5-2024

Deep Learning in Indus Valley Script Digitization

Deva Munikanta Reddy Atturu
Florida Institute of Technology, datturu2023@my.fit.edu

DEEP LEARNING IN INDUS VALLEY SCRIPT DIGITIZATION

by

DEVA MUNIKANTA REDDY ATTURU

Bachelor of Technology
Computer Science and Engineering
Siddharth Institute of Engineering and Technology
2021

A Thesis
submitted to the College of Engieering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Melbourne, Florida
May, 2024

We the undersigned committee
hereby approve the attached Thesis

DEEP LEARNING IN INDUS VALLEY SCRIPT DIGITIZATION by DEVA

MUNIKANTA REDDY ATTURU

Debasis Mitra, Ph.D.
Professor
Electrical Engineering and Computer Science
Major Advisor

Xianqi Li, Ph.D.
Assistant Professor
Mathematics and Systems Engineering

Eraldo Ribeiro, Ph.D.
Associate Professor
Electrical Engineering and Computer Science

Brian Lail, Ph.D.
Professor and Department Head
Electrical Engineering and Computer Science

Abstract

Title:

DEEP LEARNING IN INDUS VALLEY SCRIPT DIGITIZATION

Author:

DEVA MUNIKANTA REDDY ATTURU

Major Advisor:

Debasis Mitra, Ph.D.

This research introduces ASR-net(Ancient Script Recognition), a groundbreaking system that automatically digitizes ancient Indus seals by converting them into coded text, similar to Optical Character Recognition for modern languages. ASR-net, with an 95% success rate in identifying individual symbols, aims to address the crucial need for automated techniques in deciphering the enigmatic Indus script. Initially Yolov3 is utilized to create the bounding boxes around each graphemes present in the Indus Valley Seal.In addition to that we created M-net(Mahadevan) model to encode the graphemes.

Beyond digitization, the paper proposes a new research challenge called the Motif Identification Problem (MIP) related to recurring patterns (motifs) on Indus seals that appear to have specific functions within certain periods of the civilization. Despite challenges in applying deep learning to MIP, The database was created to store the ImageID, Image, the list of encoded graphemes present in that particular image fol-

lowed by the Motif on the IVC Seal in the structured format.

Table of Contents

List of Figures

List of Symbols, Nomenclature or Abbreviations

$M-net$	$Mahadevan-net$
$MIP-net$	$MotifIdentificationProblem-net$
$ASR-net$	$AncientScriptRecognition-net$

Acknowledgements

I owe a profound debt of gratitude to my thesis advisor, Dr. Debasis Mitra, for his unwavering support, invaluable guidance, and constant motivation throughout this research endeavor. His depth of knowledge and insightful critiques have been instrumental in shaping every aspect of this thesis. I extend my heartfelt thanks to my esteemed committee members, Dr. Xianqi Li and Dr. Eraldo Ribeiro, for graciously accepting to be part of my committee and for their invaluable contributions and support. Additionally, I am deeply appreciative of the staff and faculty at the Florida Institute of Technology for fostering a stimulating academic environment conducive to intellectual growth. Lastly, I extend my gratitude to all the participants whose invaluable contributions made this research possible. Special thanks also go to Shubam B, Ali N, Ujjwal B, and Mukharjee A for their significant contributions.
National Endowment for the Humanities : PR-290075-23.

Dedication

To my beloved family, whose unwavering love, encouragement, and sacrifices have been my anchor throughout this academic journey. Your steadfast support has fueled my determination to reach this milestone. This thesis is dedicated to you, with heartfelt gratitude and immense love. To my esteemed thesis advisor, Debasis Mitra, whose constant support, guidance, and motivation have been indispensable. Your expertise and insightful critiques have profoundly shaped this thesis. I also extend my sincere appreciation to my committee members, Dr. Xianqi Li and Dr. Eraldo, for their invaluable feedback and support. To the staff and faculty at Florida Institute of Technology, thank you for fostering a stimulating academic environment that has nurtured my growth and learning. And to all the participants who made this research possible, your contributions are deeply appreciated. This thesis is a reflection of the collective efforts and support that have propelled me forward. Thank you all.".

Chapter 1

Introduction

The Indus Civilization, also known as the Harappan Civilization, represents one of the world's oldest urban societies, flourishing in the vast floodplains of the Indus River and possibly the now-extinct Saraswati River in present-day Pakistan and northwest India. Spanning roughly from 2600 BCE to 1900 BCE, this ancient civilization is renowned for its advanced urban planning, sophisticated drainage systems, standardized weights and measures, and distinctive artifacts, including seals bearing inscriptions in the enigmatic Indus script. Despite its prominence, the Indus script remains undeciphered, posing a significant challenge to scholars seeking to unravel the mysteries of this ancient civilization. Unlike other ancient civilizations such as Egypt and Mesopotamia, which have benefited from the discovery of bilingual inscriptions like the Rosetta Stone, the Indus Civilization lacks a comparable linguistic key, hindering efforts to decipher its script and understand its society, economy, and culture.

Over the past century, scholars have engaged in meticulous studies of the Indus script, employing various methodologies to decipher its meaning. However, the absence of a Rosetta Stone equivalent has compelled researchers to explore alternative approaches, such as statistical analyses of grapheme sequences, intra-script grapheme associations, and contextual clues derived from archaeological artifacts. These manual efforts, while insightful, are labor-intensive, time-consuming, and limited in scalability.

In recent years, advancements in data science and machine learning have opened up new avenues for the computational analysis of ancient scripts, offering the potential to automate and expedite the decipherment process. To address the challenge of grapheme identification within the Indus script, we propose the use of ASR-net, a novel neural network architecture that combines the strengths of M-net and YOLOv3 for efficient and accurate identification of individual graphemes. ASR-net leverages the capabilities of M-net for character recognition and YOLOv3 for object detection, enabling robust detection and classification of graphemes on Indus seals.

Moreover, motif identification on Indus seals presents another significant challenge, as these motifs often serve as key elements for understanding the symbolic and cultural significance of the artifacts. To tackle this challenge, we introduce MIP-net, a machine learning framework specifically designed for motif identification in archaeological imagery. MIP-net employs convolutional neural networks (CNNs) trained on annotated datasets of Indus seals to automatically identify and classify motifs, allowing for efficient analysis of large collections of artifacts.

In light of these developments, our research aims to bridge the gap between traditional scholarship and computational analysis by proposing a machine learning-based approach for the automated identification and analysis of motifs—distinctive symbols or iconographic elements—found on Indus seals. These seals, typically made of steatite or other soft stones, feature intricate engravings comprising motifs, often accompanied by short inscriptions in the Indus script. By leveraging ASR-net for grapheme identification and MIP-net for motif identification, our proposed system seeks to automate the process of deciphering Indus seals, enabling researchers to efficiently analyze large collections of artifacts and extract valuable insights into the socio-cultural and economic aspects of the Indus Civilization.

Additionally, we have developed a comprehensive database comprising high-resolution images of Indus seals, along with metadata detailing their provenance, dimensions, and associated inscriptions where available. This database serves as a foundational resource for our research, providing a rich repository of visual and contextual data for training and validating our machine learning models. Through the development of automated tools for motif identification, we aim to contribute to the broader scholarly efforts aimed at deciphering the Indus script and shedding light on the rich tapestry of the ancient Indus Civilization. By harnessing the power of machine learning and computational analysis, we hope to unlock new avenues of research and deepen our understanding of this enigmatic ancient society.

Below is the brief description of what the chapter describes about.

Chapter 2 presents a comprehensive survey of existing literature on the decipherment of the Indus script. Traditional methodologies and computational approaches used in Indus script analysis are reviewed, critically evaluating previous efforts and identifying gaps in research.

Chapter 3 introduces key concepts and methodologies employed in the research. It explains machine learning algorithms and techniques relevant to motif identification, along with an overview of data annotation, model training, and evaluation processes.

Chapter 4 describes the proposed methodology for automated motif identification on Indus seals. It discusses the rationale behind the selection of machine learning algorithms and data preprocessing techniques, providing an outline of the workflow for data annotation, model training, and deployment.

Chapter 5 provides a detailed explanation of the implementation process, including data collection, annotation, and model training. It describes the tools and technologies utilized in the implementation phase, along with an overview of the challenges encountered and solutions devised during implementation and includes Unified Modeling Language (UML) diagrams illustrating the system architecture, data flow, and entity relationships. It explains each diagram and its relevance to the proposed approach and implementation..

Chapter 6 presents and analyzes the results obtained from the implementation phase. It evaluates the performance of the machine learning models in motif identification and discusses the implications of the results for deciphering the Indus script and understanding the Indus Civilization.

Chapter 7 discusses the challenges encountered during the research process. It explores the difficulties faced in implementing the proposed approach, including technical limitations, data quality issues, and methodological constraints.

Chapter 8 provides a summary of the research findings and their significance in the context of deciphering the Indus script. It reflects on the strengths and limitations of the proposed approach and proposes future research directions and potential improvements to the methodology.

Next we have the bibliography, listing all the references cited throughout the thesis or research paper. It provides readers with a comprehensive list of sources for further reading and verification of the information presented in the document.

The following part will elaborate on the background work associated with the project.

Chapter 2

Literature Survey

The study by Varun Venkatesh et al. [31] investigated the Indus script by analyzing patterns and positions of individual signs, pairs, and sequences. They built statistical models and algorithms to predict sign behavior based on their position. This analysis revealed significant differences in the language used in Indus texts from West Asia compared to those from the Indian subcontinent, suggesting distinct regional dialects within the Indus civilization.

Researchers have proposed a novel method to tackle the challenges of deciphering undeciphered scripts like the Indus Valley Script in a study by Shurthi Daggumati et al. [3]. This method focuses on identifying and grouping together different ways of writing the same symbol (allographs) based on their positions within the inscriptions. The authors argue that this approach can significantly simplify the script by reducing the number of unique symbols, potentially paving the way for a breakthrough in deciphering its hidden messages. They applied their method to the Indus Valley Script and identified 50 symbol pairs that could be grouped, reducing the complexity of the script by 12%. This exciting development holds promise for unlocking the secrets of these ancient languages.

In a paper by Michael Oakes et al. [17], the distribution of Indus Valley script signs found in Mahadevan's 1977 concordance is analyzed. Using Large Numbers of Rare Events (LNRE) models, the authors estimate a vocabulary of around 857 signs, including undiscovered ones. Statistical analysis reveals non-random distributions based on factors like position, archaeological site, object type, and direction of writing. The authors conclude that further analysis is needed to understand the underlying structure and meaning of the Indus Valley script.

While the study by Ansumali Mukhopadhyay et al. [16] offers an intriguing approach to deciphering the Indus Valley script using Dravidian languages, it acknowledges several key areas requiring further exploration. The connection between Dravidian languages and the Rig Veda remains a point of debate within academic circles, and the vast timeframe between the Mehargarh civilization and the Indus Valley necessitates careful consideration. Additionally, the paper highlights the uncertainties surrounding the Aryan invasion and its impact on pottery styles. By acknowledging these open questions and encouraging further research, the analysis ultimately contributes to the ongoing quest to unlock the secrets of the Indus script, even if it doesn't provide definitive answers at this stage.

In their study, S.Palaniappan et al. [18] recognize the endeavor to automate the preparation of standardized corpora for undeciphered scripts as a significant challenge, often requiring laborious manual effort from raw archaeological records. Recent efforts have sought to address this challenge by exploring the potential of machine learning algorithms to streamline the process, offering valuable insights for epigraphical research. Building upon this groundwork, authors present a pioneering deep learning pipeline tailored for the Indus script, aiming to automate the extraction and classification of graphemes from archaeological artifacts. Through the integration of convolutional neural networks and established image processing techniques, their methodology demonstrates promising advancements in accurately identifying and categorizing textual elements. This work contributes to the evolving landscape of computational epigraphy, showcasing the potential of deep learning approaches to revolutionize research methodologies in the digital humanities domain.

The related works presented by the cited papers offer valuable insights and methodologies relevant to the project of deep learning in Indus Valley script digitization. Firstly, they highlight the complexity of the script and the challenges associated with deciphering it, emphasizing the need for innovative approaches. The studies on statistical analysis and allograph identification provide crucial groundwork for understanding the patterns and structures within the script, which can inform the design of deep learning models. Additionally, the exploration of linguistic connections, such as with Dravidian languages, offers potential insights into the script's origins and linguistic context. Moreover, the efforts to automate corpus preparation and grapheme extraction demonstrate the application of advanced computational techniques, particularly deep learning, in streamlining the digitization process. By building upon these previous works, the project aims to leverage deep learning algorithms to automate the analysis and interpretation of the Indus Valley script, ultimately contributing to the broader goal of unlocking its hidden messages and historical significance.

The subsequent section will detail the array of concepts utilized in the project.

Chapter 3

Conceptual Landscape

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have revolutionized the field of computer vision by introducing powerful hierarchical representations of visual data. Unlike traditional neural networks, CNNs are specifically designed to effectively capture spatial hierarchies in images through the use of convolutional layers. These layers consist of filters that slide over input images, capturing local patterns and features at different spatial scales. By stacking multiple convolutional layers followed by pooling layers, CNNs are able to progressively learn complex representations of visual data.

The architecture of a typical CNN comprises multiple layers, including convolutional layers, activation functions, pooling layers, and fully connected layers. Convolutional layers are responsible for learning features from input images by applying convolution operations with learnable filters. Activation functions, such as ReLU (Rectified Linear Unit), introduce non-linearity to the network, allowing it to learn complex relationships between features. Pooling layers, such as max pooling or average pooling, downsample feature maps to reduce the spatial dimensions and computational complexity of subse-

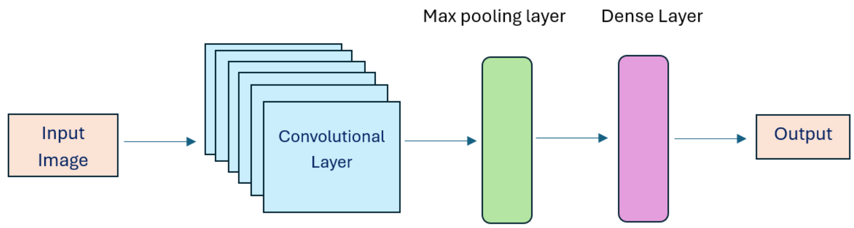

Figure 3.1: Basic CNN Architecture

quent layers. Fully connected layers integrate extracted features for final classification or regression tasks.

CNNs have demonstrated remarkable success in various computer vision tasks, including image classification, object detection, and semantic segmentation. Their ability to automatically learn hierarchical representations of visual data has led to significant advancements in fields such as medical imaging, autonomous driving, and image-based biometrics. Additionally, CNNs have been widely adopted in industry applications, powering image recognition systems in smartphones, surveillance cameras, and quality control systems.

The widespread adoption of CNNs can be attributed to their effectiveness in handling large-scale visual data, robustness to variations in input, and scalability to different tasks and domains. Their architecture and design principles have laid the foundation for numerous advancements in deep learning and computer vision research. As CNNs continue to evolve with innovations such as residual connections, attention mechanisms, and efficient architectures like MobileNet, they remain at the forefront of cutting-edge research and practical applications in the field of computer vision.

3.2 YoloV3

YOLOv3, short for You Only Look Once version 3, is an advanced object detection model renowned for its efficiency and accuracy. Introduced by Joseph Redmon and Ali Farhadi in 2018, YOLOv3 represents a significant improvement over its predecessors by incorporating several key enhancements. The fundamental concept behind YOLOv3 is its ability to perform object detection in real-time by dividing the input image into a grid and predicting bounding boxes and class probabilities directly from the grid cells.

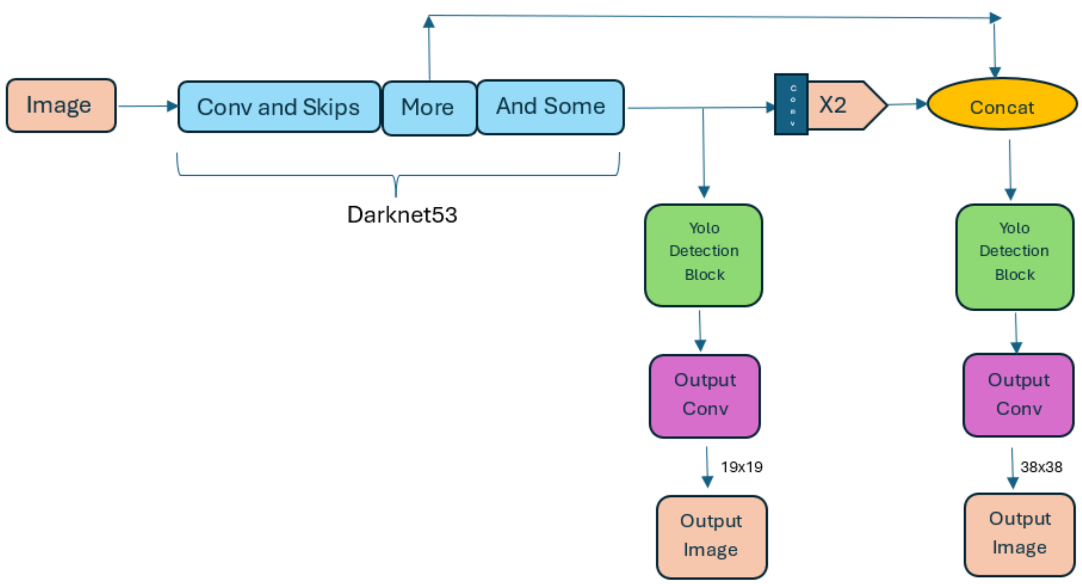

Figure 3.2: YoloV3 Architecture

The architecture of YOLOv3 is built upon a deep convolutional neural network backbone, typically based on Darknet, a custom CNN architecture designed for YOLO models. YOLOv3 consists of multiple convolutional layers followed by detection layers responsible for predicting bounding boxes and class probabilities. Notably, YOLOv3 utilizes a feature pyramid network (FPN) to extract multi-scale features from different layers of the network, enabling accurate detection of objects at various scales and resolutions.

One of the key features of YOLOv3 is its ability to predict bounding boxes at different scales using a technique called multi-scale prediction. This allows YOLOv3 to detect objects of varying sizes and aspect ratios with high accuracy. Additionally, YOLOv3 incorporates anchor boxes to improve the localization of objects by predicting bounding box offsets relative to predefined anchor shapes.

YOLOv3 has gained widespread popularity due to its impressive performance in real-time object detection tasks across diverse domains, including surveillance, autonomous driving, and robotics. Its efficiency in processing images and videos in real-time makes it a popular choice for applications requiring rapid and accurate object detection capabilities.

We integrate YOLOv3 into our project to create bounding boxes around the graphemes present on Indus seals. This enables us to accurately identify and isolate the individual graphemes for further analysis. YOLOv3's efficiency in processing images and videos in real-time makes it a popular choice for applications requiring rapid and accurate object detection capabilities.

3.3 MobileNet

MobileNet is a groundbreaking convolutional neural network (CNN) architecture specifically designed to address the computational constraints of mobile and embedded devices while maintaining high accuracy in image classification tasks. Developed by Google researchers, MobileNet introduces a novel approach known as depth-wise separable convolutions to significantly reduce the computational complexity of traditional CNNs. This technique involves decomposing standard convolution operations into two separate layers: a depth-wise convolution and a point-wise convolution. By applying these layers sequentially, MobileNet achieves a remarkable reduction in the number of parameters and computations required, making it particularly well-suited for deployment on resource-constrained platforms.

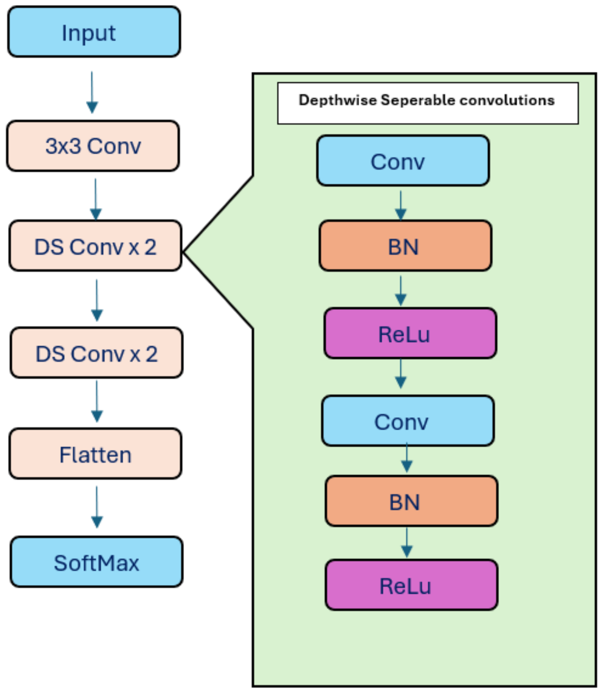

Figure 3.3: Mobilenet Architecture

The architecture of MobileNet is characterized by its depth-wise separable convolutions, which enable efficient inference and low memory footprint without sacrificing performance. MobileNet has since evolved with successive versions, each introducing improvements to further enhance efficiency and accuracy. MobileNetV2, for example, introduced inverted residuals with linear bottleneck layers, which significantly improved efficiency by reducing the computational cost of residual connections. Additionally, MobileNetV3 introduced advanced features such as squeeze-and-excitation blocks and hard-swish activation functions, further optimizing performance for mobile vision applications.

The significance of MobileNet lies in its ability to democratize deep learning on mobile devices, enabling a wide range of applications in fields such as image classification, object detection, and semantic segmentation. By reducing the computational burden without compromising accuracy, MobileNet empowers developers to deploy sophisticated computer vision models on smartphones, tablets, and other edge devices. Its efficiency makes it an ideal choice for real-time applications where latency and resource constraints are critical considerations. As a result, MobileNet has become a cornerstone in the development of mobile vision applications, driving innovation and accessibility in the field of deep learning for mobile platforms.

TensorFlow, developed by Google Brain, is an open-source machine learning framework renowned for its flexibility, scalability, and ease of use. TensorFlow provides comprehensive tools and resources for building, training, and deploying machine learning models across a variety of platforms, including mobile and embedded devices. With its robust ecosystem and support for diverse hardware accelerators, TensorFlow enables developers to seamlessly integrate sophisticated deep learning models such as MobileNet into mobile applications. Furthermore, TensorFlow's optimization techniques, such as model quantization and conversion to TensorFlow Lite format, further enhance the deployment efficiency of deep learning models on resource-constrained platforms.

In our project, we utilized MobileNet to encode graphemes, leveraging its efficient architecture to handle the computational demands of processing visual data on resource-constrained devices. By integrating MobileNet into our workflow, we were able to achieve high performance in grapheme encoding.

The upcoming chapter will detail the proposed approach.

Chapter 4

Proposed Approach

In this section, I outline the methodology employed in our project, which integrates various deep learning models to analyze and extract information from visual data.

Firstly, we utilize the YOLOv3 model as a foundational component of our system. YOLOv3 acts as a robust visual detector, efficiently identifying and delineating individual characters within input images. This is akin to the process of drawing chalk outlines around suspects at a crime scene, where each character is enclosed within a bounding box. These bounding boxes serve as the initial step in organizing and preparing the visual data for further analysis.

Following the detection stage, our approach incorporates specialized models such as M-net and MIp-net to delve deeper into the extracted bounding boxes.

M-net is responsible for decoding the sequence of graphemes represented by each bounding box. It meticulously analyzes the spatial arrangement of characters within the image, sorting them from top to bottom and investigating each row from right to left. This sequential processing mirrors the reading pattern observed in certain languages and ensures accurate character recognition, even in scenarios involving multiple lines of text.

On the other hand, MIp-net focuses on extracting information regarding motifs and symbols present in the input image. By examining the deeper context and symbolism embedded within visual elements, MIp-net enriches our understanding of the image's content beyond mere character recognition.

The collaborative approach of these models allows for efficient processing and extraction of valuable insights from diverse visual data. While YOLOv3 handles the initial detection and organization of characters, M-net and MIp-net specialize in deciphering the identities of characters and extracting contextual information, respectively. This synergy enables our system to provide comprehensive analysis and utilization of visual data stored within our database.

By combining these advanced deep learning techniques, our proposed approach aims to achieve accurate and insightful analysis of visual data, contributing to various applications such as image understanding, text recognition, and content extraction.

The subsequent chapter will provide an in-depth discussion on constructing the model.

Chapter 5

Building The Model

5.1 Bounding Box Creation

5.1.1 Description

Data Collection: The process began with the collection of images containing graphemes. These images likely consisted of text or handwritten characters that needed to be analyzed. In total, 232 images were gathered for training purposes. Annotation: Each image was meticulously annotated to mark the location of individual graphemes. This annotation process likely involved outlining or labeling each grapheme within the image. The annotations were then stored in XML files, which served as a structured format to record the coordinates and other relevant information about each grapheme's position within the image. Model Selection: YOLOv3, short for "You Only Look Once version 3," was chosen as the object detection model for this task. YOLOv3 is known for its efficiency and accuracy in detecting objects within images.

5.1.2 Dataset

Training: The YOLOv3 model was trained using the 232 annotated images. During training, the model learned to recognize the patterns and features associated with graphemes within the images, ultimately enabling it to predict bounding boxes around them. Validation: To assess the performance of the trained model and ensure its generalization ability, a separate set of 13 images with annotations was used for validation. These images were likely selected to represent a diverse range of scenarios and grapheme configurations.

5.2 Grapheme Identification

5.2.1 Description

In the initial approach, Convolutional Neural Networks (CNNs) are employed to recognize characters within bounding boxes due to their adeptness in learning and extracting features from images automatically. The M-net model is integrated into this architecture to provide further refinement in character recognition. Unlike traditional CNNs that operate on entire images, M-net focuses specifically on the characters within bounding boxes, ensuring precise decoding of sequences of graphemes.

During the process, M-net meticulously analyzes the spatial arrangement of characters within each bounding box. It follows a sequential processing approach, sorting characters from top to bottom and examining each row from right to left. This approach mirrors typical reading patterns in certain languages, ensuring accurate character recognition even in complex scenarios involving multiple lines of text or irregular arrangements.

Furthermore, as part of the validation process, multiple layers of CNN-based classification models are utilized. These models work in conjunction with M-net to validate and refine the accuracy of character recognition. The combination of CNN-based classification models and M-net's sequential processing enhances the robustness of character recognition within the bounding boxes.

Additionally, to explore avenues for further improvement, transfer learning techniques are employed. Pre-trained transfer learning-based models, including popular architectures like ResNet and DenseNet, are considered. While traditionally used for image classification tasks, these models can be adapted and fine-tuned to enhance character recognition within bounding boxes. By integrating transfer learning techniques with the M-net model, the initial approach aims to leverage the knowledge and features learned from large datasets to improve the accuracy and efficiency of character recognition in diverse scenarios.

Overall, the M-net model serves as a critical component within the initial approach, contributing to the accuracy and robustness of character recognition within bounding boxes. Its sequential processing, combined with the capabilities of CNN-based classification models and transfer learning techniques, enables comprehensive analysis and extraction of information from visual data.

5.2.2 Dataset

There are a total number of 40 classes(labels) in the dataset. The 40 labels are: M8, M12, M15, M17, M19, M28, M48, M51, M53, M59, M102, M104, M141, M162, M173, M174, M176, M204, M205, M211, M216, M245, M249, M267, M287, M294, M296, M302, M307, M326, M327, M328, M330, M336, M342, M387, M389, M391, Other. The number of Images used for Training - 12,264 (300+ images for each class) The number of Images used for Validation - 200 (5 Images for each class).

5.2.3 M-net Architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 64, 64, 3)         6

 mobilenet_1.00_224 (Functi  (None, None, None, 1024   3228864
 onal)                       )

 flatten (Flatten)           (None, 4096)              0

 dense (Dense)               (None, 64)                262208

 dense_1 (Dense)             (None, 40)                2600

=================================================================
Total params: 3493678 (13.33 MB)
Trainable params: 3471790 (13.24 MB)
Non-trainable params: 21888 (85.50 KB)
_________________________________________________________________
```

Figure 5.1: M-net Model architecture

5.3 Model Accuracy

5.3.1 M-net

The above graph showing the accuracy of a model called the M-net Model. The x-axis of the graph is labeled ”Epoch” and the y-axis is labeled ”Accuracy”. The graph shows that the accuracy of the model increases as the number of epochs increases. The training accuracy is shown in blue and the validation accuracy is shown in green. The highest training accuracy is 0.94 and the highest validation accuracy is 0.95. The model has been trained on 40 classes with around 12,264 images with pre-augmentation. The validation data doesn’t undergo the augmentation which has 200 images in total for all the classes. We can see that the accuracy started with 0.40 which reaches the 0.94 for 10 epochs.

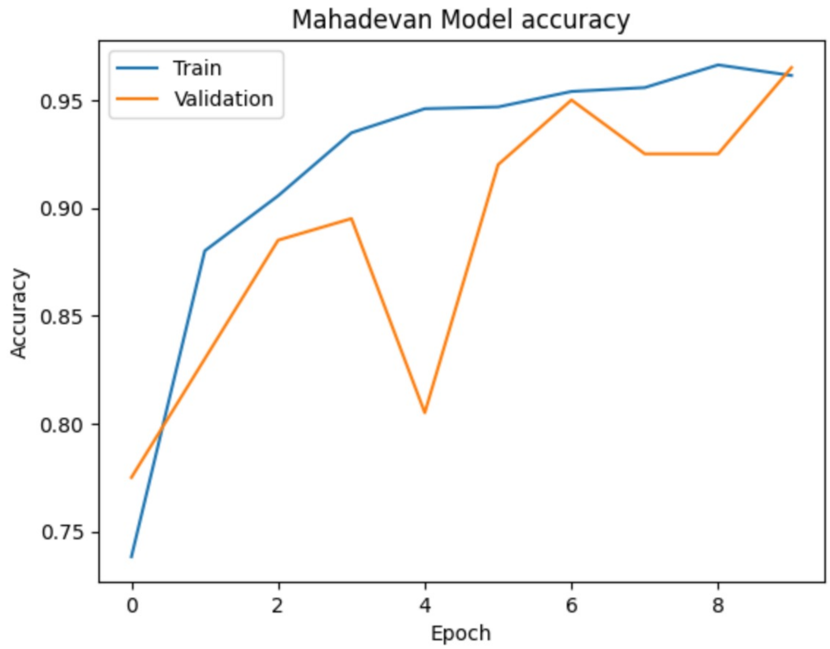

Figure 5.2: M-Net Accuracy

5.4 Motif Identification

5.4.1 Description

The MIP-net model, short for Motif Identification and Prediction Network, is a machine learning model designed for motif identification tasks. In this case, it's specifically trained for identifying motifs in images, particularly the IVC Seal image.

Here's how the process typically works:

Training the MIP-net Model: The MIP-net model is trained using a dataset of IVC Seal images, where each image is associated with a particular motif. The model learns to recognize patterns and features in the images that are indicative of different motifs.

Utilizing 11 Different Classes: The model is trained to classify the motifs into 11 different classes. These classes represent the different motifs that the model can identify. Each class corresponds to a specific motif that the model has been trained to recognize.

Input Image and Prediction: When an IVC Seal image is provided as input to the trained MIP-net model, the model predicts the probability for each of the 11 classes. This is done by passing the image through the trained neural network, which computes the likelihood or confidence score for each motif class.

Selecting the Most Probable Motif: After obtaining the probabilities for each class, the model selects the class with the highest probability as the predicted motif. In other words, the class that the model is most confident about is chosen as the output motif.

Returning the Output: Finally, the predicted motif, along with its associated probability score, is returned as the output of the model. This motif represents the pattern or feature that the model believes is present in the input IVC Seal image.

Overall, the MIP-net model serves as a tool for automatically identifying motifs in IVC Seal images, providing a systematic and efficient way to analyze and categorize these images based on their visual characteristics.

5.4.2 MIP-net Architecture

```
CNNAttention(
  (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv3): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv4): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv5): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=25088, out_features=512, bias=True)
  (fc2): Linear(in_features=512, out_features=11, bias=True)
  (attention): Sequential(
    (0): Conv2d(512, 1, kernel_size=(1, 1), stride=(1, 1))
    (1): Sigmoid()
  )
)
```

Figure 5.3: MIP-net Model architecture

5.4.3 Dataset

There are a total number of 11 classes(labels) in the dataset. 11 Labels used are "buffalo", "bull", "elephant", "horned ram", "man holding tigers", "pashupati", "sharp horn and long trunk", "short horned bull with head lowered towards a trough", "swastik", "tiger looking man on tree", "unicorn".

The number of Images used for Training - 3300. The number of Images used for Augmentation - 55 (5 Images for each class).

5.4.4 MIP-net

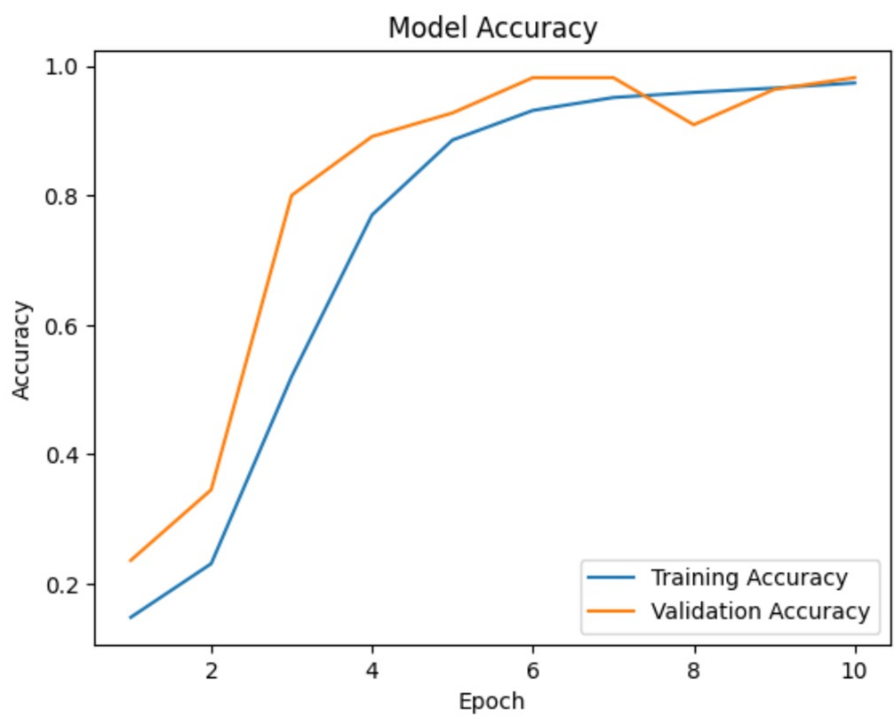

Figure 5.4: MIP-Net Accuracy

The above graph showing the accuracy of a model called the MIP-net Model. The x-axis of the graph is labeled "Epoch" and the y-axis is labeled "Accuracy". The graph shows that the accuracy of the model increases as the number of epochs increases. The training accuracy is shown in blue and the validation accuracy is shown in green. The highest training accuracy is 0.95 and the highest validation accuracy is approximately 0.96. The model has been trained on 11 classes with around 3300 images with pre-augmentation. The validation data doesn't undergo the augmentation which has 55 images in total for all the classes. We can see that the accuracy started with 0.20 which reaches the 0.96 after trained for 10 epochs.

5.5 Database

5.5.1 UML Diagrams

5.5.1.1 Class Diagram

The class diagram illustrates the structure of the system by showing the classes in the system and their relationships. . In this context, the class diagram depicts the

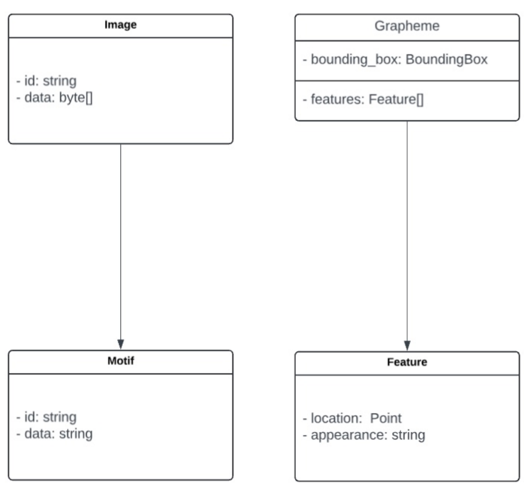

Figure 5.5: Class Diagram

main entities involved in the pipeline, such as Image, Grapheme, and Motif, along with their attributes and associations. It provides an overview of the data structure and relationships within the system, aiding in understanding the organization of the system's components

5.5.1.2 Component Diagram

The component diagram illustrates the physical deployment of components in the system and their interactions.

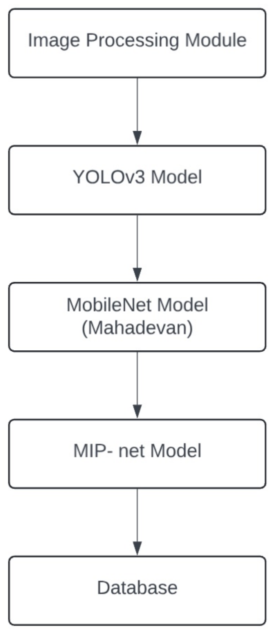

Figure 5.6: Component Diagram

In this context, the component diagram depicts the various components involved in the system, such as the Image Processing Module, YOLOv3 Model, MobileNet Model, MIP-net Model, and Database. It provides an overview of the deployment architecture of the system, showing how different components are interconnected and deployed in the system environment.

5.5.1.3 Sequence Diagram

The sequence diagram illustrates the interactions between objects in the system over time, showing the flow of messages between objects. In this context, the sequence diagram depicts the sequence of actions involved in executing the pipeline, from the user initiating the process to the various components processing the image and storing the data. It helps in understanding the dynamic behavior of the system and the sequence of activities performed during the execution of the pipeline.

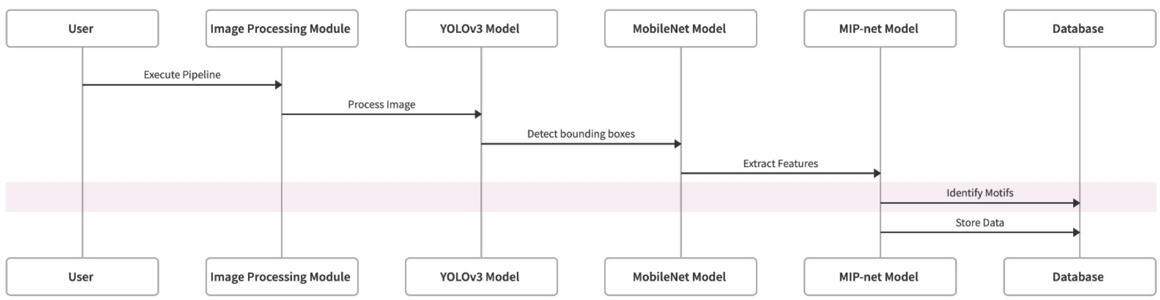

Figure 5.7: Sequence Diagram

5.5.2 Storing the Final Data

5.5.2.1 Description

Storing project results in a SQL database is crucial for data management and accessibility. This step ensures that the valuable insights gained from the previous phases of the project are preserved in a structured and organized manner. Here's a detailed breakdown of the process:

Database Setup: First, a SQL database needs to be set up. This involves creating a new database or using an existing one where the project results will be stored. The database schema should be designed to accommodate the data to be stored, ensuring that it reflects the structure of the project results.

Table Creation: Within the database, tables need to be created to represent different entities or aspects of the project results. For example, there may be a table to store image data, another table for grapheme sequences, and another for motifs. Each table should have appropriate columns to store relevant information, such as ImageID, Image, GraphemeSequence, and Motif.

Data Insertion: Once the tables are set up, the project results can be inserted into the database. This involves executing SQL INSERT statements to add records to the respective tables. For image data, the actual images may be stored in the database as binary large objects (BLOBs) or as file paths pointing to image files stored externally. Grapheme sequences and motifs are typically stored as text or varchar data types.

Data Retrieval and Querying: SQL SELECT statements can be used to extract specific data or perform analysis on the stored information.

Data Integrity and Maintenance: It's essential to ensure data integrity within the database. This involves implementing constraints, such as primary keys, foreign keys, and unique constraints, to maintain data consistency and prevent errors.

Scalability and Performance: As the project progresses and more data is collected, the database should be scalable to accommodate the growing volume of information. Overall, storing project results in a SQL database provides a centralized and structured repository for the data, enabling easy access, analysis, and collaboration among project team members. It ensures that the insights generated from the project are well-preserved and can be leveraged effectively for future research or decision-making purposes.

5.5.3 Sample Queries to Retrieve Data from Database

- SELECT * FROM details; - display all the rows from the details table which represents the ImageID, Image, GraphemeSequence and Motif.

- SELECT * FROM details where Motif in ("swastik","bull") - display all the rows of data where the Motif is either swastik or bull.
- SELECT motif, COUNT(*) AS motifcount FROM details GROUP BY motif; - Count the Number of Rows for Each Motif.
- SELECT COUNT(DISTINCT GraphemeSequence) AS uniquesequences FROM details; - Find the Total Number of Unique Grapheme Sequences.

5.6 End-to-End Workflow of Indus Script Digitization

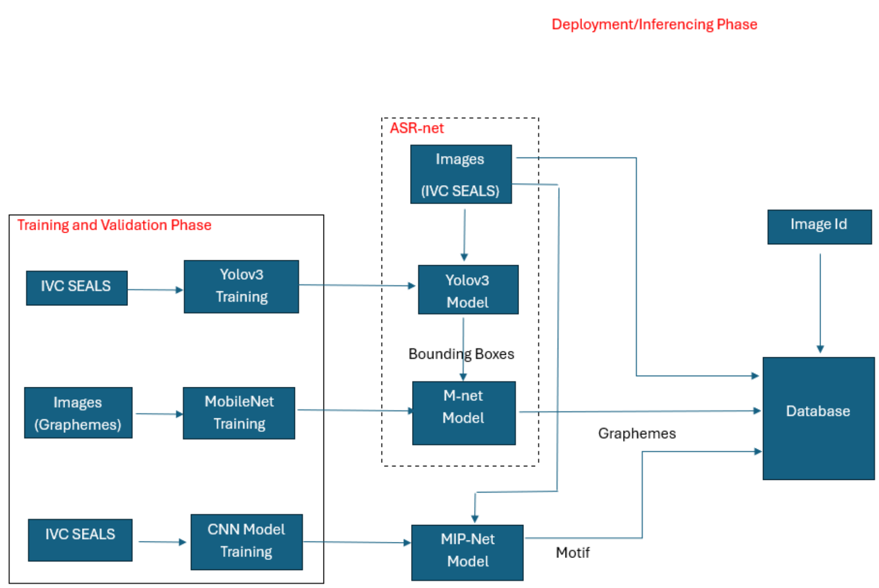

Figure 5.8: Architecture

In the upcoming chapter, you can expect a thorough exploration of the results achieved.

Chapter 6

PipeLine Results : Insights

This pipeline processes images of seals to extract information about graphemes (written symbols) and motifs (patterns) on the seal. Here's a breakdown of each step:

6.1 Input and Preprocessing

The pipeline starts with an image as input. This image is resized and reshaped to match the specific format required by the trained model. This ensures compatibility and optimal processing.

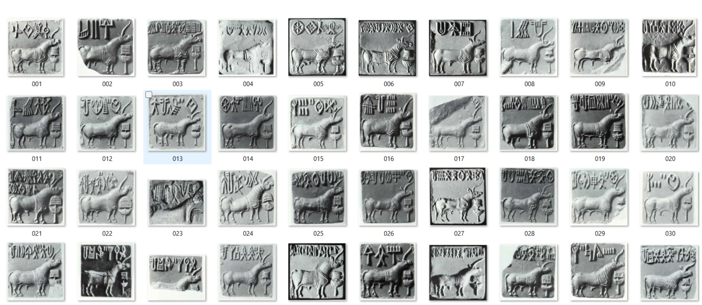

Figure 6.1: The Sample Input Data

6.2 Bounding Box creation

A YOLOv3 architecture is used to detect bounding boxes around each grapheme in the image. YOLOv3 is a powerful object detection model trained to identify specific objects in images. The coordinates of these bounding boxes are stored in a separate file associated with the original image. This file will be used in the next step.

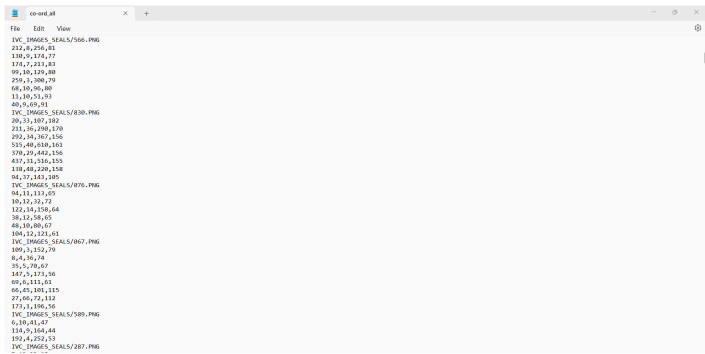

Figure 6.2: Bounding Box Coordinates

6.3 Grapheme Encoding

A MobileNet model named "Mahadevan" takes the grapheme bounding boxes from the previous step as input.

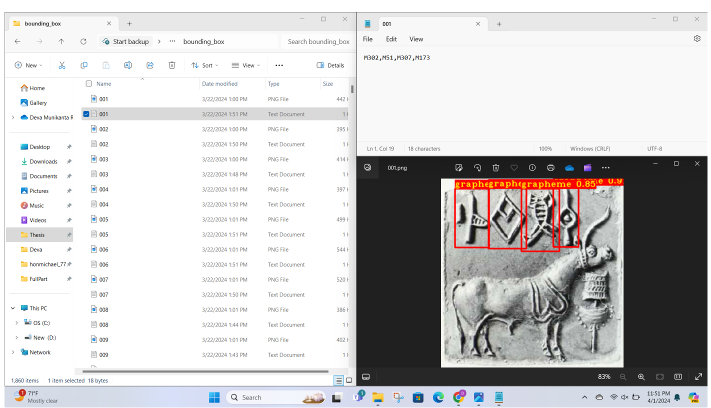

Figure 6.3: Grapheme Sequence

This model extracts features from each grapheme based on its location and appearance. These extracted features are then encoded into text format and stored in a separate text file alongside the original image.

6.4 Motif Identification

The MIP-net model analyzes the original image again, this time focusing on identifying motifs present on the seal. Motifs could be specific patterns, symbols, or designs with meaning. MIP-net extracts information about these motifs and provides it in a format understandable by the system.

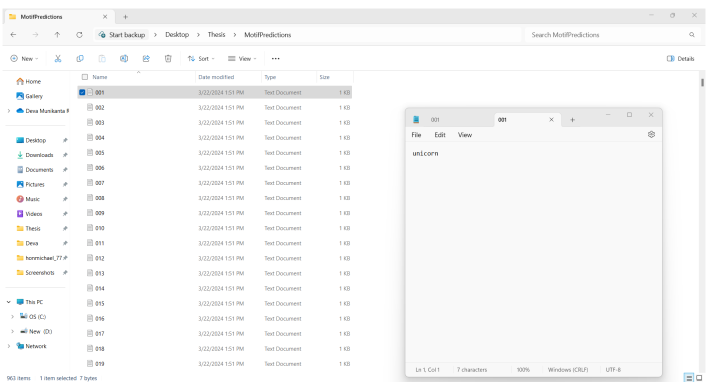

Figure 6.4: Motif

6.5 Database Storage

Finally, all the extracted information is stored in a database. This includes:

- Image ID: A unique identifier for the image.
- Image: The original image itself.
- Grapheme sequence: The order of grapheme encodings from step 3.

- Motif: Information about the identified motifs from step 4.

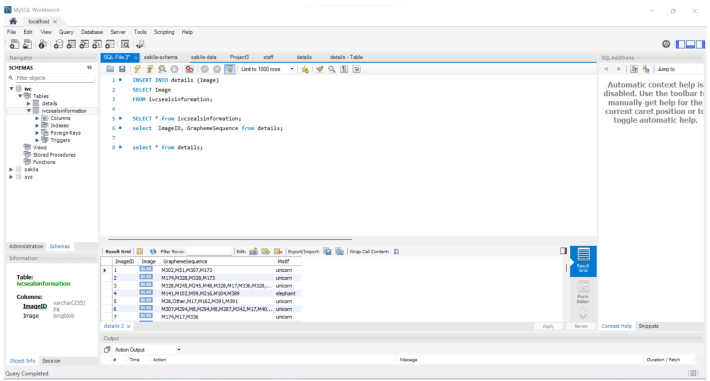

Figure 6.5: Database Structure

The subsequent chapter will outline the intricacies surrounding the challenges encountered.

Chapter 7

Challenges

7.1 Low sample size for some classes

As mentioned before, a few motifs may be sparsely present in any corpora. Please note that any corpora happens to be only a small subset of the seals produced by IC over nearly a thousand years and over a large geographical region.

Figure 7.1: Rarely Found Motif

Even if a motif is observed only once, it needs to be cataloged, and needs to be recognized when observed the next time on a newly found seal/sealing, possibly for deriving crucial information. While this may not be a problem for an experienced archaeologist, it is not feasible for most ML algorithms to learn a motif from only one sample. Extreme disbalance in sample sizes over multiple classes is our first challenge in deep learning.

7.2 Broken seals with only partially visible motif

Many seals are broken during their long burial period, or even discarded for being broken during their active lifetime. A broken seal may not have a reduced importance in archaeological research. For example, the context in which the seal was used, and the motif present on it, may infer the same conclusion irrespective of the seal being broken or not. The motif present on a broken or damaged seal may be only partially visible, and yet, it may be well recognizable by a human by observing only a small part it. Can our ML model be trained to perform at the same level as a human being in recognizing motif from only a small but relevant part of it? We address this question in this work.

Figure 7.2: Broken Seal

7.3 Stylistic variations and uncertain class

Artisans from IVC have curved motifs in many different styles and variations, either for artistic reasons or for conveying some meaning. For identification purpose, archaeologists group all such variability under one motif class or type. For example, the most frequently found Unicorn motif may have two to twelve thread patterns on their necks. Wide variation within a class, which is a challenge for deep learning algorithms, unless each variation is strongly present in the training set. Another problem is that a motif may look like a different one, even to the human eyes. For example, a "horned-zebra" may look like a "unicorn." While most such cases could be discerned easily by an expert with only a closer examination, it is not clear how can one train an ML algorithm to make such discrimination over different motifs that look very similar.

Figure 7.3: Stylistic Variations

The upcoming chapter will cover information about the future prospects and conclusion.

Chapter 8

Future Scope and Conclusion

8.1 Future Scope

8.1.1 Expanding Corpora Size

The future of analyzing ancient civilizations lies in enriching the data available for study. One key approach involves incorporating additional sources like the Parpola/Uesugi corpus and the complete motif list from the Mahadevan corpus. This expanded data pool will allow researchers to delve deeper into the linguistic nuances of ancient texts and uncover the symbolic meanings embedded in artifacts. With a more comprehensive understanding of these elements, we can gain a richer appreciation of the cultural heritage of these lost civilizations.

8.1.2 Broadening the Scope

Our understanding of the past can be further enhanced by moving beyond the study of individual civilizations. Expanding the scope of research to include other ancient societies, like Mesopotamia, Egypt, and Mesoamerica, presents exciting opportunities.

By comparing and contrasting writing systems and cultural practices across different regions and time periods, we can uncover broader patterns and trends in human development. This comparative approach will provide a richer tapestry of human history, allowing us to appreciate the diversity and interconnectedness of ancient civilizations.

8.1.3 Automatic Image Description Techniques

Technological advancements are also poised to revolutionize the analysis of ancient visual artifacts. By pioneering techniques for the automatic description of images, researchers can leverage the power of machine learning and computer vision algorithms. These innovative tools can significantly enhance the efficiency and accuracy of analyzing vast collections of artifacts, leading to a deeper understanding of the visual language employed by these ancient societies.

8.1.4 Technical Aspects

8.1.4.1 Exploring Advanced Object Detection Techniques

As I plan my future projects, I intend to explore advanced techniques for object detection beyond the current framework. While YOLOv5 has gained attention, I will also investigate alternative methodologies that align with my project requirements. By conducting this exploration, I aim to identify solutions that can significantly enhance object detection performance, ensuring the reliability and accuracy of my system.

8.1.4.2 Improving Grapheme Localization Precision

In my future projects, I aim to refine grapheme detection accuracy by investigating enhancements to localization precision. This includes evaluating the implementation of a four-coordinate format for bounding boxes to achieve finer granularity in character recognition. By adopting such an approach, I anticipate elevating the overall efficacy and performance of my systems for text analysis tasks.

8.1.4.3 Exploring Alternative Deep Learning Frameworks

Looking ahead to my future projects, I am eager to explore alternative deep learning frameworks beyond TensorFlow. While TensorFlow has been invaluable, I recognize the value in diversifying my toolkit with frameworks like PyTorch. Through this exploration, I aim to leverage unique features and streamline development workflows, ultimately enhancing the effectiveness and adaptability of my machine learning solutions.

These interdisciplinary endeavors, combining traditional archaeological methods with cutting-edge technology, hold immense promise for the future of our understanding of past civilizations. By enriching our data sources, broadening our scope of inquiry, and utilizing advanced image analysis techniques, we can unlock the secrets of the past and gain a deeper appreciation for the richness and complexity of human cultural heritage.

8.2 Conclusion

In conclusion, the introduction of ASR-net, a combination of M-net and YOLOv3, marks a significant advancement in the field of ancient script analysis, particularly focusing on the enigmatic Indus script. Achieving an impressive success rate in identifying individual symbols, ASR-net addresses the critical need for automated techniques in digitizing ancient Indus seals, akin to Optical Character Recognition systems for modern languages. Furthermore, this research introduces the Motif Identification Problem (MIP), shedding light on recurring patterns (motifs) found on Indus seals, which are believed to hold specific functions within certain periods of the civilization. Despite the challenges associated with applying deep learning to MIP, the creation of an open-source dataset of annotated seals serves as a crucial stepping stone for further theoretical archaeological research on the Indus Valley Civilization. Through the integration of advanced technological approaches and interdisciplinary collaboration, this research contributes to the ongoing efforts to decipher the ancient mysteries embedded within the artifacts of the Indus Valley Civilization.

The following news items have been made about the project:

- phys.org
- Infobae.com
- Omnia.com

In the chapter that follows, you'll find a thorough exposition on the bibliography.

Bibliography

[1] Berkeley Vision and Learning Center. BVLC GooLeNet ILSVRC 2014 Snapshot. `https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet`.

[2] Andrew Brock and et al. Biggan: Large scale gan training for high fidelity natural image synthesis. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[3] Shruti Daggumati and Peter Z. Revesz. A method of identifying allographs in undeciphered scripts and its application to the indus valley script. *Humanities and Social Sciences Communications*, 8(50):1–14, 2021.

[4] Jia Deng and et al. Imagenet: A large-scale hierarchical image database. 2009.

[5] Mark Everingham and et al. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[6] Ross Girshick. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2015.

[7] Ian Goodfellow and et al. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[8] Kaiming He and et al. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[9] Kaiming He and et al. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2017.

[10] Gao Huang and et al. Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[12] Yann LeCun and et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[14] Tsung-Yi Lin and et al. Feature pyramid networks for object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[15] Wei Liu and et al. Ssd: Single shot multibox detector. *European Conference on Computer Vision*, 2016.

[16] Ansumali Mukhopadhyay and An. Ancestral dravidian languages in indus civilization: Ultraconserved dravidian tooth-word reveals deep linguistic ancestry and supports genetics. *Humanit Soc Sci Commun*, 8(193):193, 2021.

[17] Michael Oakes and Michael P. Oakes. Statistical analysis of the tables in mahadevan's concordance of the indus valley script. *Journal of Quantitative Linguistics*, 26(4):401–422, 2019.

[18] S Palaniappan and R. Adhikari. Deep learning indus script. *PLOS Submission*, 2017.

[19] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[20] Asko Parpola. The indus script: A challenging puzzle. *World Archaeology*, 17(3):399–419, Feb. 1986.

[21] Venkatesh-Prasad Ranganath and John Hatcliff. An overview of the indus framework for analysis and slicing of concurrent java software (keynote talk - extended abstract). In *Proceedings of the Sixth IEEE International Workshop on Source Code Analysis and Manipulation (SCAM '06)*. IEEE Xplore, October 2006.

[22] S. R. Rao. Indus script and language. *Annals of the Bhandarkar Oriental Research Institute*, 61(1/4):157–188, 1980.

[23] V. N. Rao and M. K. Mohanty. Comparative visual analysis of symbolic and illegible indus valley script with other languages. *IOSR Journal of Humanities and Social Science (IOSR-JHSS)*, 20(2):66–72, 2015.

[24] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 2018.

[25] Shaoqing Ren and et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[26] Olga Russakovsky and et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[28] Christian Szegedy and et al. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[29] Christian Szegedy and et al. Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[30] Ali A. Vahdati and Raffaele Biscione. The seals from khorsan. In A. Parpola and P. Koskikallio, editors, *Corpus of Indus Seals and Inscriptions 3.3 (CISI 3.3)*, pages l–lvi. Printed in Finland by Kirjapaino Hermes Oy, Tampere, Helsinki, jan 2022.

[31] Varun Venkatesh and Ali Farghaly. Identifying anomalous indus texts from west asia using markov chain language models. pages 1–7, 2023.

[32] Jason Yosinski and et al. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27:3320–3328, 2014.