IVC (Indus Valley Civilization) MOTIF DETECTION USING CNN

Shirish Kumar Reddy Kaithi  
Artifical intelligence   
Long Island univeristy   
Brooklyn, NY [Shirishkumarreddy.kaithi@my.liu.edu](mailto:Shirishkumarreddy.kaithi@my.liu.edu)

Darshan Sanjaybhai Khunt  
Artifical intelligence   
Long Island univeristy   
Brooklyn, NY

Darshansanjaybhai.khunt[@my.liu.edu](mailto:Shirishkumarreddy.kaithi@my.liu.edu)

Sharis Stanley Rebeiro  
Artifical intelligence   
Long Island univeristy   
Brooklyn, NY  
Sharisstanley.rebeiro@my.liu.edu

*Abstract* — The main objective of the project is to use the given data about these seals and classify whether there is the unicorn symbol present in those seals or not. Some seals will have symbols, and some do not have any symbols. The dataset that we have for our project is the one which was provided to us on the canvas. This dataset has around 30 images of mixture of seals with unicorn symbol and no unicorn symbol. The main objective is to classify whether the unicorn seal is present or not. The model we used in this project is CNN is the abbreviation for Convolutional Neural Network, and it is one of the deep learning neural networks used in high-frequency image and video analysis. CNN is one of the types of deep learning models inspired by pattern recognition in images and videos by making a set of convolutional layers to extract the relevant features from the input data. These apply filters to the input data, with the purpose of highlighting important aspects of the image, such as edges or corners. Every convolutional layer is accompanied by a nonlinear activation after the output, typically ReLU for extracting important features from the data. That would be several layers of convolution and activation followed by flattening and one or more fully connected layers that perform the final classification or regression.

Keywords— CNN (conventional Neutral network), Indus Valley Civilization.

**I. Introduction**

# IVC stands for Indus Valley Civilization which is one of the bronze age civilizations which was started on the banks of river Indus. Today geographically this location is present in Pakistan. Around 3,000 B.C., a writing system, urban centers, social and economic systems belonged to one of the first great civilizations which is the Indus Valley Civilization. The remainders of two significant urban areas Mohenjo-daro and Harappa uncover astounding designing accomplishments of uniform metropolitan preparation and painstakingly executed format, water supply, and seepage.

More than 400 distinct symbols have been recorded by scholars on motifs and other surfaces. The writings are brief. There are 26 symbols in the longest. The typical number is about five. Based on the analysis of overlapping strokes, scholars believe that the symbols represent words, syllables, or sounds. They were read from right to left. Palm leaves, which could be eaten, were used for some of the earliest writing. Time has erased a lot of this. Most square seals have a prominent animal motif in the center, written text at the top, and an object often close to the animal's face. However, the animal has been drawn on inscribed objects. Unicorns are the most common animal motifs. The most acceptable depiction of unicorn is "a legendary creature with a solitary horn". Different hypotheses exist about the idea of this creature as well. One of these claims that the second horn is hidden behind the first and cannot be seen. It is interesting to note that similar figures with horns that are so symmetrical to one another in some of the structures of Persepolis in Persia look like unicorns from the side. In the literature, it has been argued that this is the edge view of a real double-horned animal; however, there are numerous clear depictions of two-horned animals with both horns visible.

Text

Description automatically generated

Figure 1.1Example of Indus valley seal

**II. Solution**

We are doing the classification of whether the seals are present or not using CNN. CNNs are very useful while working with image data. In some ways, neural networks resemble how the human brain works and learns day to day activities. Neurons are the essential and the building blocks of a neural network, and they in turn create layers. These layers represent the different ways of how a Neural Network learns the given problem and how the model is trained on different parameters. Tuned parameters are the alias for these neurons. There can be n number of layers depending on how the model is designed. The result of the previous layer is fed in as the input for the following layer. Each layer will have its own activation function which will help these layers in learning the process and get the output of that given layer. The term "terminal neurons" also refers to the output layer. In comparison to the CNNs require less preprocessing than other image classification techniques. This implies that the network learns to optimize the filters automatically, as opposed to the hand-engineered nature of older techniques. One important advantage of this feature extraction is its independence from human interaction and previous knowledge.

**III. How Does Convolutional Neural Network works**

This layer aims to give electronics the ability to see the world as humans do and to apply that understanding to a wide range of applications, including media recreation, natural language processing, image and video recognition, recommendation systems, and other related domains. The advancements of PC Vision with Profound Learning have developed over time to the point where they are essentially a Convolutional Neural Organization, which is more than just a single calculation. A Convolutional Neural Network, commonly referred to as CNN or ConvNet, is a potent computational tool that can recognize and differentiate between various elements in an image by applying learnable weights and biases to each one. ConvNet's development was primarily motivated by the operation of the Visual Cortex, which is akin to a network of neurons in the human brain. Within a specific part of the visual field called the Open Field, only certain individual neurons respond to changes. Multiple such fields overlap to encompass the entire visual area, and the Visual Cortex consists of three distinct layers.

* **Convolution Layer**: The Convolution Action intends to remove critical level components, like edges of high objects, to keep them from affecting the data picture. A ConvNet need(s) not be confined to a single Convolutional Layer. The first ConvLayer must deal with the low-level features like gradient orientation, edges, and color. As the layers increase, the architecture adapts itself to the high-level features such that the network learns to understand all images of the dataset as clearly as possible, much like humans would. The operations lead to two approaches; one diminishes the dimensionality of the convolved feature versus the input, or enlarges or, otherwise, retains it. This is accomplished using large padding in the first case and same padding in the second.
* **Pooling Layer:** In ConvNets, the Pooling Layer oversees shrinking the Convolved Feature's spatial size. By doing this, the dimensionality of the data is decreased, which lowers the processing power needed. Additionally, by eliminating prominent traits that are unaffected by position or rotation, it preserves the model's efficacy. Max Pooling and Average Pooling are the two types of pooling. The greatest value from the image's covered area is returned by Max Pooling, whereas the average of all the values in the covered area is returned by Average Pooling. Furthermore, Max Pooling decreases background noise more effectively than Average Pooling because it eliminates noisy activations and lowers dimensionality, while Average Pooling merely lowers dimensionality by suppressing noise. Thus, it may be said that Max Pooling performs better than pooling it average.

Diagram, engineering drawing

Description automatically generated

Figure: 3.1 CNN Pooling Layer

**Fully Connected Layer:** Non-linear combinations of the high-level features represented by the convolutional layer's output can be inexpensively accomplished with a Fully Connected layer. The model learns a non-linear function in that feature space at this layer. We first convert the input image and then flatten it into a column vector in order to get it ready for our Multi-Level Perceptron. A feed-forward neural network is then fed the flattened output, with each training iteration utilizing back propagation. Images are categorized using the SoftMax Classification approach, which relies on the model's capacity to distinguish between dominant and specific low-level features over time.

**Abbreviations and Acronyms**

IVC: Indus Valley Civilization

ReLU (Rectified Linear Unit),

CNN- Convolutional Neural Network

ConvNet- Convolutional Neural Network

**IV. Approach**

This research will employ a methodology that combines deep learning and computer vision techniques. To make sure the photos are appropriate for examination, they will first undergo pre-processing. This could entail scaling, cropping, and normalizing the pictures. The pre-processed photos will then be used to train the CNN (VGG 16, Resnet, etc.) model using the training dataset. The model will have the ability to identify if motifs are present in the pictures or not. Lastly, the performance of the trained model in identifying motifs on IVC seals will be assessed using the testing dataset.

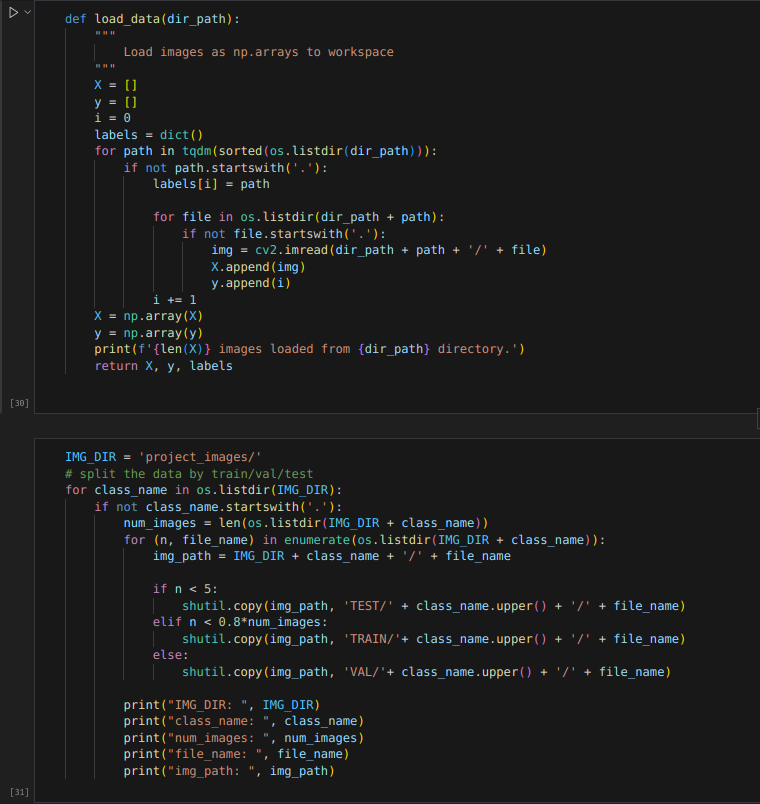
A diagram of a model

Description automatically generated

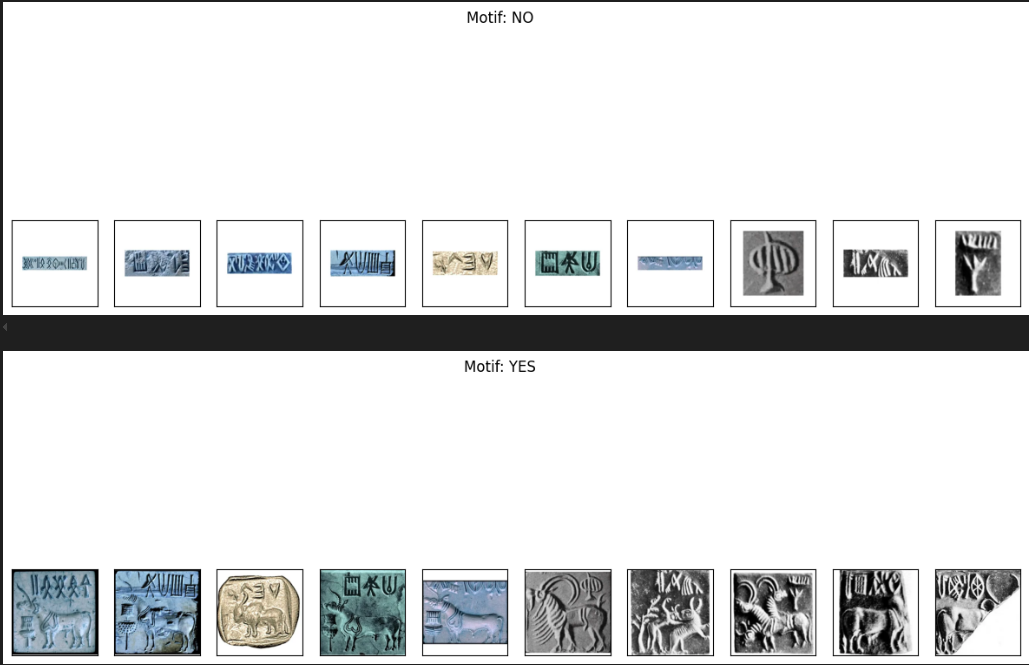
4.1: Approach Method

**V. Code and understanding**

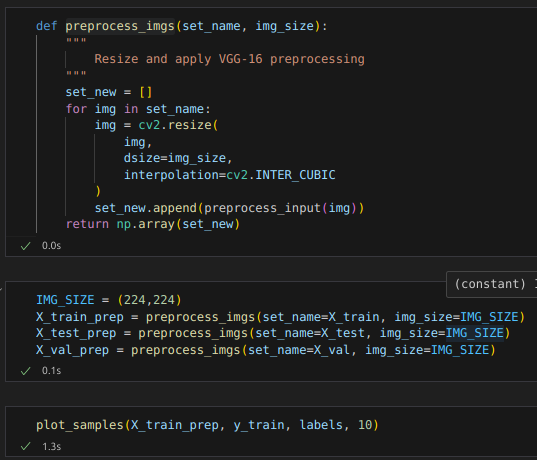
Our code starts off with getting a dataset. The dataset consisted of 52 images of IVCs in which 30 of them had the unicorn seal motif and the rest did not have them. We read these seals as input and preprocessed the images. After reading these images as the input, we had to classify them into training data, testing data and validation data. This was done in a 75-15-10%.



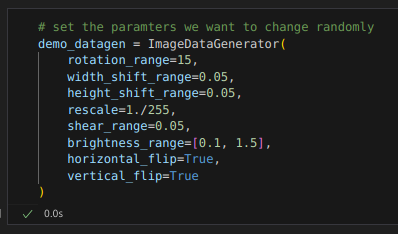
The above image represents how we have read the data into our directory and how we are diving the data into train test and validation. We are storing train data in “Train/” directory, test data into the “Test/” directory and validation data into the “VAL/” directory. The image size which we are considering in our project is 224x224.



The above image represents how we are classifying whether the motif is present or not. And the above images just represent that we are showing that few images in which motif is present and the few images in which the motif is not present. The next image shows how we are preprocessing the images and how we are representing them to the model.



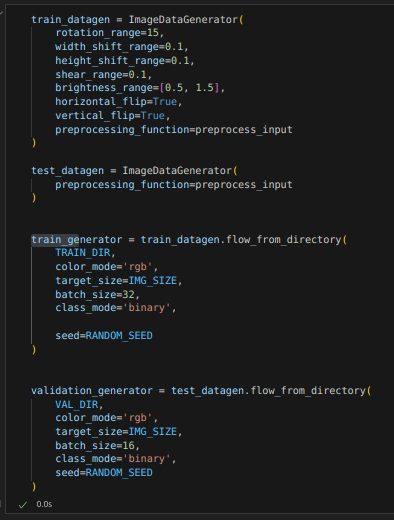
It is always required to perform data augmentation on any data so that we get more information on the given data, and which can be used by the model to understand different features of the image. The following image represents the conditions under which we are performing the data augmentation. We are performing augmentation on rotation, brightness etc., and other factors which are required to obtain more and more information on the given image.



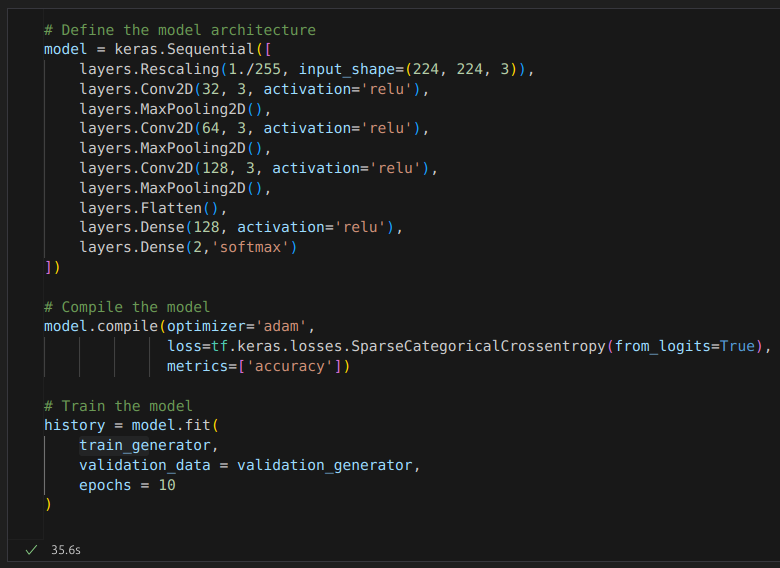
This is one of the examples in which we have the original image, and we are augmenting this image and obtaining the augmented images, which we are using in the model for training and not for testing and validation. Testing and validation in done the real images.

**VI. Model**

The following image represents the different train-datagen and the test-datagen. This is required to train the model. These are basically generators which uses the standard built-in ImageDataGenerator to check the images. The train\_generator and the validation\_generator are the generators which are used to load the data into the model in the form of batches. The most common used batch size is 32.

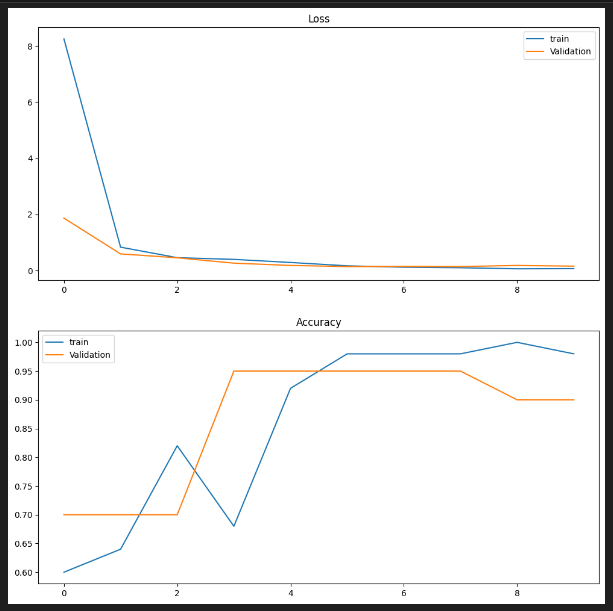


The following represents our model architecture which we are using. There are different layers such as Maxpooling, Conv2D, Flatten() , Dense(). All these represent the different layers, and the output from one layer is fed as the input to another layer.

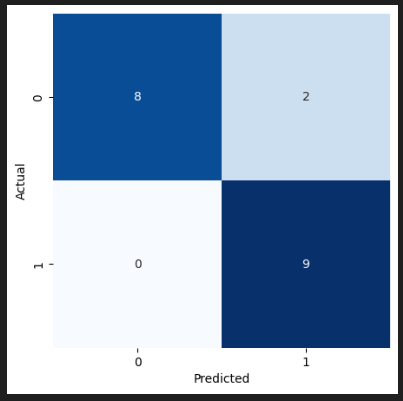


This is used to compile the model, The optimizer used is the ‘adam’ optimizer. This is the industry standard optimizer and the best out there currently in the market. This is an algorithm which is used for stochastic optimization that updates the model weights based on the gradient of the loss function. The loss function is a multi-class classification problems.

**VII. Result**



This is the graph which we have obtained. We are plotting two graphs. The first graph represents the loss, and the second graph represents the accuracy. We can see the blue line in the graph which represents the train data and the orange line which represents the validation data.



This is the confusion Matrix which identifies and classifies the 10 images. Through this we can understand how many images are predicted and how many images are actual and predicted.

**VIII. Conclusion**

We have understood Convolutional Neural Networks and how to use the same to train the model. We also tested and made predictions with the given data for us. We also learn how to choose activation function based on the necessity and number layers we need for model for each different scenario. We also learnt a lot of standard industrial solutions which are currently in use in the market. This project gives us the best way to start with Artificial Intelligence, developed interest to implement other scenarios and projects. This also helped us understand a lot of minute details which are required to understand a model.

**IX. Future Improvements**

The training of this project with more distinct photos will be a future development. As previously said, picture modification only functions when there are sufficient different pictures before the change. Since there aren't many distinct IVC artifact images online, a new method must be developed to obtain them. Getting clean data is very hard, this might take a lot of resources and time for us to figure out which data is good and bad.

We can also distinguish the images into 3 types, one which is classifiable by the model, 2 the broken seals which are tough to classify and 3rd the text motifs which can be classified manually, The model can just detect the text and prompt that this image is better off when classified manually than the model. We can also detect the text and implement some text detection which can recognize the motifs easily.

Another major improvement which can be made is to classify more types of seals. In our project we have classified only unicorn motifs, there are many other seals based on which we can classify and predict our model.

**X. Authors and Affiliations**

Shinde, V., & Petrakis, E. (2017). Deep learning for archaeology: automatic analysis of Indus script. Journal of Archaeological Science: Reports, 12, 722-728.

Li, Q., Li, Y., Li, X., & Wang, Y. (2020). Multi scale deep learning for artifact classification in archaeology. Journal of Archaeological Science: Reports, 29, 102499.

Choudhary, M. (2017). A deep learning-based approach for classification of Indus Valley civilization seals. Proceedings of the International Conference on Emerging Trends in Computing and Information Sciences (ICETCIS), pp. 703-709.

##### **XI. References**

[1] https://levity.ai/blog/neural-networks-cnn-ann-rnn

[2] Motif & Textbox Recognition on IVC artifacts - Qi Mo Michael Tishman

[3] Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016. 2, 4, 5, 6, 8

[5] https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5- way/#:~:text=A%20Convolutional%20Neural%20Network%20(ConvNet,differentiate%20one%20from%20the%20other.https://pandas.pydata.org/

[6] <https://numpy.org/>

[7] https://matplotlib.org/stable/index.htmlcited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

[8] http://www.sutrajournal.com/asko-parpola-on-the-roots-of-hinduism-by-vikram-zutshi