree used default configuration as described in the scikit documentation [24], while MLPC consisted of three hidden layers with 150, 100, and 50 neurons, respectively, with rectified linear unit (ReLU) activation function, and stochastic gradient-based optimizer, i.e. Adam. The maximum number of iterations was set to 300, the L2 regularization term was set to 0.0001, while the learning rate was set to 0.001.

2.2.3. Performance evaluation

Performance evaluation for each model was done using widely known metrics – precision, recall, F1-score, and accuracy.

Precision is the ratio of true positive cases (TP) to the total predicted positive cases (which is the sum of TP and false positive cases (FP)), and can be calculated as:

$$P = \frac{TP}{TP + FP}. \tag{1}$$

Recall is the ratio of correctly predicted positive cases to all cases in the positive class. Recall is calculated by dividing the number of true positives (TP) to the sum of TP and false negative (FN) cases, as in:

$$R = \frac{TP}{TP + FN}. \tag{2}$$

F1-score represents the harmonic mean of precision and recall, and can be calculated as:

$$F1 - score = 2 * \frac{P * R}{P + R}. \tag{3}$$

Finally, accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4}$$

3. Results and discussion

While feature extraction was performed using the pre-trained architectures, classification was done using the traditional ML models. The smallest differences in performance between different pre-trained architectures were observed for RF where for features extracted using MobileNet, DenseNet121, Xception, and EfficientNetB0 the obtained accuracies are 97.78%, while for InceptionV3 the accuracy was lower – 88.89%. For features extracted using DenseNet121 and InceptionV3, the MLPC obtained a 100% accuracy, while for the EfficientNetB0, MobileNet, and Xception, the accuracies were 97.78%, 91.11% and 86.67%, respectively. On the contrary, Naïve Bayes and Decision tree obtained the lowest performance in terms of accuracy. When feature extraction was performed using EfficientNetB0, the Naïve Bayes obtained a 100% accuracy, however for other pre-trained architectures the accuracies range from 53.33% (InceptionV3) to 95.56% (MobileNet). Considering classification using the Decision tree, the highest accuracy was obtained when features are extracted using DenseNet121 and Xception (82.22%), while for other pre-trained architectures, the accuracies range from 73.33% (MobileNet and InceptionV3) to 75.56% (EfficientNetB0). These results are presented in Table 1.

It is interesting to observe which classes the models confuse the most. When DenseNet121 was used for feature extraction, it can be seen that RF misclassified only one image of archaeological site as fresco, while Naïve Bayes classified 10 images of frescoes as archaeological sites, and six images of frescos as monastery. The decision tree misclassified five images of archaeological sites as fresco, and three images of archaeological site as monastery (Figure 2). Hence, MLPC was found to be the most successful in classifying images of cultural heritage when using the DenseNet121 architecture for feature extraction.

Table 1
Model accuracies

Pre-trained architecture	Metric	RF	MLPC	Naïve Bayes	Decision tree
Mobile Net	Accuracy	97.78	91.11	95.56	73.33
	Precision	0.98	0.91	0.96	0.73
	Recall	0.98	0.93	0.96	0.76
	F1-score	0.98	0.91	0.96	0.74
DenseNet121	Accuracy	97.78	**100**	64.44	**82.22**
	Precision	0.98	1.00	0.64	0.82
	Recall	0.98	1.00	0.83	0.88
	F1-score	0.98	1.00	0.63	0.82
Xception	Accuracy	**97.78**	86.67	84.44	**82.22**
	Precision	0.98	0.87	0.84	0.82
	Recall	0.98	0.90	0.89	0.83
	F1-score	0.98	0.87	0.84	0.81
InceptionV3	Accuracy	88.89	**100**	53.33	73.33
	Precision	0.89	1.00	0.53	0.73
	Recall	0.91	1.00	0.77	0.76
	F1-score	0.89	1.00	0.49	0.73
EfficientNetB0	Accuracy	97.78	97.78	**100**	75.56
	Precision	0.98	0.98	1.00	0.76
	Recall	0.98	0.98	1.00	0.76
	F1-score	0.98	0.98	1.00	0.76

Note: Macro average values of precision, recall, and F1-score are shown. Macro average values are calculated as the arithmetic mean of individual classes' scores.

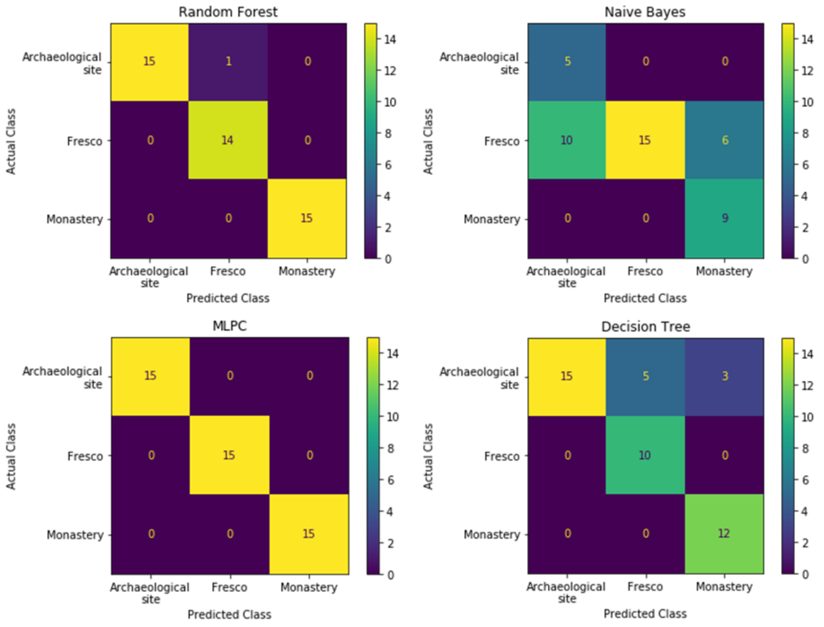
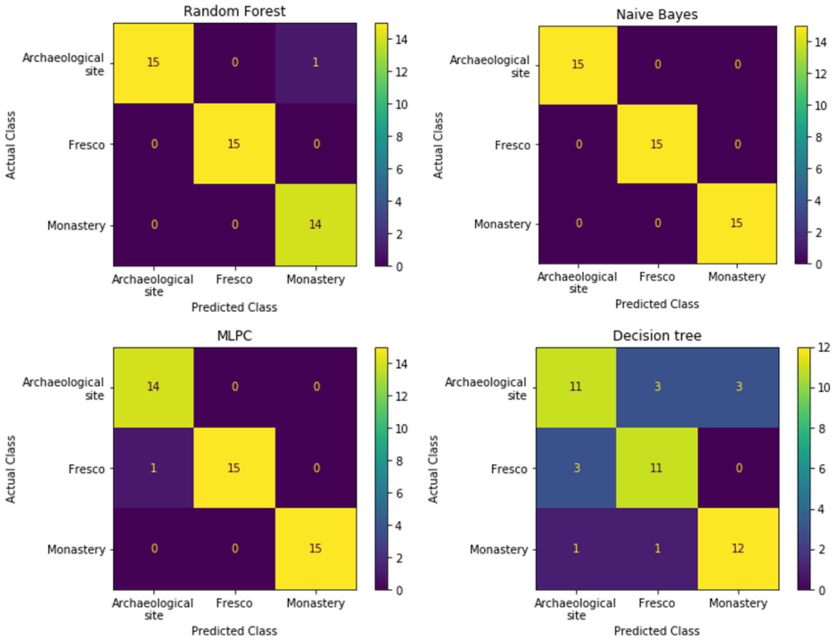


Figure 2: Confusion matrices obtained from traditional ML classification when DenseNet121 architecture was used for feature extraction

Considering the EfficientNetB0 architecture, Naïve Bayes correctly classified all the images from test set, while RF incorrectly classified only one image belonging to the class of archaeological sites as monastery, and MLPC incorrectly classified one image of fresco as archaeological site. However, the Decision tree incorrectly classified three images belonging to the fresco class as archaeological sites, one image of monastery as archaeological site, three images of archaeological sites as fresco, one image of monastery as fresco, and three images of archaeological sites as monastery (Figure 3).



Random Forest	Archaeological site	Fresco	Monastery
Archaeological site	15	0	1
Fresco	0	15	0
Monastery	0	0	14

Naive Bayes	Archaeological site	Fresco	Monastery
Archaeological site	15	0	0
Fresco	0	15	0
Monastery	0	0	15

MLPC	Archaeological site	Fresco	Monastery
Archaeological site	14	0	0
Fresco	1	15	0
Monastery	0	0	15

Decision tree	Archaeological site	Fresco	Monastery
Archaeological site	11	3	3
Fresco	3	11	0
Monastery	1	1	12

Figure 3: Confusion matrices obtained from traditional ML classification when EfficientNetB0 architecture was used for feature extraction.

When using the InceptionV3 architecture, the best performing model is MLPC as it classified all images from the test set correctly. RF incorrectly classified four images of archaeological sites as fresco, and one image of monastery as archaeological site. Furthermore, the decision tree algorithm misclassified two images of archaeological sites as frescoes, three images of frescoes as archaeological sites, five images of frescoes as monastery, and two images of monasteries as archaeological sites. Finally, the Naïve Bayes misclassified one image of archaeological sites as fresco, seven images of frescoes as archaeological site, and 13 images of fresco as monastery. Hence, MLPC is clearly the best choice when using InceptionV3 architecture for feature extraction, while Naïve Bayes is the worst choice as it misclassified most images belonging to classes archaeological sites and monasteries (Figure 4).

Classification of features extracted using the MobileNet architecture obtained good results for most models. In particular, RF misclassified only one image of archaeological site as fresco, and Naïve Bayes incorrectly classified one image of fresco as archaeological site, and one image of fresco as monastery. MLPC misclassified one image of monastery as archaeological site, and three images of monastery as fresco. Finally, the decision tree algorithm misclassified one image of fresco as archaeological site, three images of monastery as archaeological site, six images of fresco as monastery, and two images of monastery as fresco (Figure 5).

When using the Xception architecture, all models misclassified at least one image from the test set. RF misclassified one image of fresco as archaeological site, while MLPC misclassified three images of monastery as archaeological site, and three images of monastery as fresco. The Naïve Bayes incorrectly classified six images of fresco as archaeological site, and one image of fresco as monastery. Finally, the decision tree incorrectly classified one image of archaeological site as fresco, three images of fresco as archaeological site, three images of monastery as archaeological site, and one image of monastery as fresco (Figure 6).

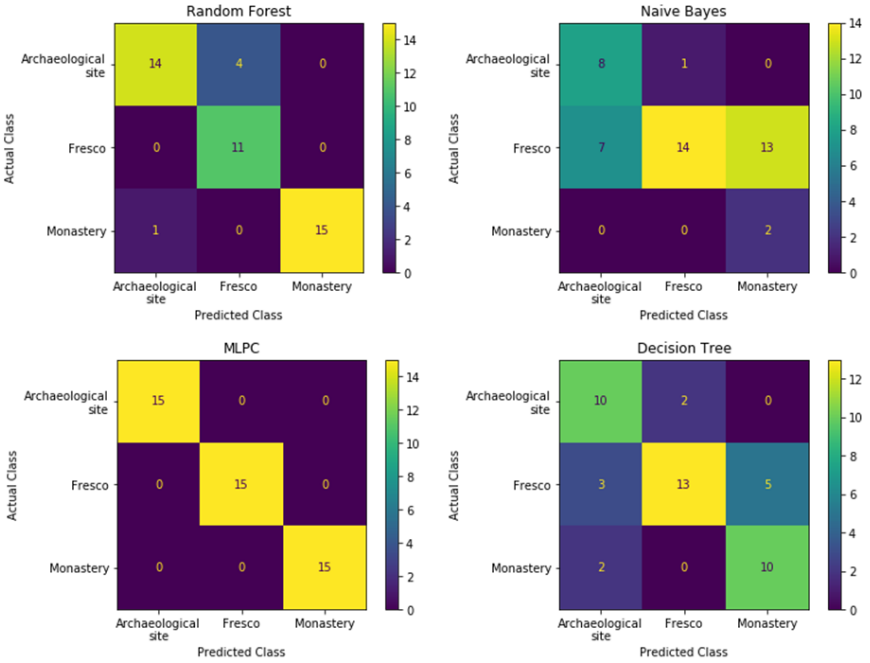


Figure 4: Confusion matrices obtained from traditional ML classification when InceptionV3 architecture was used for feature extraction.

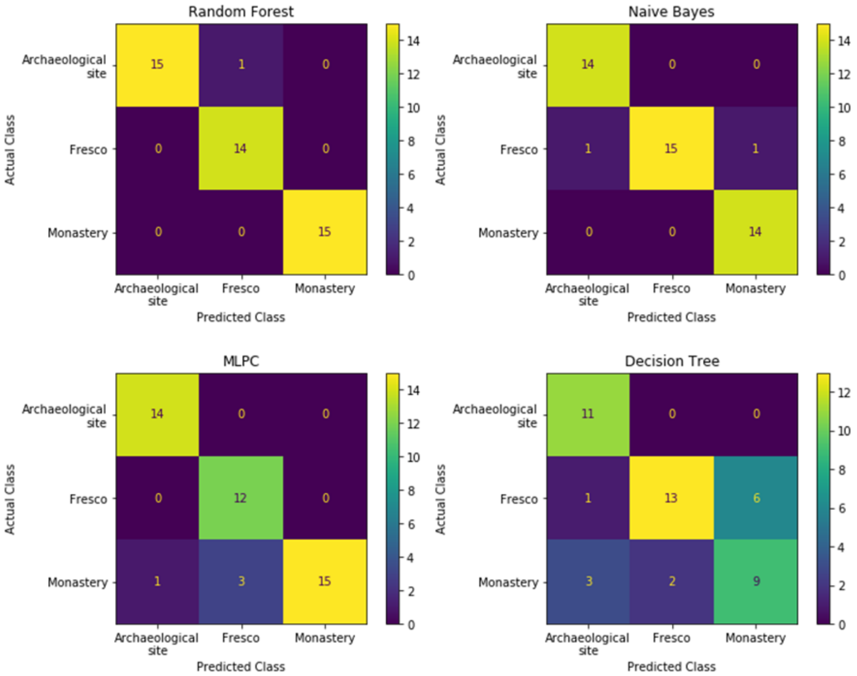


Figure 5: Confusion matrices obtained from traditional ML classification when MobileNet architecture was used for feature extraction.

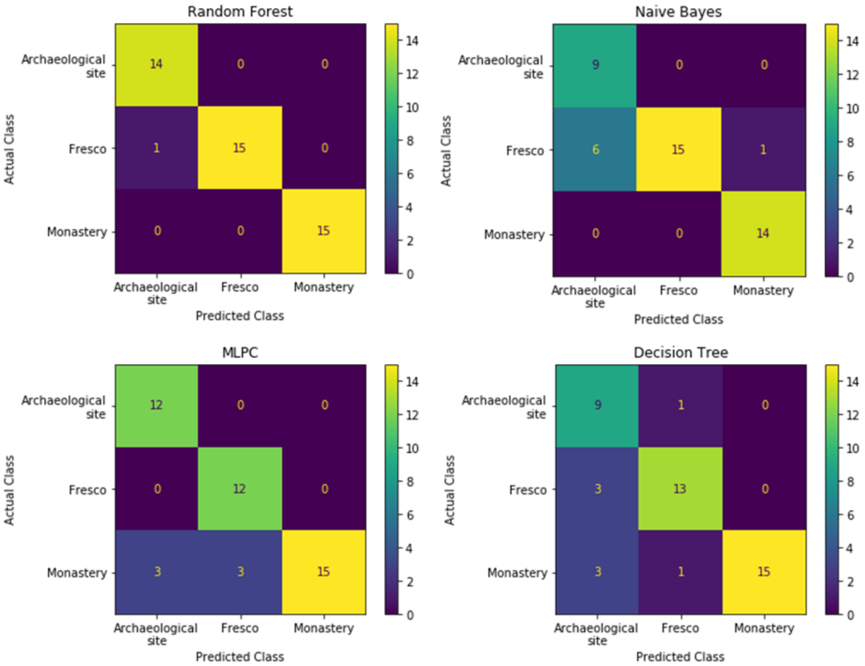


Figure 6: Confusion matrices obtained from traditional ML classification when Xception architecture was used for feature extraction.

These findings suggest the following. The RF algorithm obtained the best results when feature extraction was performed using MobileNet, DenseNet121, Xception, or EfficientNetB0. For these four pre-trained architectures, RF reached the same value of accuracy, precision, recall, and F1-score of 0.98. For MLPC, the highest values of performance metrics were obtained when features were extracted using DenseNet121 and InceptionV3 architectures. On the contrary, features extracted using Xception architecture and classified using the MLPC obtained accuracy of 87%. Naive Bayes obtained the highest accuracy of 100% when EfficientNetB0 architecture was used for feature extraction. However, when feature extraction was performed using the InceptionV3, the accuracy of Naive Bayes classification was only slightly above 50, i.e. 53.33%. Finally, considering the Decision tree, this algorithm performed the best when features were extracted using DenseNet121 and Xception architectures with accuracies of 82.22%. However, when using MobileNet, InceptionV3 and EfficientNetB0 architectures, performance is slightly weaker.

In terms of misclassified samples, the results show that in most cases the RF model misclassified the images of archaeological sites as fresco, suggesting that this algorithm is not capable to completely differentiate between the features that represent these classes. The Naive Bayes, on the other hand, frequently confused images of frescoes, classifying these as monastery or archaeological site. Furthermore, the MLPC made only several incorrect classifications, where it incorrectly classified images of monastery as archaeological site or as fresco. Finally, the decision tree was not able to successfully differentiate between the classes, as it made incorrect classifications in each class. Based on the above results, it can be concluded that RF and MLPC are the most suitable algorithms for classification of cultural heritage images when pre-trained architectures are used for feature extraction.

Comparing the obtained results to the results in [18], it can be concluded that using transfer learning approaches for feature extraction improves the performance of classification on small sets containing images of cultural heritage. In terms of precision, recall, and F1-score, the performance of RF in [18] reached 0.93, while in this study, when using transfer learning for feature extraction, the RF algorithm obtained precision, recall, and F1-score of 0.98 for four out of five pre-trained architectures. This is a significant improvement that confirms the suitability of transfer learning approaches for feature extraction.

4. Conclusion

This study was aimed at observing the differences in classification performance of traditional ML algorithms on a small set of cultural heritage images whose features were extracted using five pre-trained deep learning architectures. The dataset used in this study consists of only 150 cultural heritage images belonging to three classes (50 images per class). Feature extraction was performed using MobileNet, Dense-Net121, EfficientNetB0, InceptionV3, and Xception architectures, while classification was done using RF, Decision tree, MLPC, and Naïve Bayes algorithms.

The results suggest that the best performance was reached using RF, as well as MLPC algorithms, especially when the extraction of features was made using the DenseNet121 architecture. Although the differences in performance of RF and MLPC algorithms between the pre-trained architectures are not very extreme, the results of this study confirm the importance of a careful selection of feature extraction method. Finally, it should be noted that the decision tree obtained the lowest performance, with the differences between the pre-trained architectures ranging between 73.33% and 82.22% accuracy.

5. Acknowledgements

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia through the Mathematical Institute of the Serbian Academy of Sciences and Arts.

6. References

[1] M. Fiorucci, M. Khoroshiltseva, M. Pontil, A. Traviglia, A. Del Bue, S. James, Machine learning for cultural heritage: A survey, Pattern Recognition Letters 133 (2020) 102-108.

[2] A. Belhi, A. Bouras, A. K. Al-Ali, S. Foufou, A machine learning framework for enhancing digital experiences in cultural heritage, Journal of Enterprise Information Management (2020).

[3] S. Zhou, Y. Xie, Intelligent Restoration Technology of Mural Digital Image Based on Machine Learning Algorithm, Wireless Communications and Mobile Computing (2022).

[4] R. Hermoza, I. Sipiran, 3D reconstruction of incomplete archaeological objects using a generative adversarial network, in: Proceedings of Computer Graphics International, 2018, pp. 5-11.

[5] R. Janković, Machine learning models for cultural heritage image classification: Comparison based on attribute selection, Information 11 (2019).

[6] D. M. S. Arsa, A. A. N. H. Susila, VGG16 in batik classification based on random forest, in: International Conference on Information Management and Technology, ICIMTech, IEEE, 2019, pp. 295-299.

[7] A. J. Prieto, A. Silva, J. de Brito, J. M. Macías-Bernal, F. J. Alejandre, Multiple linear regression and fuzzy logic models applied to the functional service life prediction of cultural heritage, Journal of Cultural Heritage 27 (2017) 20-35.

[8] I. Wachtel, R. Zidon, S. Garti, G. Shelach-Lavi, Predictive modeling for archaeological site locations: Comparing logistic regression and maximal entropy in north Israel and north-east China, Journal of Archaeological Science 92 (2018) 28-36.

[9] L. Chen, J. Chen, Q. Zou, K. Huang, Q. Li, Multi-view feature combination for ancient paintings chronological classification, Journal on Computing and Cultural Heritage 10 (2017) 1-15.

[10] U. Kulkarni, S. M. Meena, S.V. Gurlahosur, U. Mudengudi, Classification of cultural heritage sites using transfer learning, in: Fifth international conference on multimedia big data, BigMM, IEEE, 2019, pp. 391-397.

[11] A. Belhi, A. Bouras, S. Foufou, Leveraging known data for missing label prediction in cultural heritage context, Applied Sciences 8 (2018) 1768.

[12] A. Yasser, K. Clawson, C. Bowerman, M. Lévêque, Saving cultural heritage with digital make-believe: machine learning and digital techniques to the rescue, in: Proceedings of the 31st British Computer Society Human Computer Interaction Conference, ACM Press, 2017, pp. 1-5.

[13] J. Llamas, P. M. Lerones, R. Medina, E. Zalama, J. Gomez-Garcia Bermejo, Classification of architectural heritage images using deep learning techniques, Applied Sciences 7 (2017) 1-26.

[14] M. Ćosović, R. Janković, CNN classification of the cultural heritage images, in: 19th International Symposium INFOTEH-JAHORINA, IEEE, 2020, pp. 1-6.

[15] A. Belhi, H. O. Ahmed, T. Alfaqheri, A. Bouras, A. H. Sadka, S. Foufou, Study and Evaluation of Pretrained CNN Networks for Cultural Heritage Image Classification, in: Data Analytics for Cultural Heritage, Springer, Cham, 2021, pp. 47-69.

[16] M. Sabatelli, M. Kestemont, W. Daelemans, P. Geurts, Deep transfer learning for art classification problems, in: Proceedings of the European Conference on Computer Vision Workshops, 2018.

[17] P. Navarro, C. Cintas, M. Lucena, J. M. Fuertes, C. Delrieux, M. Molinos, Learning feature representation of Iberian ceramics with automatic classification models, Journal of Cultural Heritage 48 (2021) 65-73.

[18] R. Jankovic, Classifying cultural heritage images by using decision tree classifiers in WEKA, in: Proceedings of the 1st International Workshop on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding Co-Located with 15th Italian Research Conference on Digital Libraries, IRCDL, 2019, pp. 119-127.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861, 2018.

[20] N. Hasan, Y. Bao, A. Shawon, Y. Huang, DenseNet convolutional neural networks application for predicting COVID-19 using CT image, SN computer science 2 (2021) 1-11.

[21] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2017, pp. 1251-1258.

[22] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105-6114.

[23] L. Breiman, Random forests, Machine learning 45 (2001) 5-32.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, Scikit-learn: Machine learning in Python, Journal of machine Learning research 12 (2011) 2825-2830.