IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 November 2024, accepted 19 December 2024, date of publication 23 December 2024,
date of current version 31 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3521319

RESEARCH ARTICLE

Oracle Bone Inscription Character Recognition Based on a Novel Convolutional Neural Network Architecture

CHRISTOPHER MAI, PASCAL PENAVA, (Member, IEEE), AND RICARDO BUETTNER, (Senior Member, IEEE)

Chair of Hybrid Intelligence, Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg, 22043 Hamburg, Germany

Corresponding author: Ricardo Buettner (buettner@hsu-hh.de)

This work was supported by the Open-Access-Publication-Fund of the Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg.

ABSTRACT Oracle bone inscriptions (OBIs) are one of the oldest characters in the world and are the predecessors of today's Chinese characters. These oracle characters recorded various human activities of the time and provide insights into Chinese history. To date, almost 4,500 different oracle characters have been discovered, with deciphering still being carried out by people with specialist knowledge. This process is labor-intensive and time-consuming, with around 2,300 characters still to be deciphered. Furthermore, the inscriptions have become increasingly illegible as a result of the aging process, frequently exhibiting characteristics such as noise or incompleteness. To address these issues, in this paper, we present a new convolutional neural network architecture for recognizing OBIs. It is based on the idea of Inception modules and the use of residual connections. To increase the diversity in the dataset, data augmentation techniques were applied. Together with these techniques, the presented architecture achieves an accuracy of 95.93%. For the purpose of comparability, known pre-trained architectures such as InceptionV3, ResNet50, and Inception-ResNet-V2 were used for comparison. The results demonstrate that the proposed architecture exhibits superior performance compared to these models across multiple evaluation metrics while simultaneously establishing a new benchmark on the Oracle-MNIST dataset.

INDEX TERMS Oracle character recognition, oracle bone inscriptions, convolutional neural networks, image classification, deep learning.

I. INTRODUCTION

Oracle bone inscriptions (OBIs) are one of the world's oldest characters [1]. In the early stages of hieroglyphic development, oracle characters were engraved on cattle bones or turtle shells approximately 3,000 years ago to facilitate divination [2]. These oracle characters recorded various human activities of the era, thereby establishing the foundation for the study of Chinese etymology and providing insight into China's history [3]. OBIs are essentially the precursor to the modern Chinese characters that are widely used across Asia to this day [4]. Therefore, research on oracle characters is essential for understanding Chinese etymologies and calligraphy. It also provides valuable insights into the culture and history of ancient China and the world at large [3]. In 2017, OBIs were chosen for inclusion in the UNESCO "Memory of the World Register", this shows that the cultural and historical value of OBIs is recognized worldwide [5]. To date, almost 4,500 different oracle characters have been discovered, but only around 2,200 characters have been successfully deciphered [6]. This means that there are still characters to be deciphered. Most oracle characters are preserved in scanned images, created by reproducing the oracle bone surface through the technique of placing a piece of paper over the subject and rubbing it with rolled ink [6], [7]. These scanned images of oracle bones often suffer from low quality due to abrasion, varying noise levels, uneven sizes, damaged characters, and uneven tilt [8]. In addition, images vary in

The associate editor coordinating the review of this manuscript and approving it for publication was Fang Yang.

contrast and brightness, and characters are often rotated and distorted, making unsupervised domain alignment methods insufficiently robust and limiting improvement [1]. There are relatively few documents describing OBIs, and only a handful of specialists are able to read them [4]. At present, the recognition and deciphering of oracle bone characters is mainly based on the manual method by experts with a high level of expertise, which is a very time-consuming and labor-intensive task [9].

With the rapid advancements in deep learning, leveraging computers to automate the detection and recognition of OBI characters has emerged as a highly promising solution for OBI decipherment. Figure 1 shows the disadvantages of the present manual deciphering method as well as a future, more efficacious approach that employs modern deep learning techniques.

Meng et al. [7] used a Single Shot Multi-box Detector (SSD), which uses the VGG16 architecture as the base network for the feature map for OBI detection. In [3], Wang, Deng, and Liu employed a structure-texture separation network to transport knowledge from labeled hand-printed oracle data to the unlabeled scanned domain. However, their approach yielded an accuracy rate of less than 50% when applied to scanned oracle characters. In a related study [1], Weng, Deng, and Sung employed an unsupervised discriminative consistency network that enhances augmentation consistency and discrimination between classes in the target area. Their approach yielded a success rate exceeding 60% on scanned image data. Qiao and Xing [10] used a ResNet50 model for feature extraction with unsupervised domain matching and pseudo-labels for text recognition and transcription, enhancing robustness by integrating GANs to generate challenging sample data. In the work by Liu et al. [11], five classical network models (AlexNet, VGG19, InceptionV3, SqueezeNet, and ResNet) were selected to compare their effectiveness in OBI detection using an OBI dataset containing incomplete characters. Fu et al. [12] used the three convolutional neural networks (CNNs), LeNet, AlexNet, and VGGNet, as the basis for recognizing oracle characters. These standard architectures were modified in different ways, resulting in ten different variants. The findings of their study demonstrated that OBIs can be effectively and practically identified within the context of deep CNNs.

CNNs have been shown to possess the architectural properties necessary for achieving strong performance in oracle character recognition. While some studies focus on datasets featuring manually crafted (hand-printed) oracle characters, datasets containing scanned oracle characters are more representative of real-world conditions and thus offer greater authenticity. Moreover, it demonstrates, specifically within the context of supervised learning, that much of the prior research has primarily relied on pre-built architectures, which, although effective to some degree, do not adequately address challenges such as high noise levels or substantial damage to the oracle characters. Consequently, there remains a need to improve future architectures for real-world oracle character recognition, as these are often characterized by substantial noise caused by age or incompleteness.

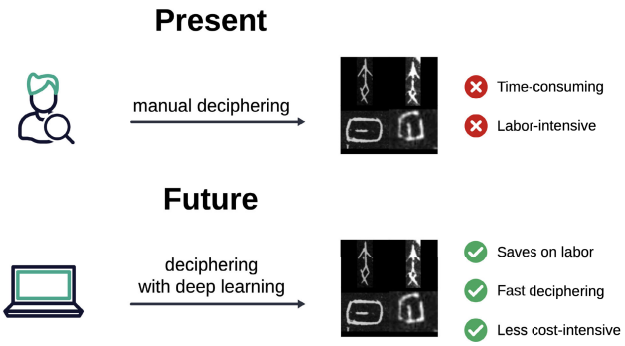


FIGURE 1. Schematic representation of the current manual process for deciphering oracle characters and a possible future improved process to simplify deciphering through the use of deep learning methods.

Given the great importance of oracle character recognition, we address this issue and propose a novel architecture designed to achieve high accuracy in identifying oracle characters in images, even when they are severely affected by noise, low resolution, or partial incompleteness. Our novel architecture is inspired by Inception-ResNet-V2 [13] and integrates the benefits of ResNet's residual connections with the Inception module concept, which combines multiple convolution layers with varying kernel sizes in parallel. This architecture has fewer parameters and achieves higher accuracy. Furthermore, data augmentation techniques are employed to simulate real-world scenarios, such as varying scaling or font variations, with the objective of enhancing the diversity of the dataset. To evaluate the performance of the new architecture, the Oracle-MNIST dataset presented by Wang and Deng [6] was selected. In order to create a reference to reality, only scanned oracle characters are employed. The dataset includes images of scanned OBIs, some of which are characterized by high levels of noise and substantial damage. This is an intentional aspect that the presented architecture is designed to address, as OBIs can also be of poor quality in reality. Furthermore, the dataset comprises an adequate number of images per class, thereby avoiding the issues of long-tailed distributions and highly imbalanced classes. Strong imbalances in the dataset cause the training of deep neural networks (DNNs) to be heavily biased towards head classes, preventing the learned model from achieving robust classification for tail classes [14]. To establish a connection with oracle characters in reality, it is essential to utilize only scanned images. In light of these limitations and the aforementioned reasons, other established datasets, such as OBC306 [15], Oracle-20K [14], and Oracle-AYNU [2], are not included in this study. Consequently, the primary contributions of this study are as follows:

1) It is shown that the application of data augmentation techniques as pre-processing steps leads to improved accuracy.

2) The proposed model was validated on the benchmark oracle dataset Oracle-MNIST. Experimental results show that the model can yield state-of-the-art performance as opposed to other existing oracle recognition methods.

The paper is structured as follows: Section II provides an overview of methods for recognizing oracle characters and discusses related work. Section III contains information about the deep learning approach used, a brief description of the deep learning architectures used, and the methods used to pre-process the images, followed by a detailed explanation of the training process and the dataset used for evaluation. In Section IV, the results are presented, followed by the discussion in Section V. Finally, Section VI wraps up the paper by summarizing the key findings and highlighting the study's contributions.

II. RELATED WORK

A. ORACLE CHARACTER RECOGNITION

One of the most crucial aspects of oracle bone studies is the identification and deciphering of oracle characters [3]. As technology advances, several studies have started to focus on recognizing oracle characters through the lens of computer vision. Early studies are based on classical pattern recognition methods. Lv et al. [16] proposed a Curvature Histogram-based Fourier Descriptor to classify oracle characters. In the work [17], Li et al. proposed a method to recognize oracle inscriptions based on graph isomorphism. Liu and Liu [18] used support vector machine (SVM) and block histogram-based features to recognize characters. Meng [19] developed a system for detecting OBIs and proposed a detection method using the Hough transform. The experimental results show that almost 80% of inscriptions are recognized. Another paper by Meng [20] proposed a two-stage detection system consisting of line points and checkpoint detection to detect OBIs. At best, 90% of the inscriptions are recognized with this method. Guo et al. [14] collected the Oracle-20K dataset, consisting of 20,000 hand-printed oracle characters across 261 categories, and developed an innovative hierarchical representation that integrates a Gabor-based low-level representation with a sparse-encoder-based mid-level representation. However, most traditional approaches depend heavily on hand-crafted features, making them suitable only for small-scale and clean datasets [21].

B. DEEP LEARNING METHODS

The versatility of deep neural networks is extensive. Several studies in recent years have demonstrated various methods for the recognition of oracle characters as well as Chinese script, which is considered the successor of oracle characters. Zhuo and Zhang [22] employ a deformable convolutional network with an attention mechanism to process Chinese characters from different dynasties. The essence of the attention mechanism is the efficient allocation of information processing resources, allowing the model to prioritize key points in an image by assigning them high weights, ignoring less relevant information with low, and dynamically adjusting these weights to focus on crucial details across varying contexts. In another study [23], the authors also employ a deformable convolutional network, where they address the issue of offset variation by normalizing the offsets generated within the network, effectively reducing variance. Additionally, they introduce a new data augmentation technique known as Matt data augmentation to further improve the recognition performance. To facilitate the identification of ancient Chinese scripts, Zheng et al. [24] proposed an enhanced Swin-Transformer model comprising four stages, each incorporating Linear Embedding and Swin-Transformer blocks. Each stage is further supplemented by a Context-Transformer Block to improve local feature extraction. The Context-Transformer Block strengthens the model's capability to capture local features effectively. According to the authors, this model progressively reduces the resolution of the input feature map while expanding the receptive field layer by layer. There are datasets containing both hand-printed and scanned real-world oracle characters. Models trained on hand-printed images frequently perform poorly when applied to scanned images. In their work, Wang, Deng, and Liu [3] investigate transferring knowledge from hand-printed oracle data to the scanned domain, utilizing a structure-texture separation network to achieve this. STSN decomposes features into structural and textural components through the use of generative models. By aligning handwritten and scanned data within the structural feature space, the mitigates the negative impact of significant noise during the matching process. This transformation is achieved by exchanging learned textures across the domains. A classifier is then trained for the final classification. With this approach, an accuracy of 95% is attained on hand-printed oracle characters, though accuracy drops to below 50% for scanned images. In a related study by Weng, Deng, and Sung [1], the authors also focused on the transfer of knowledge from labeled handwritten oracle characters to unlabeled scanned data. To achieve this, the researchers employed an Unsupervised Discriminative Consistency Network (UDCN), which was designed to enhance the consistency of augmentations and improve class discrimination within the target domain. Using this approach, they achieved an accuracy exceeding 60% on scanned oracle characters. In [25], the authors utilize an adaptive network, termed UFCNet, to detect oracle characters. This approach combines the Fourier transform with a convolutional attention fusion module (CAFM). The CAFM integrates both deep and shallow features to obtain more comprehensive information, while the Fourier transform converts the image from the spatial domain to the frequenting domain, aiming to capture the edge details of the Qiao and Xing [10] utilized a ResNet50 model for feature extraction, incorporating unsupervised domain matching that employs multiple pseudo-labels to facilitate efficient text recognition and transcription. To enhance the

model's repair capabilities, they combine ResNet50 with GANs that generate difficult-to-discriminate sample data, thereby strengthening robustness. Furthermore, a U-Net-based text repair model is employed to enhance performance through the integration of dense connectivity and spectral normalization.

C. DEEP LEARNING METHODS IN SUPERVISED LEARNING CONTEXT

In recent years, numerous researchers have successfully implemented deep learning techniques for oracle bone inscription character recognition, achieving excellent results [27]. Meng et al. [28] introduced an initiative to recognize OBIs by using a CNN architecture called AlexNet. They generated an OBI dataset composed of real rubbing images. In order to train sufficiently before recognition, data augmentation methods such as rotation, adding Gaussian noise, cropping, brightness change, and inversion were used. This method turned one image into 3,072 new images. They tested various hyperparameters to improve the recognition capability. They obtained an accuracy of 92.3%. Liu et al. [29] designed a CNN with five convolutional layers with a kernel size of 3×3 and two fully connected layers for classifying oracle bone inscription characters. They demonstrated the superiority of their method on a dataset containing 44,868 oracle bone inscription characters, achieving a top-1 accuracy of 91.56%. Huang et al. [15] trained variants of AlexNet, VGGNet, and ResNet to perform classification of scanned oracle data. For this, they used the dataset OBC306 presented in their study. All models were pre-trained using the ImageNet [30] dataset and then fine-tuned on the proposed training set. The ResNet50 architecture reached a top-10 accuracy of 83.00%. Zhang et al. [2] used a CNN with DenseNet block combined triplet loss and performed recognition by nearest neighbor classifier. They reached an accuracy of 83.37% on the Oracle-AYNU dataset and 92.43% on the Oracle-20K [14] dataset. In the work by Liu et al. [11], five classical network models such as AlexNet, VGG19, InceptionV3, SqueezeNet, and ResNet, were selected to compare their effectiveness in OBI detection. For the evaluation, they used a dataset with incomplete characters. With the SqueezeNet model, they achieved a top-5 accuracy of 94.38%. In another work [31], a VGG16 model was used, which has been pre-trained in the ImageNet dataset. The authors investigated the effects of transfer learning and fine-tuning on the model. In another work, Guo et al. [27] designed an improved neural network model based on InceptionV3 for OBI recognition. They applied the CNN to two datasets while also comparing the performance of well-known architectures such as AlexNet, InceptionV3, and VGG19. They achieved a top-5 accuracy of 99.844% on HWOBC [32] dataset. Fu et al. [12] used the three CNNs, LeNet, AlexNet, and VGGNet, to recognize oracle characters. These standard architectures served as the basis for modifications to the architecture, for example, three variants of AlexNet were created, which differed in the number of convolution layers and pooling layers. A total of ten variants of the models were created. With a modified VGGNet, they achieved an accuracy of 99.5% on their own dataset named OBI-100. In [6], Wang and Deng presented a new benchmarking dataset, the Oracle-MNIST dataset, which includes 28×28 px grayscale images of 30,222 ancient characters from ten categories. The CNN architecture, comprising two convolutional layers, a max pooling layer, and two fully connected layers, established the initial benchmark with an accuracy of 93.8%. This benchmark was subsequently surpassed by Ganj et al. [26], who achieved an accuracy of 95.13%. They used their architecture, which was influenced by residual connections and Inception modules. The existing literature demonstrates the efficacy of deep learning methodologies in oracle character recognition, particularly in the domain of recognizing OBIs. Some studies have concentrated on the recognition of images produced by hand [2], [27], [29], while others have focused on scanned images. However, the recognition of scanned oracle characters is of particular importance as they are more realistic. This is because an oracle character can also appear in a poor state in reality. Previous studies have demonstrated significant progress in the recognition of oracle characters from scanned images. However, there remains room for improvement in recognition performance. Notably, much of this prior research has relied on pre-built architectures, which, while effective to some extent, do not explicitly account for challenges such as high levels of noise or significant damage to the oracle characters. We address this issue with the distinctive characteristics of oracle characters by introducing a novel architecture designed to recognize OBIs accurately, even in the presence of damage, font variations, and noise. The Oracle-MNIST [6] dataset serves as an ideal benchmark for evaluating our model, as it closely reflects real-world conditions. It encompasses scanned oracle characters exhibiting diverse levels of noise, instances of significant damage, and natural variations in font styles. The application of the novel architectural methodology established a new benchmark for this dataset. Table 1 provides an overview of the works mentioned that used the Oracle-MNIST dataset, including the models employed and the accuracy values achieved.

TABLE 1. Summary of related work on the classification of OBIs using the Oracle-MNIST dataset.

Author	Year	Model	Accuracy
Wang and Deng [6]	2022	CNN (own architecture)	93.8%
Ganj et al. [26]	2023	CNN (own architecture)	95.13%

III. METHODOLOGY

In this section, the proposed architecture of a CNN specifically designed for the classification of OBIs is presented in

detail. This architecture is designed to focus on capturing the local features in oracle bone images that are crucial for accurate classification. To improve the extraction of these detailed features, the architecture mainly uses 3×3 convolutional kernels. These kernels were selected based on their effectiveness in detecting fine details in the images to ensure that the features of the inscriptions are captured. To evaluate and validate the performance of the proposed architecture, comparisons are made with state-of-the-art pre-trained CNN architectures. This section also deals with the training procedure for the proposed CNN architecture. The procedures and techniques for optimizing the performance of the model are also described here.

A. MODEL ARCHITECTURE

In this paper, an improved model for recognizing OBI characters is proposed. The architecture is inspired by that of Inception-ResNet-V2 [13]. The Inception module approach, which combines multiple convolutional layers with different kernel sizes together with residual connections, is employed. The proposed architecture is different from others. For example, instead of additive residual connections, it uses concatenative ones, which also have a special positioning. The architecture demonstrates distinct characteristics in the arrangement of its convolutional layers, which utilize only two specific kernel sizes. In addition, batch normalization is consistently applied following each convolutional layer. In contrast to traditional architectural designs, this configuration utilizes four consecutive fully connected layers with dropout layers between them, thereby enhancing the learning capabilities of features. The proposed architecture is primarily composed of two identical blocks based on the idea of Inception modules. Figure 2 provides a schematic illustration of the resulting architecture, including data augmentation techniques.

Before the images are processed, they undergo various transformations using data augmentation techniques. These methods help artificially increase the diversity of the dataset by applying random modifications such as rotations, zooming, and translation. After this augmentation, the images are fed into the first of two main blocks. In this paper, these blocks are referred to as Conv blocks, owing to their composition of multiple convolutional layers. Following these Conv-Blocks are four fully connected (dense) layers. In each convolutional layer and in each dense layer, except the last one, the ReLU activation function seen in equation (1) [33] with x as input value is used to add nonlinearity to the model.

$$ReLU(x) = max(0, x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The number of neurons in the dense layers is arranged in descending order, leading to the last layer, which contains ten neurons. In this last layer, a softmax activation function, shown in equation (2) [34] with k classes, is used to output the probabilities of the outputs of the ten classes.

$$Softmax(x_j) = \frac{e^{x_j}}{\sum_{i=1}^{k} e^{x_i}} \quad with\ j = \{1, \ldots, k\} \tag{2}$$

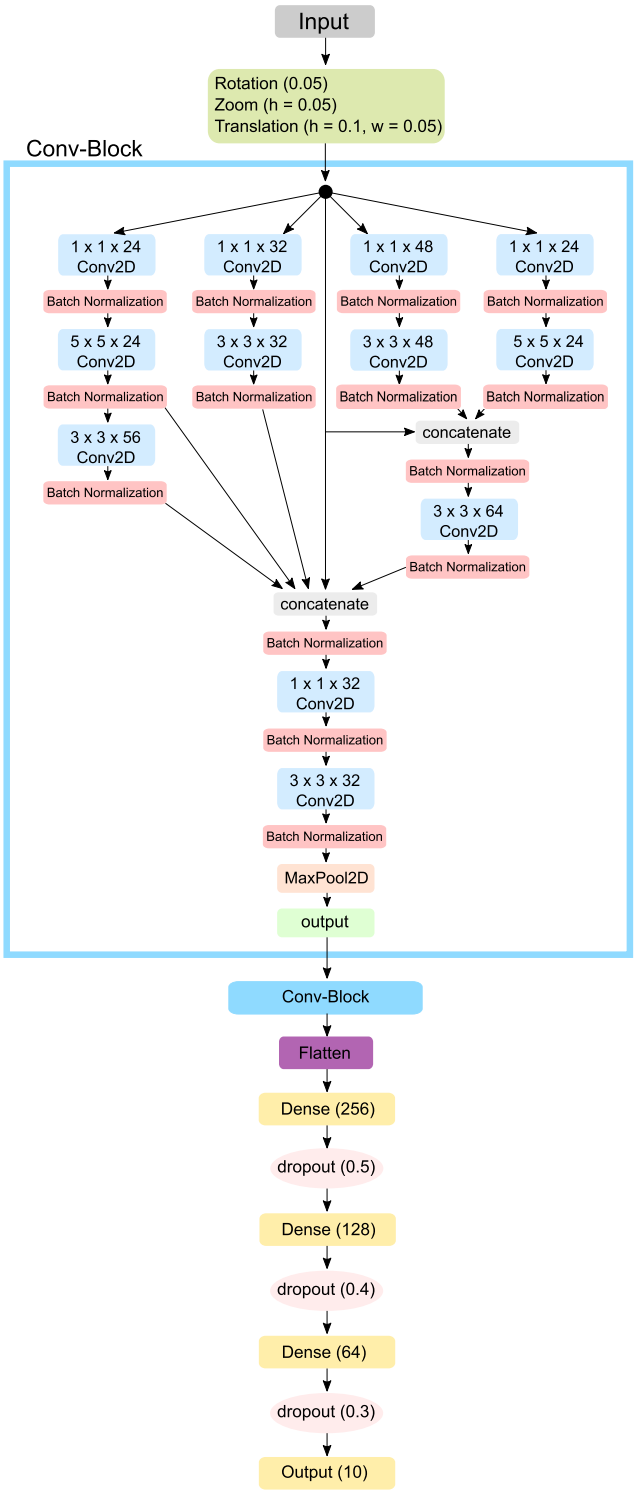


FIGURE 2. An illustration of the structure of the architecture with data augmentation techniques.

To reduce the risk of overfitting, dropout layers of different sizes are strategically placed between the dense layers. These dropout layers randomly deactivate some of the neurons during training, which improves the generalization of the model to new, unseen data. The use of 7×7 kernel, as found in Inception-ResNet-V2, was avoided due to the high computational overhead and ineffectiveness. In previous tests, 7×7 kernel were built into the architecture, but these compromised accuracy. Therefore, they were excluded to optimize performance.

In each part of a Conv-Block, the process starts with a 1×1 kernel, among other things, to integrate non-linearity into the process at an early stage. Next, on the left side of the block, a 5×5 kernel is applied to capture more global information from the input. Following this, a 3×3 kernel is used to extract details from the features obtained from the previous layer. Parallel to this process, as illustrated in the center of the structure, another pathway commences with a 1×1 kernel, which is directly followed by a 3×3 kernel. This parallel approach enables the simultaneous occurrence of multiple feature extraction processes, thereby enhancing the model's capacity to capture a diverse set of features. On the right side of the block, a 3×3 kernel is concatenated to a 5×5 kernel, combining the detailed and global information extracted from these kernels. The concatenation of the output of two convolutional layers $A \in \mathbb{R}^{H \times W \times C_A}$ and $B \in \mathbb{R}^{H \times W \times C_B}$ where C is the number of channels and H and W denote the height and width, can be described mathematically as follows [35]:

$$Concatenate(A, B) = C \in \mathbb{R}^{H \times W \times (C_A + C_B)} \tag{3}$$

The spatial dimensions H and W of the feature maps remain unchanged during concatenation. However, as the information from both layers is combined, the total number of channels increases accordingly. This additional step was specifically included to enhance the benefits of both kernel sizes. The 3×3 kernel is employed to extract more localized information, while the 5×5 kernel is utilized for the extraction of more expansive, global information. The concatenation of more than two outputs of convolution layers is also possible, which will also be implemented in a later step of the architecture. Subsequently, a further 3×3 kernel is employed to provide further detail on the extracted features. All feature sets are then concatenated. Finally, a 1×1 and 3×3 kernel follows to extract the details before the max-pooling layer selects the maximum values.

After each convolutional layer, a normalization technique known as Batch Normalization was applied to optimize the training parameters. The Batch Normalization (BN) layer begins by standardizing each feature within a mini-batch and subsequently learns a shared scaling factor (slope) and offset (bias) for the entire mini-batch [36]. Formally, given the input to a BN layer $X \in \mathbb{R}^{n \times p}$, where n denotes the batch size, and p is the feature dimension, BN layer transforms a feature $j \in \{1 \ldots p\}$ into [36]:

$$\widehat{x_j} = \frac{x_j - \mathbb{E}[X_{.j}]}{\sqrt{\mathrm{Var}[X_{.j}]}}, \quad y_j = \gamma_j \widehat{x_j} + \beta_j \tag{4}$$

where x_j and y_j are the input and output scalars corresponding to the response of a single neuron for a given data sample; $X_{.j}$ represents the j-th column of the input data; $\mathbb{E}[X_{.j}]$ the mean of $X_{.j}$ and γ_j, β_j are the parameters to be learned [36]. This normalization step not only speeds up the training process but also accelerates convergence by smoothing the loss surface, making it easier to find the global optimum during training [37]. After the convolutional layers and their corresponding batch normalization steps, a flatten layer was introduced. The flattening layer is used to convert the two-dimensional feature maps generated by the convolutional layers into a one-dimensional vector, which can then be processed by the following four fully connected layers for classification.

B. OVERVIEW OF PRE-TRAINED ARCHITECTURES FOR EVALUATION

In order to assess the efficacy of the proposed architecture, pre-trained architectures were trained with the Oracle-MNIST dataset for comparative purposes. Furthermore, these models utilize the weights of the ImageNet dataset. A concise overview of their architectures is provided below.

The degradation problem is a phenomenon that can occur when training deep neural networks. There comes a point where accuracy does not improve or degrades with increasing depth, even though more layers are added. This phenomenon is not due to overfitting, as adding more layers paradoxically increases the training error [38]. To solve this problem, He et al. [38] introduced an architecture named Residual Network (ResNet). ResNet has been proposed in different depths, such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152, where the numbers indicate the number of layers in each model. A residual block consists of a series of layers that are connected to each other by so-called shortcut connections. These connections allow activations to be passed directly from one layer to the next, improving the flow of information and the efficiency of modeling [38]. So, if the current layer is not needed, it can be bypassed thanks to this identity, which reduces overfitting issues [38]. The ResNet50 structure is as follows: first, there is a Convolutional layer with a 7×7 kernel, followed by a max-pooling layer of 3×3, both layers with a step size of 2. Various Residual blocks are then run through, with a structure of three convolutional layers with the kernel (1×1, 3×3, 1×1). The blocks differ according to the selected filters in the layers. Szegedy et al. [39], [40] introduced InceptionV3 as Google's third version of their Inception CNN. The idea of the Inception models is to use different kernel-sized filters for the convolution on the same level. Thereby, a variation in the location of information can be intercepted, and thus, overfitting problems sink, and classification results increase for images with distributed information. The Inception models thus increase accuracy not

by getting "deeper" but "wider" [39], [40]. The architecture of InceptionV3 consists of several convolutional layers, some of which are combined in so-called Inception modules. Compared to its predecessors V1 and V2, InceptionV3 is more efficient, less computationally intensive, has a deeper network, uses auxiliary classifiers as regulators, and factorized convolutions to break up large convolutions, e.g., a 7×7 convolution is broken down into two 3×3 convolutions [40]. In [13], Szegedy et al. introduced another variant of Inception, named Inception-ResNet-V2. This architecture uses the concept of Residual Connections to facilitate the training of deep networks and improve performance. The architecture comprises a stem module for pre-processing the input images, followed by three distinct types of Inception-ResNet modules ($5\times$A, $10\times$B, $5\times$C). These modules are interspersed with two reduction blocks that serve to diminish the spatial dimensions of the feature maps [13]. Specifically, the reduction modules are positioned after the Inception-ResNet modules A and B. The final classification is performed by a fully connected layer with a softmax activation function.

C. TRAINING PROCESS

Before the training process, the oracle images were divided into a training and a test set, as the dataset structure already suggests. Both sets were then normalized so that the pixel values were in the interval [0, 1]. Furthermore, the influence of data augmentation techniques such as zoom, rotation, and translation should be investigated, as they can increase the diversity in the dataset and, therefore, help to increase the accuracy of the model. For this purpose, Bayesian optimization with KerasTuner [41] is used to identify the best hyperparameters. These techniques were selected to enable adaptability to variations in image presentation. Images may be scaled differently, or the image content may vary in position. These techniques simulate these scenarios and are intended to reduce the model's sensitivity to such variations. The upper limit of the search interval was established in this study in order to ensure that the resulting maximum value is not excessively high. The value was set at 0.2 (20%), which is not an exaggerated but still realistic value. To prepare for hyperparameter tuning, 10% of the training data is randomly selected as a validation dataset while preserving the ratio of classes to the original training dataset. The provided options for the Bayesian optimization can be found in Table 2.

For the optimization process, ten trials are conducted to identify the parameters that yield the lowest validation loss. Each of these ten trials is permitted to run for a maximum of 100 epochs. To avoid overfitting and to save computing time during the optimization process, *early stopping* was used, which terminates the training process if the validation loss has not decreased after 20 epochs. Once the optimal parameters are identified through this process, they are then used to train the final model. Once the optimal parameters have been determined through this process, they are used to train the final model. The final model is trained over 100 epochs with a batch size of 64 and a learning rate of 1×10^{-3}, using the Adam optimizer. These parameters were selected based on their prior utilization in other studies [11], [26], [42]. Additionally, early stopping is employed during the training process to mitigate the issue of overfitting. In this case, the training process is terminated after 20 consecutive epochs in which the validation loss has not decreased. The different effects of early stopping can be seen in Figure 6, where both models 100 epochs should be trained. The training of both models was terminated at different times, as the validation loss curve reached its lowest point at different epochs, indicating that the models had reached the point of optimal performance. An overview of the training parameters used, including early stopping, can be found in Table 3.

TABLE 2. Overview of the hyperparameters for data augmentation with associated search interval.

Hyperparameter	Minimum Value	Maximum Value	Step
Random Rotation	0.05	0.2	0.05
Random Zoom Hight	0.05	0.2	0.05
Random Translation Height	0.05	0.2	0.05
Random Translation Width	0.05	0.2	0.05

TABLE 3. Overview of parameters used for the training process.

Parameter	Value
Number of epochs	100
Batch size	64
Learning rate	1×10^{-3} (1×10^{-5} for fine-tuning the pre-trained architectures)
Solver	Adam
early stopping	20 consecutive epochs without decrease in validation loss

The training of the pre-trained architectures InceptionV3, Inception-ResNet-V2, and ResNet50 as comparative models to the architecture presented is similar. The big difference, however, is that transfer learning is used in this case. Before training, the oracle images are scaled to the required image size and dimension, as the pre-trained architectures cannot process an image size of 28×28 px and also require three dimensions. The input images have the following format: $32 \times 32\times3$ (ResNet50) and $75 \times 75\times3$ (InceptionV3, Inception-ResNet-V2). Each pre-trained architecture uses the original structure, but the final layer is customized to have ten neurons based on the ten classes. In the next step, the

training set was expanded using common data augmentation techniques such as zoom, rotation, and translation. The hyperparameters are then tuned using Bayesian optimization with the same settings as described above and the same parameter search intervals as in Table 2. The model is then trained with the best parameters at 100 epochs with a batch size of 64, Adam optimizer, and early stopping. As soon as the validation loss has not decreased for 20 epochs, the training is terminated. The model of the epoch with the best results is automatically restored. During the tuning of the hyperparameters and the Bayesian optimization, the base model is frozen. The final step is to apply transfer learning to the model. In this step, the base model is unfrozen so that it can be fine-tuned for improved performance. The learning rate of the optimizer is set to a minimum of 1×10^{-5}. The number of epochs for this phase is determined by adding 20 epochs to the best epoch of the previous training.

D. EVALUATION METRICS

The following performance indicators are used to evaluate and interpret the performance of the model: Accuracy, Balanced accuracy, True positive rate (TPR, sensitivity or recall), True negative rate (TNR or specificity), Positive predictive value (PPV or precision), Negative predictive value (NPV), Cohen's Kappa and at least F1-Score.

The Accuracy determines the overall effectiveness of a model [43]. To mitigate inflated performance estimates on imbalanced datasets, balanced accuracy can be utilized, defined as the arithmetic mean of sensitivity and specificity [44], [45]. The True positive rate metric indicates the accuracy of classifying the positive class, and maximizing it increases the likelihood of correctly identifying true members of the positive class [46]. The True negative rate, on the other hand, indicates how effectively a classifier identifies negative classes [43]. The Positive predictive value metric assesses prediction accuracy for the positive class by indicating the proportion of positive predictions that correctly match true positive instances [46]. The Negative Predictive Value, equivalent to the precision for the negative class, is the ratio of correctly classified negative samples to all samples classified as negative [47]. Cohen's kappa describes the reliability of a model by measuring the agreement between two judgments. It ranges from -1 to 1, where a Cohen's kappa of -1 indicates complete disagreement, and a Cohen's kappa of 1 signifies perfect agreement [48]. The harmonic mean of precision and recall, called F1-Score, ranges from 0 to 1, with the minimum (0) occurring when all positive samples are misclassified (true positives $= 0$) and the maximum (1) occurring for perfect classification (false negatives $=$ false positives $= 0$) [49].

E. DATASET

In this study, the Oracle-MNIST dataset [6] is used. This dataset consists of a total of 30,222 ancient characters, divided into ten classes. The images are in 28×28 px grayscale image format, 27,222 of them for training and 3,000 for testing; furthermore, all images are disjoint. The same data format as the original MNIST dataset The number of images in the classes is between 2,332 and 3,399. The imbalance ratio for the minority class to the majority class is $\approx$ 2:3 (0.6861). All images are pre-processed as follows: grayscaled, negated, resized, and extended to 28×28 px. The images have a gray or black background and vary in resolution. Additionally, as they were scanned from the original surfaces of the oracle bone, the images may be damaged to varying degrees and exhibit a high degree of noise. Figure 3 shows six example images of the character *wood* from the test set of Oracle-MNIST. The differences in relation to the noise, the non-uniformity of the characters, and the variety of damage are clearly visible. The oracle characters on the right are examples of increased noise and damage.

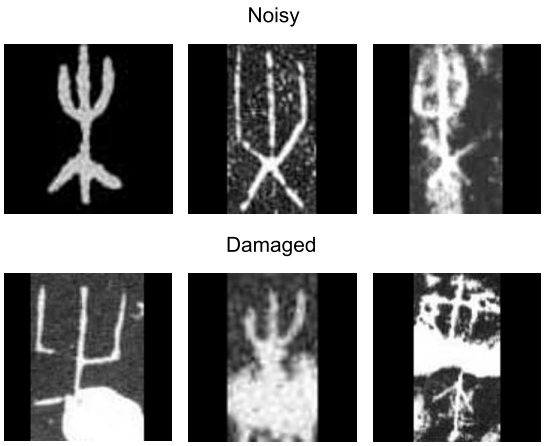


FIGURE 3. Six sample images of the *wood* class from the Oracle-MNIST test set, three each with varying degrees of noise and damage.

F. SETUP

For training and testing the models, Google Colaboratory with an NVIDIA Tesla T4 GPU with 16GB memory is used. The models were trained with a batch size of 64 over a number of 100 epochs. To avoid overfitting and to save computation time, the callback function *early stopping* was used, which stops the training after 20 consecutive epochs in which the validation loss has not decreased. To compare the performance of the proposed architecture, the pre-trained architectures Inception-ResNet-V2, InceptionV3, and ResNet50 were also trained on the dataset. Since these architectures do not accept grayscale images of size 28×28 px as input, the images were converted to the needed format.

IV. RESULTS

The architecture presented and the use of three pre-trained architectures as a comparison led to the results shown in Table 5. The architectures were evaluated in terms of accuracy, class-averaged sensitivity (true positive rate), positive predictive value (precision), and Cohen's Kappa. Additionally, the F1-score, balanced accuracy, true negative rate, and negative predictive value (NPV) are reported. It can

be seen that the proposed architecture performs better than the three compared architectures. For the accuracy, it achieves the highest value of 95.93%. This is 0.8% more than the second-highest accuracy of 95.13%, which was the previous benchmark of the dataset. Looking at Table 4, it can also be seen that the architecture presented uses the fewest parameters.

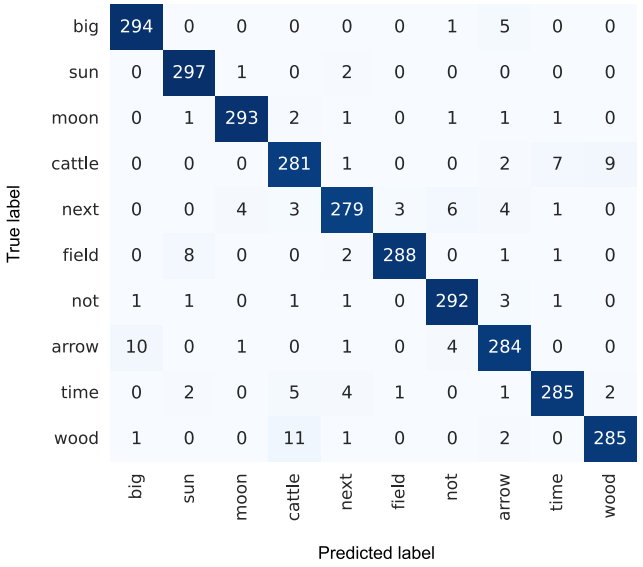


FIGURE 4. Confusion matrix of the proposed architecture with data augmentation.

A significance test was also conducted to demonstrate the robustness and reliability of the model, both with and without data augmentation. The model was trained and tested a total of 40 times for each specification (with and without data augmentation). For each specification, two datasets with 20 accuracy values each were created. Both datasets were normally distributed. To avoid side effects such as the sharing of weights from previous runs, each run was performed individually with a new runtime in Google Colaboratory. The two-tailed t-test was employed to investigate whether a significant difference existed between the mean values of the two datasets. The null hypothesis states that the mean values μ of the two datasets are equal ($H_0 : \mu_1 = \mu_2$). H_0 is rejected if the significance level α is less than 5%. In the case of the two datasets where data augmentation was applied, the resulting p-value was 99.00%. This leads to the conclusion that $p > \alpha$, indicating that H_0 cannot be rejected, and thus the datasets do not exhibit a statistically significant difference. In the second case of the two datasets where no data augmentation was applied, the resulting p-value was 90.03%. This result leads to the conclusion that $p > \alpha$, suggesting that H_0 cannot be rejected and, consequently, that the datasets do not show a statistically significant difference. The total of 40 runs per scenario produced an average accuracy of 95.98% with a standard deviation of $\pm$ 0.245% for use with data augmentation and 95.31% $\pm$ 0.281% without the use of data augmentation, which corresponds to the values in Table 5. In addition to the Tables 4 and 5, Figure 6 illustrates four images, with the left-hand side representing the results obtained without data augmentation and the right-hand side representing the results obtained with data augmentation. The top row depicts the training accuracy in comparison to the validation accuracy over the number of epochs, while the bottom row shows the loss with validation loss over the number of epochs. The graphs show that the preferred model with data augmentation is more robust and does not suffer from overfitting. Figure 4 shows the confusion matrix of the proposed architecture using data augmentation. As illustrated in the data, the class labeled *next* is the most frequently misclassified, with 21 misclassifications. The character *sun*, on the other hand, is best recognized with just three misclassifications. To evaluate the architecture's performance specifically on noisy and damaged images, 15 images exhibiting pronounced noise and damage were selected subjectively for each class. The corresponding confusion matrices are illustrated in Figure 5. The results indicate that the architecture demonstrates superior recognition accuracy for noisy images compared to damaged images. The balanced accuracy achieved was 92.67% for noisy images and 84.67% for damaged images.

V. DISCUSSION

The goal of this study was to achieve high accuracy with a suitable architecture for noisy and partially refracted images, as found in the Oracle-MNIST dataset. The results with various metrics, which are shown in Table 5, show that the presented architecture achieves the highest accuracy. For comparison, three state-of-the-art pre-trained architectures were selected and trained accordingly with transfer learning and data augmentation. The highest accuracy is achieved by the Inception-ResNet-V2 model, which is 3.93% lower than the proposed architecture. The results show that the use of ResNet is not particularly beneficial in the context of the Oracle-MNIST case, while InceptionV3 is more powerful. Inception-ResNet-V2, in contrast, employs both the residual connections of ResNet and the architectural design of the Inception module, which integrates disparate convolutional layers with varying kernel sizes in parallel. This integration results in enhanced accuracy and substantiates its efficacy. The presented architecture is also founded upon this architectural concept and attains an accuracy of 95.93% with supplementary data augmentation techniques. With regard to the number of parameters and the associated complexity of the model, the pre-trained architectures also achieve high values, which leads to a longer training time. The proposed architecture achieves lower values for the number of parameters (727,858).

The results of the significance tests conducted in section IV also demonstrated that the proposed architectural solution is capable of delivering robust and reliable results. It is important to note that this study does not examine the impact of novel, unseen oracle characters on the model's robustness, as the model would need to be specifically adapted for such

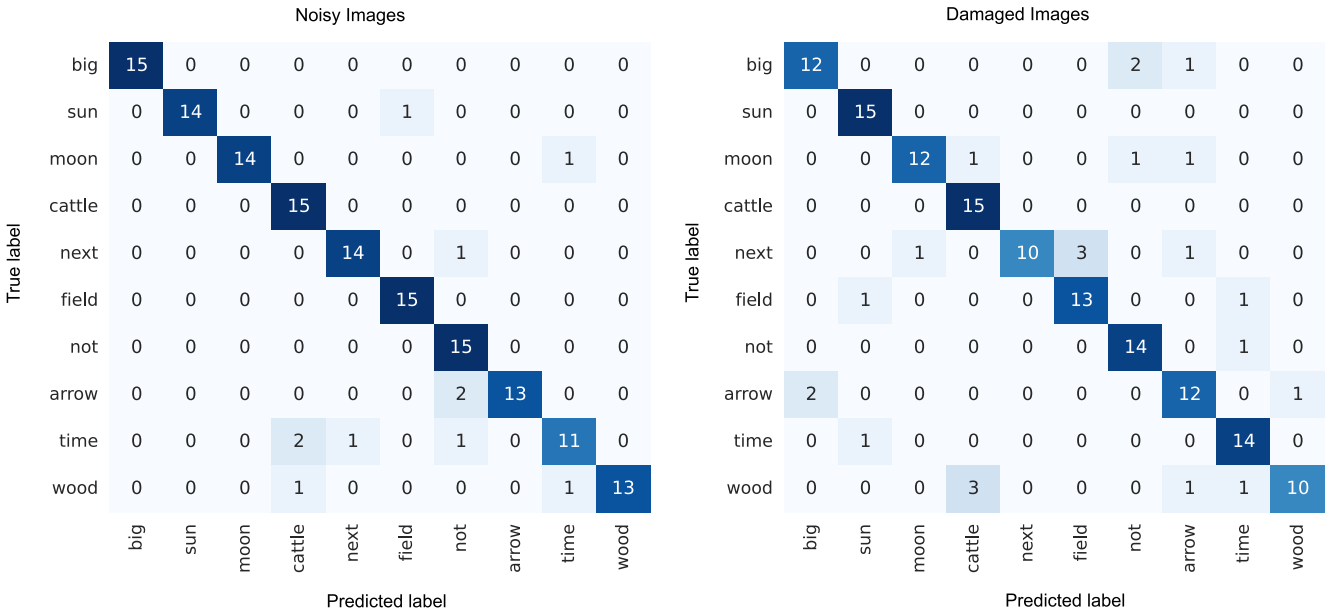


FIGURE 5. Two confusion matrices to illustrate the results for two different scenarios. The matrix on the left represents the case of images with increased noise, the matrix on the right the case of images with damage.

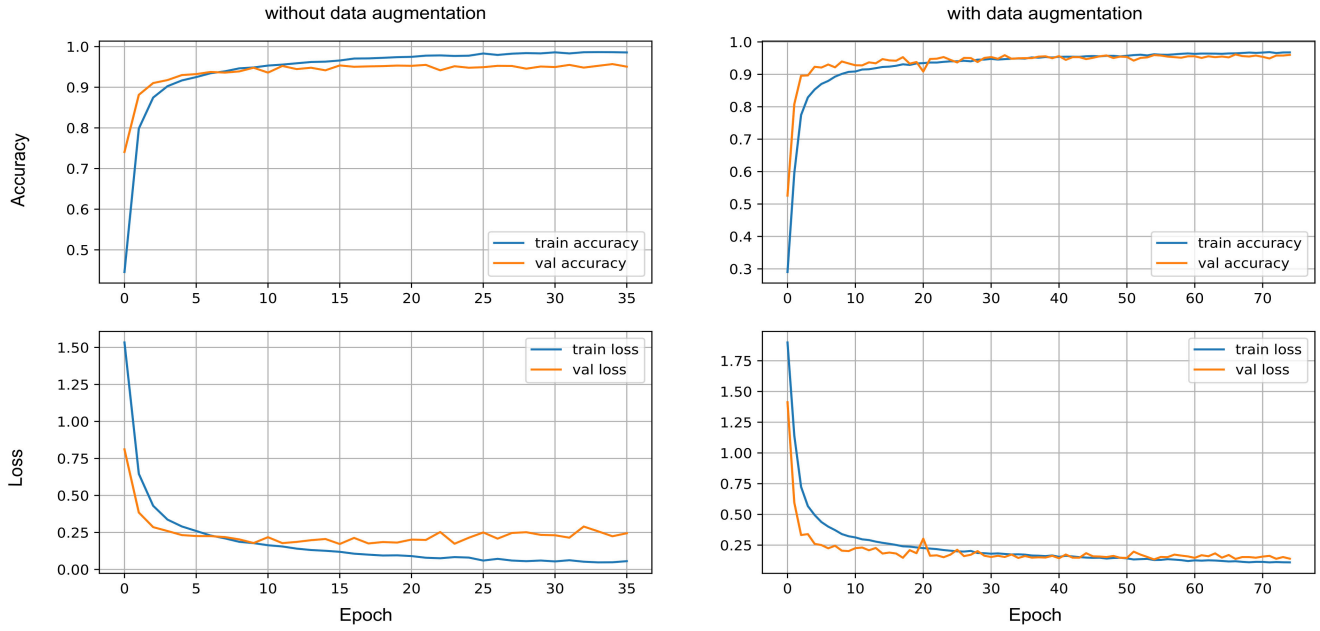


FIGURE 6. Accuracy and loss curves with associated validation for the case with and without use of data augmentation techniques. It can be seen that the training process was terminated at various epochs due to the application of the early stopping callback function.

TABLE 4. Overview of the number of parameters of four different architectures besides the proposed architecture using the Oracle-MNIST dataset.

Model	Trainable parameters	Non-trainable parameters	Total Parameters
InceptionV3	25,967,978	51,648	26,019,626
ResNet50	23,501,9362	106,240	23,608,202
Inception-ResNet-V2	54,261,290	90,816	54,352,106
LR-Net [26]	1,028,234	3,236	1,024,998
Proposed architecture	**724,974**	**2,884**	**727,858**

TABLE 5. Overview of the evaluation indicators of the proposed architecture and three different architectures on the Oracle-MNIST dataset.

Performance indicator	InceptionV3	ResNet50	Inception-ResNet-V2	proposed architecture | without data augmentation
Accuracy	91.53%	87.27%	92.00%	95.93% | 95.37%
Balanced accuracy	91.53%	87.27%	92.00%	95.48% | 95.37%
True positive rate	91.53%	87.27%	92.00%	95.93% | 95.37%
True negative rate	99.06%	98.59%	99.11%	99.55% | 99.49%
Positive predictive value	91.53%	87.27%	92.00%	95.93% | 95.37%
Negative predictive value	99.06%	98.59%	99.11%	99.55% | 99.49%
Kappa	0.9059	0.8585	0.9111	0.9548 | 0.9485
F1-Score	91.53%	87.27%	92.00%	95.93% | 95.37%

cases. Knowledge distillation, which has shown promising results in other fields [50], [51], may present a potential solution to this challenge, and we aim to investigate this approach in future research. In the work of Ganj et al. [26], the authors also employ a convolutional neural network based on the concept of residual connections and Inception modules. In contrast, we do not utilize a convolutional layer with a 7×7 kernel in our architecture, as we hypothesize that more global features play a less significant role than smaller kernels. In addition to the distinct configuration and quantity of convolutional layers, we employ batch normalization following each layer. In conclusion, our model demonstrates greater accuracy with fewer parameters, thereby corroborating its efficacy.

To examine the effects of noise and damage on the images, a test set comprising 15 images per class was selected for each case, noisy and damaged. These images subjectively and obviously showed an increased degree of noise or damage and approximately correspond to those depicted on the right-hand side of Figure 3. The corresponding confusion matrices are presented in Figure 5. A comparison of these matrices, in conjunction with the balanced accuracy referenced in section IV, indicates that the proposed architecture is more effective in correctly identifying images with noise than images with damage. A potential explanation for the lower accuracy in recognizing damaged images lies in the varying degrees of damage present. While some images are damaged, the general shape of the Oracle character can still be discerned. In other images, the damage is so extensive that a significant portion of the oracle character is missing, making it challenging for the architecture to accurately assign it to the correct character. Additionally, the structural similarity between certain characters introduces further complications to the classification process.

In relation to the confusion matrix (Figure 4), which contains the complete test set and not just a specific subtest set (Figure 5), the character *wood* is most frequently confused with the character *cattle* and vice versa, with *cattle* most frequently classified as *wood*. This is probably due to the fact that the two characters look quite similar. Similar cases can be observed with *arrow* and *big* as well as with *time* and *cattle*. Whereby *time* and *cattle* show greater differences with regard to the head of the character than other oracle bone characters. Figure 7 shows the five characters *cattle*, *wood*, *big*, *arrow*, and *time*; it can be seen that the pairs per row look similar. Further misclassifications can most likely be traced back to the condition of the OBIs. Some are very noisy or very badly damaged, with half or more of the symbol missing. In addition to the accuracy, the two metrics, NPV and TNR, stand out due to their significantly high values. The NPV evaluates the accuracy of the model with regard to the prediction of negative classes. Negative classes are classes that do not belong to the target class. A high value indicates the accuracy of the negative prediction. TNR, on the other hand, indicates the efficiency of a model in correctly identifying the negative classes. The opposite

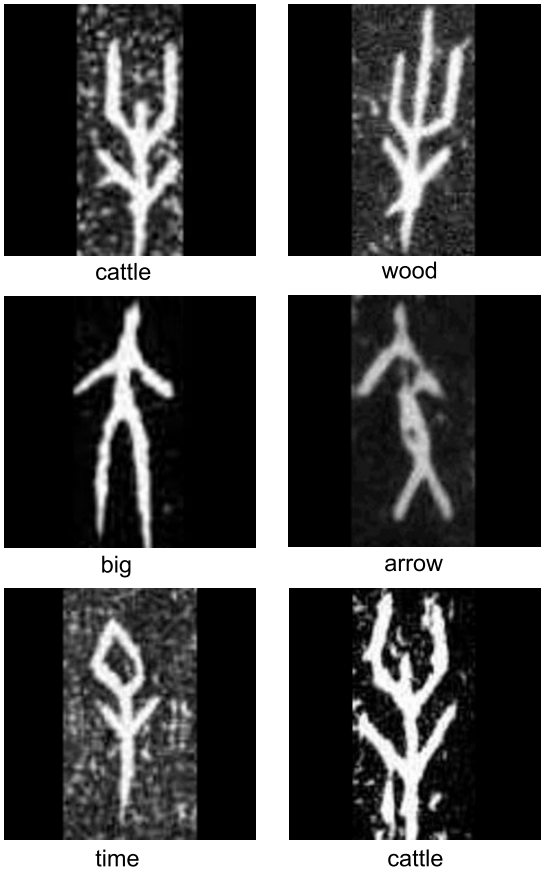


FIGURE 7. Comparison of three pairs of characters that have similarities (cattle and wood; big and arrow, time and cattle).

cases are PPV and TPR. Since both NPV and TNR have a value of 99.55%, they are 3.62% higher than PPV and TPR. This indicates that the model is more accurate in classifying actual negative cases as negative than vice versa. Furthermore, the negative predictions are highly reliable. Table 5 also shows the metrics relating to the model without the application of data augmentation. In comparison, the effectiveness of this method was shown, which resulted in an increase in accuracy of 0.56%. Upon examination of the average accuracy achieved in 40 independent runs, it was observed that the accuracy increased by 0.67%. This demonstrates the efficacy of data augmentation in enhancing model performance. To find the best parameters of the pre-processing layer, which included rotation, zoom, and translation, a Bayesian optimization process was employed. The optimization results indicated that smaller values for these parameters generally led to better accuracy, with the exception of the translation in the width direction.

This performance provides further evidence that while data augmentation is a valuable tool for improving model performance, the success of the proposed architecture is not contingent upon it alone. For instance, the proposed architecture was demonstrated to outperform the previous benchmark of 95.13% even in the absence of data augmentation, thereby indicating that the success of the architecture is not contingent upon these techniques alone. The application of data augmentation techniques facilitates the improvement of generalization and robustness by increasing the diversity of the training data. This is particularly beneficial in reducing the occurrence of overfitting, the presence of this can be seen in Figure 6. While the validation loss on the left side increases again from approximately the 20th epoch onwards, indicating the presence of overfitting, the validation loss on the right side does not exhibit a discernible decrease but rather a process of convergence. For this reason, the use of the callback function *early stopping* is of great importance. Furthermore, an examination of the validation accuracy curve indicates that it reaches nearly maximum levels after relatively few epochs. This observation suggests that the test dataset may not present a significant challenge for the model, in contrast to scenarios typically observed in studies such as [52]. Alternatively, it could indicate that the model is highly efficient at rapidly identifying patterns. The test dataset ensures that its images are not included in the training dataset; however, certain similarities to characters from the training data are not uncommon due to the limited size of the dataset.

VI. CONCLUSION

Deep learning has emerged as a modern and powerful tool for the recognition of OBIs, an area of increasing importance in character recognition. In this research, a novel architecture was introduced specifically for the Oracle-MNIST dataset, which contains numerous images of oracle characters. These images pose a significant challenge as they are highly noisy or damaged, making accurate classification difficult. Furthermore, data augmentation techniques were applied to increase the diversity in the dataset; this method proved to be beneficial. Using these techniques, the proposed architecture was able to outperform three comparable state-of-the-art models in terms of accuracy, setting a new benchmark for this dataset with 95.93%. To further validate the reliability of the model, 40 independent runs were performed, resulting in an average accuracy of 95.98% with a standard deviation of $\pm$ 0.245%. This performance underscores the robustness of the model. Furthermore, it was shown that the success of the architecture is not solely dependent on data augmentation techniques. Even without data augmentation, the model achieved an impressive accuracy of 95.37%, outperforming the previous benchmark. For further comparison, three pre-trained architectures, InceptionV3, ResNet50, and Inception-ResNet-V2, were trained using transfer learning. Currently, the most commonly used method for OBI recognition and deciphering relies heavily on human expert knowledge, which is both time-consuming and labor-intensive. The method presented in this study enables highly precise recognition of oracle character and offers considerable potential for saving personnel time and thus reducing costs.

A. LIMITATIONS

Although the classification model shows a satisfactory performance in the recognition of oracle bone inscriptions, it is important to recognize its limitations. For example, the achieved accuracy is based on ten classes, where the classes are relatively balanced in terms of the number of images (ratio between the smallest and the largest class $\approx$ 2:3). This relatively balanced distribution is not a standard case and may lead to suboptimal results when confronted with more unbalanced datasets. In addition, the performance of the model may vary when applied to datasets with substantially more classes, highlighting the need for further validation and possible adjustments. Future work should take these limitations into account.

B. FUTURE WORK

Future research will aim to assess the performance, external validity, and generalization of the classification model presented here by applying it to additional qualitative datasets containing scanned characters. In order to facilitate a comprehensive analysis of the model's scalability, datasets from varying historical periods will be incorporated in future studies. In addition, the architecture of the model will be further researched and refined to improve its effectiveness in dealing with noisy and damaged images. This includes reducing the number of parameters in the model, which not only decreases the computational requirements but can also increase the efficiency and speed of the model. Furthermore, the potential of knowledge distillation for oracle characters will be explored with a view to enhancing the model's robustness in the context of new oracle characters. In addition, the effects of generating synthetic images on the accuracy of the dataset are to be evaluated. This is achieved by including synthetic images in the dataset. Moreover, the existing

architecture is to be developed further in the form of a hybrid model, as proposed by Hax et al. [53], with the objective of achieving reliable recognition of oracle characters in videos.

REFERENCES

[1] M. Wang, W. Deng, and S. Su, "Oracle character recognition using unsupervised discriminative consistency network," *Pattern Recognit.*, vol. 148, Apr. 2024, Art. no. 110180.

[2] Y.-K. Zhang, H. Zhang, Y.-G. Liu, Q. Yang, and C.-L. Liu, "Oracle character recognition by nearest neighbor classification with deep metric learning," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 309–314.

[3] M. Wang, W. Deng, and C.-L. Liu, "Unsupervised structure-texture separation network for Oracle character recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 3137–3150, 2022.

[4] X. Yue, H. Li, Y. Fujikawa, and L. Meng, "Dynamic dataset augmentation for deep learning-based Oracle bone inscriptions recognition," *J. Comput. Cultural Heritage*, vol. 15, no. 4, pp. 1–20, Dec. 2022.

[5] Y. Tian, W. Gao, X. Liu, S. Chen, and B. Mo, "The research on rejoining of the Oracle bone rubbings based on curve matching," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 6, pp. 1–17, Nov. 2021.

[6] M. Wang and W. Deng, "A dataset of Oracle characters for benchmarking machine learning algorithms," *Sci. Data*, vol. 11, no. 1, p. 87, Jan. 2024.

[7] L. Meng, B. Lyu, Z. Zhang, C. V. Aravinda, N. Kamitoku, and K. Yamazaki, "Oracle bone inscription detector based on SSD," in *Proc. Int. Conf. Image Anal. Process.* Cham, Switzerland: Springer, Jan. 2019, pp. 126–136.

[8] Y. Fujikawa, H. Li, X. Yue, C. V. Aravinda, G. A. Prabhu, and L. Meng, "Recognition of Oracle bone inscriptions by using two deep learning models," *Int. J. Digit. Humanities*, vol. 5, nos. 2–3, pp. 65–79, Apr. 2022.

[9] W. Wang, T. Zhang, Y. Zhao, X. Jin, H. Mouchère, and X. Yu, "Improving Oracle bone characters recognition via a CycleGAN-based data augmentation method," in *Proc. Neural Inf. Process.*, Singapore. Springer, Jan. 2023, pp. 88–100.

[10] Y. Qiao and L. Xing, "Applying deep learning algorithms for automatic recognition and transcription of texts in Oracle bones and golden texts," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, pp. 1–16, Jan. 2024.

[11] M. Liu, G. Liu, Y. Liu, and Q. Jiao, "Oracle bone inscriptions recognition based on deep convolutional neural network," *J. Image Graph.*, vol. 8, no. 4, pp. 114–119, 2020.

[12] X. Fu, Z. Yang, Z. Zeng, Y. Zhang, and Q. Zhou, "Improvement of Oracle bone inscription recognition accuracy: A deep learning perspective," *ISPRS Int. J. Geo-Inf.*, vol. 11, no. 1, p. 45, Jan. 2022.

[13] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1, pp. 1–7.

[14] J. Guo, C. Wang, E. Roman-Rangel, H. Chao, and Y. Rui, "Building hierarchical representations for Oracle character and sketch recognition," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 104–118, Jan. 2016.

[15] S. Huang, H. Wang, Y. Liu, X. Shi, and L. Jin, "OBC306: A large-scale Oracle bone character recognition dataset," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 681–688.

[16] X. Lv, M. Li, K. Cai, X. Wang, and Y. Tang, "A graphic-based method for Chinese oracle-bone classification," *J. Beijing Inf. Sci. Technol. Univ.*, vol. 25, pp. 92–96, 2010.

[17] Q. Li, Y. Yang, and A. Wang, "Recognition of inscriptions on bones or tortoise shells based on graph isomorphism," *Comput. Eng. Appl.*, vol. 47, no. 8, pp. 112–114, 2011.

[18] Y. Liu and G. Liu, "Oracle bone inscription recognition based on SVM," *J. Anyang Normal Univ.*, vol. 2, pp. 54–56, 2017.

[19] L. Meng, "Recognition of Oracle bone inscriptions by extracting line features on image processing," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, 2017, pp. 606–611.

[20] L. Meng, "Two-stage recognition for Oracle bone inscriptions," in *Proc. Image Anal. Process. - ICIAP*, Jan. 2017, pp. 672–682.

[21] J. Li, Q.-F. Wang, K. Huang, X. Yang, R. Zhang, and J. Y. Goulermas, "Towards better long-tailed Oracle character recognition with adversarial data augmentation," *Pattern Recognit.*, vol. 140, Aug. 2023, Art. no. 109534.

[22] S. Zhuo and J. Zhang, "Attention-based deformable convolutional network for Chinese various dynasties character recognition," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121881.

[23] S. Zhuo, J. Zhang, and C. Zhang, "A novel data augmentation method for Chinese character spatial structure recognition by normalized deformable convolutional networks," *Neural Process. Lett.*, vol. 54, no. 6, pp. 5545–5563, Dec. 2022.

[24] Y. Zheng, Y. Chen, X. Wang, D. Qi, and Y. Yan, "Ancient Chinese character recognition with improved swin-transformer and flexible data enhancement strategies," *Sensors*, vol. 24, no. 7, p. 2182, Mar. 2024.

[25] Y. Zhou, G. Liu, Y. Yang, L. Ru, D. Liu, and X. Li, "UFCNet: Unsupervised network based on Fourier transform and convolutional attention for Oracle character recognition," in *Proc. 1st Workshop Mach. Learn. Ancient Lang. (ML4AL)*, 2024, pp. 98–106.

[26] A. Ganj, M. Ebadpour, M. Darvish, and H. Bahador, "LR-net: A block-based convolutional neural network for low-resolution image classification," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 47, no. 4, pp. 1561–1568, Dec. 2023.

[27] Z. Guo, Z. Zhou, B. Liu, L. Li, Q. Jiao, C. Huang, and J. Zhang, "An improved neural network model based on inception-v3 for Oracle bone inscription character recognition," *Sci. Program.*, vol. 2022, pp. 1–8, May 2022.

[28] L. Meng, N. Kamitoku, and K. Yamazaki, "Recognition of Oracle bone inscriptions using deep learning based on data augmentation," in *Proc. Metrol. Archaeol. Cultural Heritage (MetroArchaeo)*, Oct. 2018, pp. 33–38.

[29] G. Liu, "Oracle-bone inscription recognition based on deep convolutional neural network," *J. Comput.*, vol. 13, no. 12, pp. 1442–1450, Dec. 2018.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[31] J. Gao and X. Liang, "Distinguishing Oracle variants based on the isomorphism and symmetry invariances of oracle-bone inscriptions," *IEEE Access*, vol. 8, pp. 152258–152275, 2020.

[32] B. Li, Q. Dai, F. Gao, W. Zhu, Q. Li, and Y. Liu, "HWOBC–A handwriting Oracle bone character recognition database," *J. Phys., Conf. Ser.*, vol. 1651, no. 1, Nov. 2020, Art. no. 012050.

[33] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022.

[34] S. Obla, X. Gong, A. Aloufi, P. Hu, and D. Takabi, "Effective activation functions for homomorphic evaluation of deep neural networks," *IEEE Access*, vol. 8, pp. 153098–153112, 2020.

[35] C. Du, Y. Wang, C. Wang, C. Shi, and B. Xiao, "Selective feature connection mechanism: Concatenating multi-layer CNN features with a feature selector," *Pattern Recognit. Lett.*, vol. 129, pp. 108–114, Jan. 2020.

[36] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, Aug. 2018.

[37] S. Chen, H. Xu, G. Weize, L. Xuxin, and M. Bofeng, "A classification method of Oracle materials based on local convolutional neural network framework," *IEEE Comput. Graph. Appl.*, vol. 40, no. 3, pp. 32–44, May 2020.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[41] T. O'Malley et al. (2019). *Kerastuner*. [Online]. Available: https://github.com/keras-team/keras-tuner

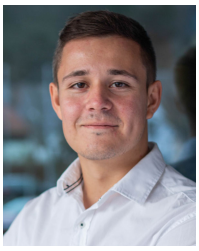
[42] Z. Wang, "VGG-16 framework with dropout for Oracle bone inscription recognition using transfer learning," *Highlights Sci., Eng. Technol.*, vol. 115, pp. 381–388, Oct. 2024.

[43] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[44] Q. Wei and R. L. Dunbrack, "The role of balanced training and testing data sets for binary classifiers in bioinformatics," *PLoS ONE*, vol. 8, no. 7, Jul. 2013, Art. no. e67863.

[45] D. Chicco, N. Tötsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Mining*, vol. 14, no. 1, pp. 1–22, Feb. 2021.
[46] C. Sharpe, T. Wiest, P. Wang, and C. C. Seepersad, "A comparative evaluation of supervised machine learning classification techniques for engineering design applications," *J. Mech. Design*, vol. 141, no. 12, Dec. 2019, Art. no. 121404.
[47] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Sci. Rep.*, vol. 12, no. 1, p. 5979, Apr. 2022.
[48] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.
[49] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, Dec. 2020.
[50] A. Amirkhani, A. Khosravian, M. Masih-Tehrani, and H. Kashiani, "Robust semantic segmentation with multi-teacher knowledge distillation," *IEEE Access*, vol. 9, pp. 119049–119066, 2021.
[51] A. Banitalebi-Dehkordi, A. Amirkhani, and A. Mohammadinasab, "EBCDet: Energy-based curriculum for robust domain adaptive object detection," *IEEE Access*, vol. 11, pp. 77810–77825, 2023.
[52] A. Amirkhani and A. H. Barshooi, "DeepCar 5.0: Vehicle make and model recognition under challenging conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 541–553, Jan. 2023.
[53] D. R. T. Hax, P. Penava, S. Krodel, L. Razova, and R. Buettner, "A novel hybrid deep learning architecture for dynamic hand gesture recognition," *IEEE Access*, vol. 12, pp. 28761–28774, 2024.

CHRISTOPHER MAI received the B.S. and M.S. degrees in mechanical engineering from the Brandenburg University of Technology Cottbus-Senftenberg, Germany, in 2019 and 2023, respectively. He is currently pursuing the Ph.D. degree with the Chair of Hybrid Intelligence, Helmut-Schmidt-University, Hamburg. His research interest includes machine learning and deep learning.



PASCAL PENAVA (Member, IEEE) received the B.S. degree in information systems from Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, in 2021, and the M.S. degree in digitalization and entrepreneurship from the University of Bayreuth, Bayreuth, Germany, in 2023. He is currently pursuing the Ph.D. with the Chair of Hybrid Intelligence, Helmut-Schmidt-University, Hamburg. His research interests include the development of EEG-based BCIs and machine learning-based analyses of time-series data.

RICARDO BUETTNER (Senior Member, IEEE) received the Dipl.-Inf. degree in computer science and the Dipl.-Wirtsch.-Ing. degree in industrial engineering and management from the Technical University of Ilmenau, Germany, the Dipl.-Kfm. degree in business administration from the University of Hagen, Germany, the Ph.D. degree in information systems from the University of Hohenheim, Germany, and the Habilitation (venia legendi) degree in information systems from the University of Trier, Germany. He is currently a Chaired Professor of Hybrid Intelligence at the Helmut-Schmidt-University, Germany. He has published over 150 peer-reviewed articles, including *Electronic Markets*, *AIS Transactions on Human–Computer Interaction*, *Personality and Individual Differences*, *European Journal of Psychological Assessment*, *PLOS ONE*, and IEEE Access. He has received 18 international best papers, the best reviewer, and the service awards and award nominations, including the Best Paper Awards by *AIS Transactions on Human–Computer Interaction*, *Electronic Markets* journal, and HICSS, for his work.