Research Article

BI-DIRECTIONAL CLASSIFICATION OF ROMAN PERIOD COINS BY DEEP LEARNING METHODS

Authors: Kıyas KAYAALP, Fehmi ÖZKANER

To cite to this article: Kayaalp, K. & Özkaner, F. (2023). BI-DIRECTIONAL CLASSIFICATION OF ROMAN PERIOD COINS BY DEEP LEARNING METHODS . International Journal of Engineering and Innovative Research ,5(2),p161-169 . DOI: 10.47933/ijeir.1269680

DOI: 10.47933/ijeir.1269680

To link to this artice: https://dergipark.org.tr/tr/pub/ijeir/archive

International Journal of Engineering and Innovative Research

http://dergipark.gov.tr/ijeir

BI-DIRECTIONAL CLASSIFICATION OF ROMAN PERIOD COINS BY DEEP LEARNING METHODS

Kıyas KAYAALP[1*], Fehmi ÖZKANER[1]

[1]Isparta University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, Isparta, Turkey.

*Corresponding Author: kiyaskayaalp@isparta.edu.tr
(**Received:** 23.03.2023; **Accepted:** 27.05.2023)

https://doi.org/10.47933/ijeir.1269680

Abstract: In this study, the problem of classification of coins, which have historical importance and can only be distinguished by experts, is discussed with pre-learning deep learning algorithms. In the solution to the problem, the RRC-60 dataset, which consists of the images of the coins used in the Roman Republic period, was used. In this study, pre-learning Xception, MobileNetV3-L, EfficientNetB0, and DenseNet201 models were trained using the images on both sides of the coins in the data set. As a result of the training, the best values, Precision, Recall and F1-Score metrics in the MobileNetV3-L model were 98.2%, 96.8%, and 97.5%, respectively, and the test accuracy was 95.2%.

Keywords: Classification, CNN, Coin, Deep Learning.

1. INTRODUCTION

Money has been used as metal and paper since its first invention until today. Different metals are preferred for metal coins. When the coins are examined, a lot of information can be obtained about the period and the region used. When comparing today's coins with the coins used in the past, the old ones are more valuable because of the metals used. The value of these precious coins varies according to the century and the metal used. The determination of which century the coins belong to is carried out by the evaluation of experts in the field. In states that shaped history such as the Roman and Ottoman Empire, each sultan minted coins with his own figures such as seals, signatures, turra, or pictures when he ascended the throne. Therefore, it is very difficult to examine and classify the coins belonging to these periods.

Deep learning architectures, on the other hand, are used in many different fields, from the classification of images to military fields, from medical applications to education. For example; By examining the leaf image of the apple plant with deep learning methods, many examples can be given, such as detecting the disease in the plant [1], and classifying galaxies in space [2].

Some of the studies in the literature regarding the classification of coins belonging to ancient periods are as follows. Temiz et al. brought to the literature a data set consisting of 11080 coin images, consisting of 138 different classes and printed from 1924 to 2019. The authors, who used the ResNet50 deep learning algorithm in their work, achieved a 97.71% money recognition success rate by using the StyleGAN2 method for data duplication [3]. Aslan et al., in their study using the RRC-60 dataset consisting of 6000 images, obtained a 97.00% coin classification success rate with semi-supervised learning methods called Graph Transduction Games [4]. Yılmaz et al. used the RRC-60 dataset to classify the reverses of Roman Republic coins. In the study, 91%, 91%, and 83% classification success rates were obtained with the pre-learned ResNet152, VGG16, and Inception deep learning models, respectively [5]. Anwar et al. illustrated a graphical user interface by classifying Roman coins into 60 classes and 600 images. In this study, a 98% classification success rate was obtained by using the reverse side of the coins [6]. Ma and Arandjelović reached an average classification performance of 74.25% with a tone-based random forest classifier in a study using 400 Roman coins used for four different denominations during the reign of the Roman emperor Dominican [7]. Anwar et al. developed a special neural network model called CoinNet for the data set consisting of 228 object classes. In the study, they obtained 81.33% classification success by separating the data for 70% train and 30% testing [8].

As the application of machine learning and computer vision in ancient coin analysis (mostly Ancient Roman coins) is a relatively new but rapidly growing field of research, this study addressed the problem of the Classification of Ancient Roman Republic Coins. For this purpose, a classification study was carried out by using pre-learning EfficientNet, DenseNet201, Xception, and MobileNetV3 deep learning models.

2. MATERIAL AND METHODS

3.1. Image Dataset

In the study, the first data set was determined as RRC-60 [9]. A data set was created from coins drawn with a white background. The data set was filed separately as Observe and Reverse.

The purpose of Coinage Roman Republic Online (CRRO) is to offer an online version of Michael Crawford's Roman Republic Coin (RRC) publication from 1974, which is still the main typology used to describe different kinds of Roman Republican coins. Significant dating changes have been made since the series' 1974 publication as a result of the discovery of new successions, but no effort has yet been made to update the published typology to take these changes into account or make any other modifications. The descriptions of these coins were altered to comply with the requirements of the collection management system used by the British Museum, but they are still based on the type described in the RRC. Richard Witschonke of the American Numismatic Society (ANS) added new species to this catalog that are not part of the British Museum's holdings [10]. There are 60 classes in the data set where sample images are given in Figure 1.

Figure 1. Sample images from the RRC-60 dataset.

The time span of the coins in the dataset is 145 B.C. - 7 B.C. The coins here are Aces, Quadrans, Semis, Sextans, Triens, and Uncia, with the majority being dinars (denarius). The images of the gods Dioscuri, Hercules, Janus, Menerva, Mercury, Rome, and Saturn were embroidered on the coins, in which silver and bronze metals were used as materials. The starting point for the Roman Republic coins in the British Museum collection is the data set used in the research. These can be found in an online special catalog that was created in 2010 as a supplement to the 1910 catalog of Grueber's two collections. The Open Database License makes the coin type data accessible.

Performance evaluation is presented in the RRC-60 using gradually increasing size training sets and primarily training, validation, and test set sections were created (80%, 10%, and 10%). The data set was obtained from images consisting of 12000 RRC coins. The data set consists of 6000 Observes, 6000 Reverses.

In the experimental study for the classification of deep learning-based Roman coins, a data set consisting of 60 classes, 6000 observe and 6000 reverse images was used. As shown in Table 1, by applying the random selection process, 80% of the data set is divided into training, 10% validation and 10% test data.

Table 1. Training, validation and testing distribution of the dataset

Category	Remarks	Image	Distribution
	Train	4800	80%
RRC60-Observe	Validation	600	10%
	Test	600	10%
	Train	4800	80%
RRC60-Reverse	Validation	600	10%
	Test	600	10%

2.2. Deep Neural Networks

The number of deep learning methods used in many different fields, especially in the defense industry, health, space research and agriculture, and the number of studies used to classify images and detect objects from within the image is increasing day by day. Many deep learning algorithms such as Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM), Autoencoder (AE), and Deep Belief Networks (DBN) are used and new ones are brought to the literature by researchers every day. Especially in the classification of images, CNN-based models give more successful results. For this purpose, EfficientNet, DenseNet201, Xception, and MobileNetV3 models, which are popular in this field, were preferred in the classification study of historical Roman coins and brief information about the models was given.

2.2.1. EfficientNet

To build up models simply but effectively, EfficientNet employs a method known as the compound coefficient. Rather than randomly scaling width, depth, or resolution, compound scaling scales each dimension equally with a given set of fixed scaling coefficients. Using the scaling method and AutoML, seven models of various sizes have been developed, most with cutting-edge accuracy and much higher efficiency than CNN. The EfficientnetB0 model was used in the study.

It is argued that scaling single dimensions helps improve model performance, while balancing the scale in all three dimensions (width, depth, and image resolution) best improves overall model performance, considering the variable resources available [11].

EfficientNet is based on the underlying network developed by neural architecture search using the AutoML MNAS framework. The mesh is fine-tuned to achieve maximum accuracy, but is also penalized if the mesh is too computationally heavy. It also penalizes for slow inference time when the network takes too much time to make predictions. The architecture uses a mobile inverted bottleneck convolution similar to MobileNetV2, but much larger in size due to the increase in FLOPS. This base model is scaled up to achieve the EfficientNets family.

2.2.2. DenseNet-201

Each layer is immediately connected to the others in a feed-forward fashion by the DenseNet architecture. The network's aggregated data for DenseNet includes a very small collection of feature maps and a very thin layer (12 filters per layer). Densenet's benefits include illuminating the gradient issue, promoting attribute reuse, and minimizing the number of parameters required to perform various functions [12].

A 201-layer CNN is called DenseNet-201. It loads a pre-trained mesh that has been learned on more than a million images using the ImageNet database. The network divides images into more than a thousand distinct object groups, including keyboards, mice, pens, and a wide range of animals. The network has therefore amassed vast feature representations for a variety of image types. The network's image entry size is 224×224. We demonstrate that each layer includes convolution with a filter, ReLU activation, and bulk normalization (BN). The batch normalization step, which helps prevent over-learning during training, is the next step after each block receives an input in the shape of a matrix corresponding to an image pixel. ReLU activation for a value change if it is negative, but not if it is positive or vice versa. A preprocessed matrix value is produced by multiplying a matrix image that has successfully completed the ReLU activation phase by the convolution matrix using a filter [13].

2.2.3. Xception

It is possible to fully deconstruct the mapping of inter-channel correlations and spatial correlations in CNN feature maps. The suggested architecture is known as Xception, which stands for "Extreme Inception," because it is a more robust version of the underlying hypothesis of the original architecture.

The feature extraction basis of the network in the Xception architecture is composed of 36 convolutional layers. With the exception of the first and last modules, all of the 14 modules made up of the 36 convolutional layers have linear residual links surrounding them. In essence, the current Xception architecture is a linear stack of highly separable convolution layers connected together [14].

2.2.4. MobileNetV3

Recently, with the increase in portable devices and devices, transfer learning-based algorithms have developed. One of them is MobileNet based architectures. There are three different versions of the MobileNet architecture. One of them is the MobileNetV1 architecture. The MobileNetV2 architecture has been proposed with some changes such as the bottleneck structure in the MobileNetV1 architecture. Another recently developed architecture based on MobileNet architectures is MobilnetV3 architecture. There are also two different versions of the MobileNetV3 architecture. The first of these is the MobileNetV3 Small and the other is the MobileNetV3 Large architecture. MobileNetV3 architecture is assisted by search optimization algorithms of NAS and NetAdapt networks. At the same time, it uses the h-swish activation function instead of the ReLU activation function, which is predominantly used in the MobileNetV2 architecture. The Swish function is an activation function like ReLU [15].

2.3. Performance Evalution

When creating a CNN model, it is important to use the most appropriate loss function and to choose the right parameters for this loss function, and it directly affects the performance and the result of the training. The loss function is also called the cost function or the error function. In the training phase loss functions, a regulation is used to prevent over-learning, in other words, it has an important role in over-fitting the loss functions. The loss process is the stage that compares the test data and the predicted output values and measures the performance of the modeling in the training stage of the neural network. The training aims to minimize the loss between targets and forecast outputs and is updated to minimize the average loss with hyperparameters.

In a typical data classification problem, the evaluation metric is used in two phases, the training phase (learning process) and the testing phase. To put it another way, the evaluation metric is used to identify and choose the best answer that will result in a more precise estimate of the future evaluation of a specific classifier. The evaluation metric, meanwhile, was employed as an evaluator to assess the performance of the classifier created in the testing step when tested with hidden data. In Figure 2, the columns in Figure 2 show the actual class, while the rows represent the predicted class. From this confusion matrix, true positive (TP) or true positive and true negative (TN) or true negative indicates the number of correctly classified positive and negative samples. Meanwhile, false positive (FP) or false positive and false negative (FN) or false negative indicate the number of misclassified negative and positive samples, respectively.

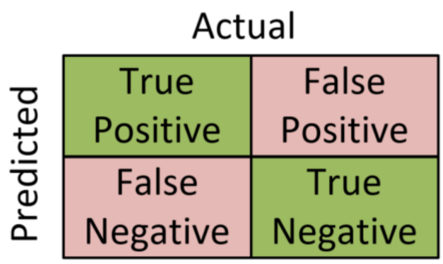

Figure 2. Confusion matrix

3. RESULT AND DISCUSSION

3.1. Experimental Procedure

Model training of experimental studies, model performance evaluation, which was trained with Tesla P100 as GPU, was made over Google Colab. Trainings on the classification of coins belonging to the Roman Republic Period were conducted with the pre-learning deep learning models MobileNetV3, EfficientNetB0, Densenet201 and Xception. Images of 224×224×3 dimensions were transferred to the inputs of the deep learning models. During the training phase, the base layers of the models are frozen. The hyperparameters used for model training are given in Table 2. Logistic Regression was used in the tuning method of hyperparameters [15]. In the training of pre-learning models, 32 values from four different Batch size values, Adam from five different optimizers, Cross Entropy from two different loss functions, 100 from fifteen different Epochs, 0.2 from three different dropouts, and 0.001 as learning rate were used. The class value and activation function of the output layer of the models are also set as Softmax. Average training times of Xception, MobileNetV3-L, EfficientNetB0 and DenseNet201 models were measured as 59, 32, 33 and 71 minutes, respectively. After training the models, train and validation accuracies were obtained. Confusion matrix was used to evaluate the classification performance of test data.

Table 2. Training hyperparameters

Hyperparameter	Setting	Best Value
Batch size	16, 32, 64,128	32
Optimizer	Adam, Adamax, Nadam, Adagard, Adadelta	Adam
Loss function	Focal, Cross Entropy	Cross Entropy
Epochs	10, 20, 30, ….,150	100
Dropout	0.1, 0.2, 0.5	0.2
Learning rate	0.00001 - 0.01	0.001

4.2. Comparison of Performance of Deep Learning Models

Pre-learning Xception, MobileNetV3-L, EfficientNetB0 and DenseNet201 deep learning models were trained with the data set created for the study. Since there are 12000 images in the data set, no data augmentation was required. The accuracy and loss graphs obtained as a result of the training performed with the original data are given in Figure 3.

Lowest validation loss in training models; Values of 0.48 were obtained with EfficientNetB0, 0.34 with MobileNetV3-L, 0.46 with Xception and 0.22 with DenseNet201. The highest validation accuracy achieved in training models; EfficientNetB0 values 0.90, MobileNetV3-L 0.94, Xception 0.91 and DenseNet201 0.92 values.

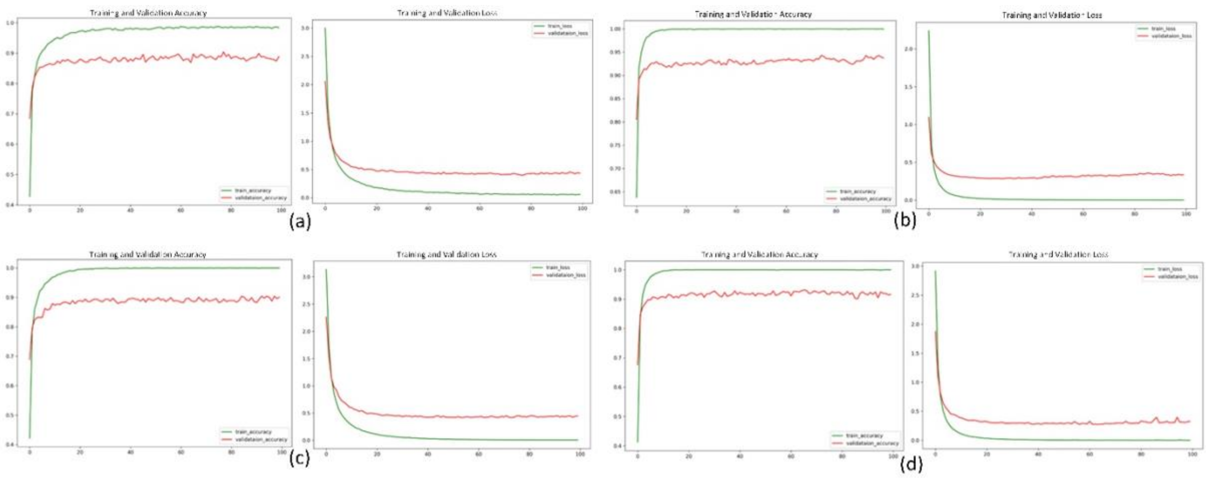

Figure 3. Training and validation loss-accuracy graphs of deep learning models (a)EfficientNetB0, (b)MobileNetV3-L, (c)Xception, (d)DenseNet201

Looking at the Precision values (Table 3) obtained from the test results of the models, it was observed that the highest value was obtained with MobileNetV3-L with 98.2% and the lowest value with the EfficientNetB0 model with 93%. According to the Recall metric, DenseNet201 has the best performance with 97.4%, while the EfficientNetB0 model has the lowest performance, as in Precision. When F1-Scores are examined, the highest value was obtained with MobileNetV3-L with 97.5%, and the lowest value was obtained with EfficientNetB0 model with a margin of 4.61%. After the training phase of four different pre-trained deep learning models, the highest success rate was achieved with the MobileNetV3-L model with 95.2%, according to the accuracy values obtained by testing the models with images that had not been seen before. In the study, where an average of 91.92% test accuracy was obtained, the weakest performance was obtained with EfficientNetB0, which was 4.45% below the average.

Table 3. Performance evaluation results

Metrics	EfficientNetB0	MobileNetV3	Xception	DenseNet201
Precision	93	**98.2**	95.8	96
Recall	93.4	96.8	94.8	**97.4**
F1-Score	93.2	**97.5**	95.3	96.7
Accuracy	88	**95.2**	91	93.5

The confusion matrix can be used to perfectly analyze the potential of a classifier algorithm. The first diagonal in the confusion matrix shows the correctly classified results. Therefore, the values of the elements outside the first diagonal of the algorithm with the best classification performance are expected to be zero or close to zero. In the study, the confusion matrix obtained from the MobileNetV3-L model, which gives the best test accuracy as a result of the trainings made with the pre-learning deep learning models, is given in Figure 4.

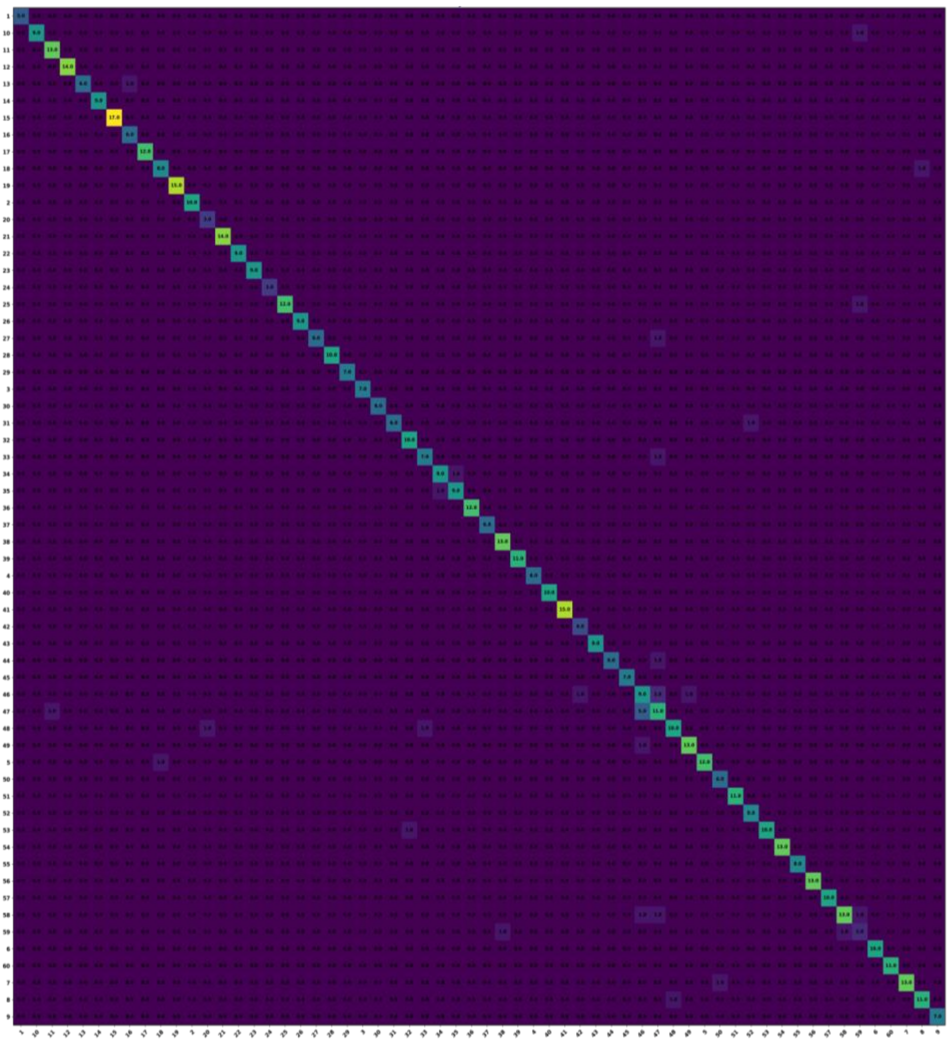

Figure 4. Confusion matrix from the MobileNetV3 model

When the confusion matrix is examined, it is observed that the coins with class numbers 46 and 47 are less successful in classifying than other labels. In the experimental study carried out, P. Sepullius Macer, BC. 44 (46) and P. Sepullius Macer, BC. 44 (47) coins are misclassified because the reverse motifs of the images are very similar (Figure 5).

Figure 5. Coins with class numbers 46 and 47

In Table 4, the previous studies using the data set used in the study and the differences of the proposed study are summarized. In the studies in the literature, classification processes were carried out using only the reverse part of the coins in the data set, and in the study, both sides of the coins were used for classification. A higher success rate than the one obtained in the study was obtained only by Anwar et al. [6] and Aslan et al. [4]. Anwar et al. [6] studied only 600 images and one face, while Aslan et al. [4] used 6000 images, one face and one classification algorithm. In the study, 12000 images, bidirectional and four different classification algorithms were used.

Table 4. A summary of the approaches used in the literature for the classification of Roman period coins

References	Techniques used	Side	Image No	Classes	Accuracy (%)
Aslan et al. [4]	ResNet152	Reverse	6000	60	97.00
Yılmaz et al. [5]	VGG-16, ResNet, Inception	Reverse	6000	60	91, 91, 83
Anwar et al. [6]	Image-based framework	Reverse	600	60	98
Ma and Arandjelović [7]	Hue-based representation and A random forest classifier	Reverse	400	4	74.25
Anwar et al. [8]	CoinNet	Reverse	18000	228	81.33
Ours	EfficientNetB0, Xception, DenseNet201, MobileNetV3	Reverse-Observe	12000	60	88, 91, 93.5, **95.2**

5. CONCLUSION

In daily life, people can only recognize coins in circulation. Numismatists and collectors related to the subject, on the other hand, can recognize coins with historical value with years of experience. In this study, coin classification was made using Xception, MobileNetV3-L, EfficientNetB0 and DenseNet201 pre-learning models used in image classification. In the classification process, the RRC60 dataset, which contains the coins used in the Roman Republic period, was used. Approximately 12000 images belonging to 60 classes in the dataset were divided into datasets for training, validation and testing of four different pre-learning models. The MobileNetV3-L model was the model that completed the 100 iterations of the training in the shortest time possible. Looking at the loss rate and Accuracy graphs, Densenet201 and MobileNetV3-L models became ideal in a very short time. In the performance evaluations of the models, only the EfficientNetB0 model was below 90% in the accuracy of the test data. Considering the confusion matrix of the MobileNetV3-L (95.2%) model, which has the most successful classification test accuracy, it is seen that the model's classes 46 (P. Sepullius Macer, BC 44) and 47 (P. Sepullius Macer, BC 44) are classified as their own. mixes between In future studies, it is aimed to develop deep learning-based web and smartphone applications for coin recognition, which will facilitate numismatists and collectors.

REFERENCES

[1] A. I. Khan, S. M. K. Quadri, S. Banday, and J. L. Shah, "Deep diagnosis: A real-time apple leaf disease detection system based on deep learning," *Comput. Electron. Agric.*, vol. 198, p. 107093, 2022.

[2] F. Şenel, "A Hyperparameter Optimization for Galaxy Classification," *Comput. Mater. Contin.*, vol. 74, no. 2, 2023.

[3] H. Temiz, B. Gökberk, and L. Akarun, "TurCoins: Turkish republic coin dataset," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 2021, pp. 1–4.

[4] S. Aslan, S. Vascon, and M. Pelillo, "Two sides of the same coin: Improved ancient coin classification using Graph Transduction Games," *Pattern Recognit. Lett.*, vol. 131, pp. 158–165, 2020.

[5] A. G. YILMAZ, Ç. SAYGILI, and V. NABİYEV, "Roma Dönemi Madeni Paralarının Derin Öğrenme Tabanlı Sınıflandırılması."

[6] H. Anwar, S. Sabetghadam, and P. Bell, "An Image-Based Class Retrieval System for Roman Republican Coins," *Entropy*, vol. 22, no. 8, p. 799, 2020.

[7] Y. Ma and O. Arandjelović, "Classification of ancient roman coins by denomination using colour, a forgotten feature in automatic ancient coin analysis," *Sci*, vol. 2, no. 2, p. 37, 2020.

[8] H. Anwar, S. Anwar, S. Zambanini, and F. Porikli, "Deep ancient Roman Republican coin classification via feature fusion and attention," *Pattern Recognit.*, vol. 114, p. 107871, 2021.

[9] G. Ciaburro and B. Venkateswaran, *Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles*. Packt Publishing Ltd, 2017.

[10] M. Crawford, "Coinage of the Roman Republic Online," 2021. http://numismatics.org/crro/.

[11] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[13] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning," *J. Biomol. Struct. Dyn.*, vol. 39, no. 15, pp. 5682–5689, 2021.

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

[15] A. Howard *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.