SilentScript - Deep Learning Based Damage Restoration of Ancient Inscription Characters

W.M.S. Mesandu

(IT22108890)

BSc (Hons) in Information Technology

Specializing in Data Science

Department of Data Science

Sri Lanka Institute of Information Technology

Sri Lanka

April 2026

SilentScript - Deep Learning Based Damage Restoration of Ancient Inscription Characters

W.M.S. Mesandu
(IT22108890)

(Final documentation in partial fulfilment of the requirement for the Degree of Bachelor of Science Special (honors) In Information Technology Specializing in Data Science)

BSc (Hons) in Information Technology Specializing in Data Science

Department of Data Science

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

DECLARATION OF THE CANDIDATE AND SUPERVISOR

I declare that this is my own work and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
W.M.S. Mesandu	IT22108890	

The supervisor/s should certify the proposal report with the following declaration.

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

……………………………..   ……………………

Signature of the supervisor: Date

……………………………..   ……………………

Signature of the co-supervisor: Date

ABSTRACT

Ayurvedic medicines have a vital role in preserving physical and mental health of human beings. Especially in Sri Lanka we have our own set of rare Ayurvedic herbs which have been utilized by generations as medicinal treatments for a variety of diseases. Lack of experts in this field makes proper identification and classification of medicinal plants a tedious task, which is essential for better treatment. Hence, a fully automated system for medicinal plant classification is highly desirable. Automatic classification of trees from leaves is a popular computer vision/machine learning task and has important applications. With this approach, there are existing applications which can identify plants with low prediction accuracies. However, these applications are based on foreign plant data sets that do not include valuable herbs and shrubs with medicinal qualities. Hence this research proposes a centralized mobile application unique to medicinal plants, which can identify a group of ayurvedic herbs in Sri Lanka, which is able to be performed in both online and offline approach. Here, a new ayurvedic plant dataset prepared from scratch, and preliminary results for classification of 5 types of herbs, compared with several CNN models based on transfer learning are presented. The major objective of the study was to analyze the noisy image set using deep neural network architectures based on transfer learning, choose the best architecture, and create a deep learning model that can be applied effectively for this application. The methodology included gathering data, and transfer learning based on deep Convolutional Neural Networks used on noisy image set for processing them using TensorFlow in a local computer. Images were retrained on the available neural network architectures, fine-tuned from pre-trained weights and then the best architecture was selected. The selected algorithm was fine-tuned using data augmentation techniques on the labeled dataset and hyper-parameter tuning. Experimental results indicate VGG-16 as the best CNN classification model which reached a promising testing accuracy of 99.53%. Conclusively, the outcome of this study will be able to be used by locals in identifying herbal plants worldwide accurately.

Keywords: Ayurvedic leaf classification, computer vision, convolutional neural network, deep learning, transfer learning

ACKNOWLEDGEMENT

Several people played important roles in accomplishing this research. First, I would like to express my deepest appreciation and full-hearted gratitude to my supervisor Mrs. Thamali and co-supervisor Mr. Samadhi who guided us to success in this research and gave entire support to complete this project. Also, I would like to thank Sri Lanka Institute of Information Technology for encouraging us to carry out this study and the academic staff for the continuous guidance, support, and insightful comments, which motivated us to move further. Furthermore, I am grateful to all the resource people for collecting relevant information as well as photographs of some selected valuable inscriptions in Sri Lanka. Finally, I wish to extend my warm appreciation to my family members for their encouragement and support to complete this task.

TABLE OF CONTENTS

LIST OF FIGURES

9

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN-Artificial Neural Network

CNN-Convolutional Neural Network

MSF-CNN-Multi-Scale fusion Convolutional Neural Network

DNN-Deep Neural Network

WHO-World Health Organization

HOG- Histograms of Oriented Gradients

LBP- Local Binary Patterns

SVM- Support Vector Machines

ELM- Extreme Learning Machines

PCA- Extreme Learning Machines

KNN- K-Nearest Neighbor

PNN- Probabilistic Neural Network

CBIR- Content-Based Image Retrieval

1. INTRODUCTION

Ancient stone inscriptions are one of the most valuable historical resources for understanding the cultural, political, religious, and linguistic heritage of Sri Lanka. These inscriptions preserve important information about ancient kingdoms, administrative systems, social structures, and the evolution of early writing systems such as Brahmi inscriptions. However, many of these inscriptions have been exposed to natural weathering, erosion, biological growth, cracks, and physical damage over long periods of time, causing many characters to become unclear, partially missing, or heavily damaged.

The process of identifying and restoring damaged inscription characters is traditionally performed manually by archaeologists, historians, and epigraphists. This process requires extensive domain knowledge, high levels of expertise, and significant time. Manual restoration is often difficult because many inscription characters are incomplete, visually degraded, or affected by environmental noise such as shadows, surface cracks, and stone texture variations. In many cases, the restoration process depends heavily on expert interpretation, which may introduce subjectivity and inconsistency.

With the advancement of computer vision, image processing, and deep learning techniques, automated restoration of damaged historical inscriptions has become a promising research area. Deep learning models, especially convolutional neural networks (CNNs), have shown strong performance in image restoration, segmentation, denoising, and reconstruction tasks. These techniques can help improve the readability of damaged inscriptions by learning visual patterns from existing clean and damaged character samples.

In this research, a deep learning-based approach is proposed for the restoration of damaged inscription characters. The system uses paired datasets consisting of clean character images and synthetically damaged versions of those same characters. Synthetic damage generation is applied using methods such as black blobs, crack lines, eraser patches, white patches, Gaussian noise, and blur to simulate realistic inscription damage conditions found in ancient stone surfaces . An Improved U-Net architecture with residual blocks is used as the core restoration model to reconstruct damaged characters into cleaner and more readable forms .

The final system supports both known and unknown inscription restoration. Known characters can be matched directly using hash-based retrieval from the database, while unknown damaged characters are restored using the trained deep learning model through the restoration pipeline implemented in the

API system . This approach helps preserve historical inscription data while reducing manual effort and improving restoration efficiency.

This research contributes to the digital preservation of ancient inscriptions and supports historians, archaeologists, and researchers by providing an intelligent restoration system for damaged inscription characters.

1.1 Background & Literature Survey

An overall idea about the importance of Ayurveda in Sri Lanka, current problems faced by people in this domain, apparent tasks that should be there in any automated identification and classification as well as the limitations of previous researches carried out regarding automated leaf classification worldwide have been given in the introduction section. This section brings several works and important attention in the focus of this research which is plant species identification and classification.

As mentioned in the introduction section, researchers have tried many methodologies to extract the features and identify the plant species automatically. According to previous work done, there are existing applications which can identify plants with low prediction accuracies. However, these applications are based on foreign plant data sets that do not include valuable herbs and shrubs with medicinal qualities. Additionally, most of these methods make use of the combination of many parameters like color, shape, texture features etc.

In 2019, the research paper "Oak Leaf Classification: An Analysis of Features and Classifiers" [14] has presented a new oak leaf dataset and preliminary results for classification of 8 types of oak trees. The novelties include comparative analysis of a small set of hand-crafted geometric features and popularly used high-dimensional appearance features, such as Local Binary Patterns (LBP) and Histograms of Oriented Gradients (HOG). They have further compared commonly used Support Vector Machines (SVM) classifier with a recently popular, fast and robust learner called Extreme Learning Machines (ELM). Results indicate that a small set of geometric features reach an accuracy of 75%, while high dimensional appearance features can boost the performance up to92%.

In 2019, in the study of "Combination of Deep Features and KNN Algorithm for Classification of Leaf-Based Plant Species" [15], they have proposed an approach based on the combination of deep architectures. Deep features were extracted from the plant leaves using the fc6 layer of the previously trained AlexNet and VGG16 models. Then, the reduction of the number of deep features by using the Principal Component Analysis (PCA) method was done quickly and the best distinguishing features were obtained. Finally, the classification performances were calculated using the K-Nearest Neighbor (KNN) method. Flavia and Swedish plant leaf data sets were used to test the proposed system. According to the experimental results, the accuracy scores for Flavia and Swedish data sets was obtained as 99.42% and 99.64%, respectively.

In 2019, in the study of “AyurLeaf: A Deep Learning Approach for Classification of Medicinal Plants”[20], they have proposed AyurLeaf, a Deep Learning based Convolutional Neural Network (CNN) model, to classify medicinal plants using leaf features such as shape, size, color, texture etc. This research work also has proposed a standard dataset for medicinal plants, commonly seen in various regions of Kerala, the state on southwestern coast of India. The proposed dataset contains leaf samples from 40 medicinal plants. A deep neural network inspired from Alexnet is utilized for the efficient feature extraction from the dataset. Finally, the classification is performed using Softmax and SVM classifiers. Their model, upon 5-cross validation, achieved a classification accuracy of 96.76% on AyurLeaf dataset.

In 2019, in the study of “Fine-Grained Plant Identification using wide and deep learning model” [21], they have proposed a model to address the fine-grained plant image classification task by using the wide and deep learning framework which combines a linear model and a deep learning model. Proposed method sums the result of the wide and deep learning model using a logistic function so that discrete features can be considered simultaneously with continuous image content. Their works have used metadata such as the date of flowering and locational information for the wide model. Their experiment shows that the proposed method gives better performance than a baseline method.

In 2018, the study of “A Multiscale Fusion Convolutional Neural Network for Plant Leaf Recognition” [16], a multiscale fusion convolutional neural network (MSF-CNN) is proposed for plant leaf recognition at multiple scales. First, an input image is down-sampled into multiples low resolution im- ages with a list of bilinear interpolation operations. Then, these input images with different scales are step-by-step fed into the MSF-CNN architecture to learn discriminative features at different depths. At this stage, the feature fusion between two different scales is realized by a concatenation operation, which concatenates feature maps learned on different scale images from a channel view. Along with the depth of the MSF-CNN, multiscale images are progressively handled, and the corresponding features are fused. Third, the last layer of the MSFCNN aggregates all discriminative information to obtain the final feature for predicting the plant species of the input image. Experiments show the proposed MSF-CNN method is superior to multiple state-of-the art plant leaf recognition methods on the MalayaKew Leaf dataset and the Leaf Snap Plant Leaf dataset.

In 2018, the study of "Improving Plant Recognition using Hybrid features from Connectionist and Knowledge-Based Approaches" [22] has proposed architecture that combined knowledgebased approach to improve the accuracy of plant recognition. Towards this, hybrid features are constructed by merging three types of knowledge-based features; morphological feature, texture feature and color feature with convolutional neural network extracted features. Their architecture consists of three main stages which are data pre-processing, feature extraction and classification. Before features are extracted, images will be resized and augmented in the pre-processing stage. To classify the species of leaf, they consider decision tree and artificial neural network as a classifier. They have experimented on two datasets: Flavia and Swedish dataset. The experimental result shows that the proposed architecture can predict unseen images correctly more than existing models.

In 2017, the study of "Plant Classification using Convolutional Neural Networks" [23] has proposed a Convolutional Neural Network (CNN) architecture to classify the type of plants from the image sequences collected from smart agro-stations. First challenges introduced by illumination changes and deblurring are eliminated with some preprocessing steps. Following the preprocessing step, Convolutional Neural Network architecture is employed to extract the features of images. The construction of the CNN architecture and the depth of CNN are crucial points that should be emphasized since they affect the recognition capability of the architecture of neural networks. In order to evaluate the performance of the approach proposed in this paper, the results obtained through CNN model are compared with those obtained by employing SVM classifier with different kernels, as well as feature descriptors such as LBP and GIST. The performance of the approach is tested on dataset collected through a government supported project, TARBIL, for which over 1200 agro-stations are placed throughout Turkey. The experimental results on TARBIL dataset confirm that the proposed method is quite effective.

In 2016, the study of "Ayurvedic Plant Species Recognition using Statistical Parameters on Leaf Images" [30], has proposed a simple and efficient methodology for Ayurvedic plant classification using digital image processing and machine vision technology. The three major phases in proposed methodology are pre-processing, feature extraction and classification. Preprocessing is done in order to highlight the relevant features to be used in the proposed methodology as well as to reduce unwanted noise from the input image, which reduces the chance of getting optimal feature values. In feature extraction phase, different morphologic features such as mean, standard deviation, convex hull ratio, isoperimetric quotient, eccentricity and entropy are extracted from the pre-processed leaf image. In the third phase, a new approach to classify ayurvedic plant species is adopted to recognize plant species by calculating the leaf factor of the input leaf using the extracted feature values and it is compared with the trained values that are stored in the database. An accuracy of 93.75% is obtained for the proposed methodology.

In 2015, the study of "Recognition of Whole and Deformed Plant Leaves using Statistical Shape Features and Neuro-Fuzzy Classifier" [24] has proposed a methodology for recognition of plant species by using a set of statistical features obtained from digital leaf images. As the features are sensitive to geometric transformations of the leaf image, a preprocessing step is initially performed to make the features invariant to transformations like translation, rotation and scaling. Images are classified to 32 pre-defined classes using a Neuro fuzzy classifier. Comparisons are also done with Neural Network and k-Nearest Neighbor classifiers. Recognizing the fact that leaves are fragile and prone to deformations due to various environmental and biological factors, the basic technique is subsequently extended to address recognition of leaves with small deformations. Experimentations using 640 leaf images varying in shape, size, orientations and deformations demonstrate that the technique produces acceptable recognition rates.

In 2015, the study of "A Convolutional Neural Network for Leaves Recognition Using Data Augmentation" [25], a seven-layer ConvNet using data augmentation is proposed for leaves recognition. First, they implement multiform transformations (e.g., rotation and translation etc.) to enlarge the dataset without changing their labels. This novel technique recently makes tremendous contribution to the performance of ConvNets as it is able to reduce the over-fitting degree and enhance the generalization ability of the ConvNet. Moreover, in order to get the shapes of leaves, they sharpen all the images with a random parameter. This method is similar to the edge detection, which has been proved useful in the image classification. Then have trained a deep convolutional neural network to classify the augmented leaves data with three groups of test set and finally find that the method is quite feasible and effective. The accuracy achieved by their algorithm outperforms other methods for supervised learning on the popular leaf dataset Flavia.

In 2015, the study of “DEEP-PLANT: PLANT IDENTIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS” [26] studies convolutional neural networks (CNN) to learn unsupervised feature representations for 44 different plant species, collected at the Royal Botanic Gardens, Kew, England. To gain intuition on the chosen features from the CNN model (opposed to a ’black box’ solution), a visualization technique based on the deconvolutional networks (DN) is utilized. It is found that venations of different order have been chosen to uniquely represent each of the plant species. Experimental results using these CNN features with different classifiers show consistency and superiority compared to the state-of-the art solutions which rely on handcrafted features.

In 2014, the study of “Ayurvedic leaf recognition for Plant Classification” [31], the performance of different features extraction methods are compared, different combinations of features and a number of classifiers applied for leaf identification process are also discussed.

In 2012, the study of “An Efficient Leaf Recognition Algorithm for Plant Classification Using Support Vector Machine” [27] uses an efficient machine learning approach for the classification purpose. This proposed approach consists of three phases such as preprocessing, feature extraction and classification. The preprocessing phase involves a typical image processing steps such as transforming to gray scale and boundary enhancement. The feature extraction phase derives the common DMF from five fundamental features. The main contribution of this approach is the Support Vector Machine (SVM) classification for efficient leaf recognition. 12 leaf features which are extracted and orthogonalized into 5 principal variables are given as input vector to the SVM. Classifier tested with flavia dataset and a real dataset and compared with kNN approach, the proposed approach produces very high accuracy and takes very less execution time.

In 2012, the study of “Classification of Selected Medicinal Plants Leaf Using Image Processing” [32] has aimed at implementing a system using image processing with images of the plant leaves as a basis of classification. The software returns the closest match to the query. The proposed algorithm is implemented, and the efficiency of the system is found by testing it on 10 different plant species. The software is trained with 100 (10 number of each plant species) leaves and tested with different plant species) leaves. The efficiency of the implementation of the proposed algorithms is found to be 92%.

In 2010, the study of "Plant Leaf Classification Using Plant Leaves Based on Rough Set" [28] a method of plant leaf classification is proposed based on the neighborhood rough set. They mainly show that this is applicable to plant leaf classification. Experimental results on plant leaf image database demonstrate that the proposed method is effective and feasible for leaf classification.

In 2008, the study of "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network" [29] they employ Probabilistic Neural Network (PNN) with image and data processing techniques to implement a general-purpose automated leaf recognition for plant classification. 12 leaf features are extracted and orthogonalize into 5 principal variables which consist the input vector of the PNN. The PNN is trained by 1800 leaves to classify 32 kinds of plants with an accuracy greater than 90%. Compared with other approaches, their algorithm is an accurate artificial intelligence approach which is fast in execution and easy in implementation.

1.2 Research gap

The restoration of ancient inscription characters remains a challenging and partially unsolved problem in the fields of computer vision, historical document preservation, and digital archaeology. Although many studies have been conducted on image restoration, document enhancement, and character recognition, very limited research has specifically focused on the restoration of damaged stone inscription characters, especially in the context of ancient Sri Lankan inscriptions.

Traditional image restoration techniques such as thresholding, filtering, and exemplar-based image inpainting have shown reasonable performance for general image repair tasks [3], [4]. However, these methods are often insufficient when handling inscription characters affected by severe erosion, cracks, missing regions, biological growth, and long-term weathering. Ancient stone inscriptions contain highly irregular damage patterns, making restoration significantly more complex than standard document restoration.

Similarly, Optical Character Recognition (OCR) systems and document binarization methods have been widely used for printed and handwritten historical documents [11], [12]. However, these systems usually assume relatively clean and readable text regions. In the case of ancient inscriptions, characters are often incomplete, fragmented, or partially invisible due to surface degradation, making direct OCR application unreliable without prior restoration.

Deep learning methods such as Convolutional Neural Networks (CNNs), Fully Convolutional Networks (FCNs), and U-Net architectures have demonstrated strong results in segmentation and restoration tasks [1], [8]. However, most existing implementations focus on medical imaging, scanned documents, or modern handwritten text rather than ancient stone inscriptions. The structural complexity and unique visual characteristics of inscription characters are significantly different from these domains.

Furthermore, there is a lack of publicly available paired datasets containing both damaged and clean versions of inscription characters. This creates a major limitation for supervised deep learning approaches. Most previous studies rely on naturally damaged images without accurate clean ground truth references, reducing restoration reliability. In this research, synthetic damage generation is introduced to overcome this limitation by creating paired damaged-clean datasets using controlled artificial degradation methods such as black blobs, crack lines, white patches, Gaussian noise, blur, and erosion simulation.

The major research gap identified in this study is the absence of an automated deep learning-based restoration system specifically designed for ancient inscription character damage restoration using paired character-level datasets and residual U-Net architecture. Existing systems either focus only on segmentation, OCR, or general historical document restoration without addressing the full restoration of damaged inscription characters.

Therefore, this research aims to fill this gap by proposing a specialized restoration framework that supports both known inscription matching through database retrieval and unknown damaged character restoration using a trained deep learning model. This approach contributes to preserving ancient inscription knowledge while reducing manual restoration effort and improving restoration consistency and accuracy.

Reference	**Existing System / Method**	**Functionality**	**Technology Used**	**Limitations**
Bertalmio et al. [4]	Image Inpainting	Restores missing image regions using neighboring pixel propagation	PDE-based Image Inpainting	Poor performance for complex structural damage
Criminisi et al. [3]	Exemplar-Based Inpainting	Restores damaged regions using texture patch copying	Patch-based Inpainting	Cannot preserve character stroke structure accurately
Smith [11]	OCR for Documents	Recognizes printed and scanned text	OCR Engine (Tesseract)	Fails when characters are incomplete or heavily damaged

Diem et al. [12]	Historical Document Restoration	Binarization and enhancement of old manuscripts	Document Binarization + Restoration	Focuses on manuscripts, not stone inscriptions

Table 1.2.1: Comparison with existing systems and identifying gaps

1.3. Research problem

Ancient stone inscriptions are one of the most valuable sources of historical knowledge, preserving important information about ancient civilizations, administrative systems, religious practices, and the development of early writing systems. In Sri Lanka, Brahmi and other early inscriptions play a major role in understanding the country's archaeological and cultural heritage. However, many of these inscriptions have been exposed to natural weathering, erosion, cracks, biological growth, and physical damage over long periods of time, causing characters to become unclear, incomplete, or permanently damaged.

At present, the restoration and interpretation of damaged inscription characters are mainly performed manually by archaeologists, historians, and epigraphists. This process requires extensive domain expertise, deep historical knowledge, and significant time. Manual restoration is often difficult because many characters are only partially visible, affected by environmental noise, or damaged beyond simple recognition. Since restoration depends heavily on expert judgment, different interpretations may occur, leading to inconsistency and reduced reliability.

Traditional image processing techniques such as thresholding, filtering, segmentation, and inpainting provide only limited support for restoring damaged inscription characters. These methods often fail when the damage is severe, especially when important character strokes are missing or when cracks and erosion distort the original structure. Similarly, Optical Character Recognition (OCR) systems cannot perform accurately when characters are fragmented or heavily degraded because OCR requires relatively clean and complete character inputs.

Researchers working in archaeology, epigraphy, linguistics, and digital preservation face significant challenges when attempting to preserve and interpret ancient inscriptions. Identifying damaged characters manually across large collections of inscriptions is both labor-intensive and inefficient. In addition, the absence of automated restoration systems specifically designed for inscription character reconstruction creates a major limitation in preserving valuable historical knowledge.

Another major problem is the lack of paired datasets containing both damaged and clean versions of inscription characters. Since ancient inscriptions naturally exist only in damaged form, obtaining accurate clean ground truth data is extremely difficult. This limits the effectiveness of supervised deep learning approaches for restoration tasks.

Therefore, there is a strong need for an automated and reliable inscription character restoration system that can reconstruct damaged characters accurately while preserving their original visual structure. Such a system should reduce manual effort, improve restoration consistency, and support the digital preservation of historical inscriptions for future generations.

This research addresses this problem by proposing a deep learning-based restoration system using synthetic damage generation, paired damaged-clean character datasets, and an Improved U-Net model with residual blocks to restore damaged inscription characters efficiently and accurately.

1.4. Research Objectives

1.4.1 Main Objective

The primary objective of the proposed solution is to develop an automated deep learning-based system for restoring damaged ancient inscription characters using image processing and deep neural networks. The system should be capable of taking damaged inscription character images as input and producing restored, cleaner, and more readable character images as output.

The restoration process is intended to support archaeologists, historians, epigraphists, and researchers by reducing the difficulty of manually restoring damaged inscriptions. The proposed system aims to preserve the structural integrity of the original inscription characters while improving readability and restoration consistency.

The restoration functionality is achieved through a hybrid approach that supports both known and unknown inscription characters. Known characters are identified using database matching techniques, while unknown damaged characters are restored using a trained deep learning model based on an Improved U-Net architecture with residual blocks.

The methodology and development strategy used in this research can be further extended for large-scale inscription preservation systems and future historical document restoration applications.

1.4.2 Specific Objectives

1. To prepare a paired dataset consisting of clean inscription character images and their corresponding damaged versions for supervised deep learning training.
2. To generate realistic synthetic damage on clean character images using methods such as black blobs, crack lines, white patches, Gaussian noise, blur, and erosion simulation in order to mimic real-world inscription damage conditions.
3. To design and implement an Improved U-Net architecture integrated with residual blocks for restoring damaged inscription characters with higher visual quality and structural accuracy.

4. To compare restoration performance and improve model effectiveness using preprocessing, data augmentation, and optimization techniques during training.
5. To implement a hybrid restoration system that supports both direct database retrieval for known characters and deep learning-based restoration for unknown damaged characters.
6. To develop an API-based restoration pipeline that allows practical use of the trained model for inscription restoration and future integration with larger inscription digitization systems.

2. METHODOLOGY

2.1 Understanding the key pillars of the research domain

The proposed system mainly relies on the key pillars of Deep Learning, Convolutional Neural Networks (CNN), U-Net architecture, Residual Learning, and Synthetic Damage Generation. These concepts form the technical foundation of the inscription character damage restoration system and directly support the restoration of damaged ancient inscription characters.

2.1.1 Deep Learning

Traditional image restoration and document enhancement methods mainly rely on classical image processing techniques such as thresholding, filtering, segmentation, and morphological operations. Although these methods can improve image visibility to a certain extent, they often fail when handling severe structural damage such as missing character strokes, deep cracks, erosion, and surface degradation commonly found in ancient stone inscriptions.

Deep learning is a branch of machine learning in which neural networks automatically learn meaningful representations directly from image data without requiring manual feature extraction. Instead of depending on handcrafted features, deep learning models learn hierarchical visual patterns from large datasets and improve performance through repeated training and optimization. According to LeCun et al., deep learning has become one of the most powerful techniques in computer vision because of its ability to handle large-scale image analysis tasks with high accuracy [5]. It has shown strong performance in image classification, segmentation, denoising, object detection, and image restoration tasks. Higher accuracy, automatic feature learning, and the ability to process complex visual patterns make deep learning highly suitable for damaged inscription restoration.

In this research, deep learning is used to restore damaged inscription characters by learning the relationship between damaged character images and their corresponding clean versions. The model is trained using paired datasets where synthetic damage is applied to clean character images to simulate real-world inscription degradation. Through this supervised learning process, the network learns how to reconstruct missing regions and recover damaged character structures.

Since ancient inscriptions contain complex visual noise and irregular damage patterns, deep learning provides a more reliable and scalable solution compared to traditional restoration methods. Therefore, it forms the core foundation of the proposed restoration system.

2.1.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning architecture specifically designed for image processing and visual recognition tasks. CNNs are highly effective in extracting spatial features from images such as edges, textures, shapes, and structural patterns. Because of this capability, CNNs are widely used in image classification, segmentation, restoration, denoising, and reconstruction problems.

Unlike traditional neural networks, CNNs preserve the spatial relationship between pixels using convolution operations. A CNN mainly consists of convolution layers, activation layers, pooling layers, and fully connected layers. The convolution layers automatically learn important visual features, while pooling layers reduce computational complexity and improve generalization. This allows the network to detect both low-level features such as lines and curves and high-level features such as complete character shapes.

According to Krizhevsky et al., deep CNNs significantly outperform traditional handcrafted feature-based methods in image recognition tasks [7]. Similarly, Fully Convolutional Networks (FCNs) introduced by Long et al. demonstrated that convolution-based architectures can perform highly accurate pixel-level segmentation and reconstruction tasks [8]. These findings show that CNNs are highly suitable for restoration tasks involving damaged text and ancient inscriptions.

In the context of inscription restoration, the damaged characters often contain broken strokes, missing regions, surface cracks, and visual noise caused by erosion and environmental effects. Traditional image processing methods struggle to restore such complex patterns accurately. CNNs are capable of learning these damage patterns directly from training data and reconstructing the original character structure with better accuracy.

In this research, CNN-based architecture is used as the foundation of the restoration model. The Improved U-Net model is built using convolutional operations that allow the system to learn both global structure and fine character details. This helps preserve important inscription features while improving the readability of damaged characters.

Therefore, Convolutional Neural Networks serve as the core mechanism for feature extraction and reconstruction in the proposed inscription character damage restoration system.

.

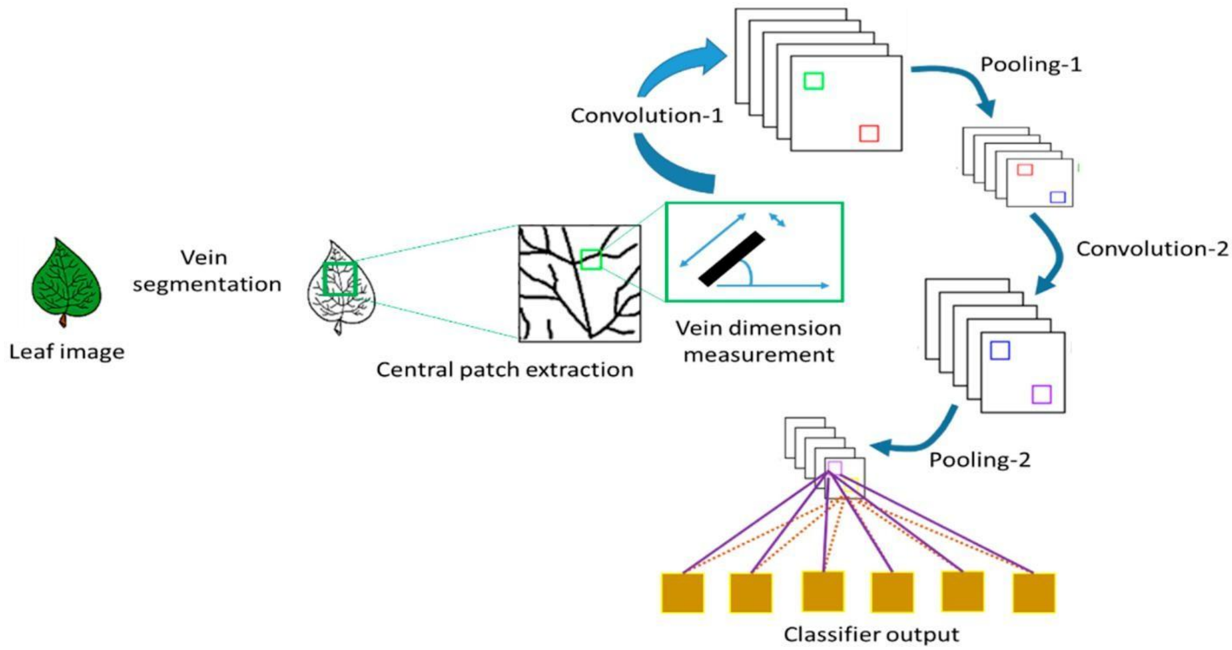

Figure 2.1.2.1: Review on techniques for plant leaves CNN: Source: https://www.google.com/https://www.mdpi.com

2.1.3 U-Net Architecture

U-Net is a specialized Convolutional Neural Network architecture designed for image segmentation and image restoration tasks. It was originally introduced by Ronneberger et al. for biomedical image segmentation and later became widely used in various image reconstruction problems because of its strong ability to preserve fine structural details while performing accurate restoration [1].

The U-Net architecture follows an encoder-decoder structure. The encoder path, also known as the contracting path, captures high-level contextual information from the input image through repeated convolution and pooling operations. The decoder path, also known as the expanding path, reconstructs the output image by gradually restoring spatial resolution using upsampling and convolution operations.

One of the most important features of U-Net is the use of skip connections between corresponding encoder and decoder layers. These skip connections help preserve low-level spatial information such as edges, boundaries, and fine stroke details that may otherwise be lost during downsampling. This is especially important for inscription character restoration because the accurate preservation of character strokes directly affects readability and recognition.

Ancient inscription characters often contain damaged regions where parts of the original structure are missing due to cracks, erosion, and weathering. Restoring such characters requires both global contextual understanding and precise local reconstruction. U-Net is highly suitable for this task because it combines both high-level semantic understanding and low-level detail preservation within a single architecture.

In this research, an Improved U-Net architecture is used for damaged character restoration. The model consists of multiple encoder and decoder stages along with skip connections to reconstruct damaged character images into cleaner outputs. The implementation also includes residual blocks to further improve feature learning and restoration quality .

The final output layer uses a sigmoid activation function to generate restored grayscale images with pixel values normalized between 0 and 1, allowing smooth reconstruction of damaged character surfaces .

Because of its strong restoration capability and structural preservation, U-Net serves as the primary model architecture for the proposed inscription character damage restoration system.

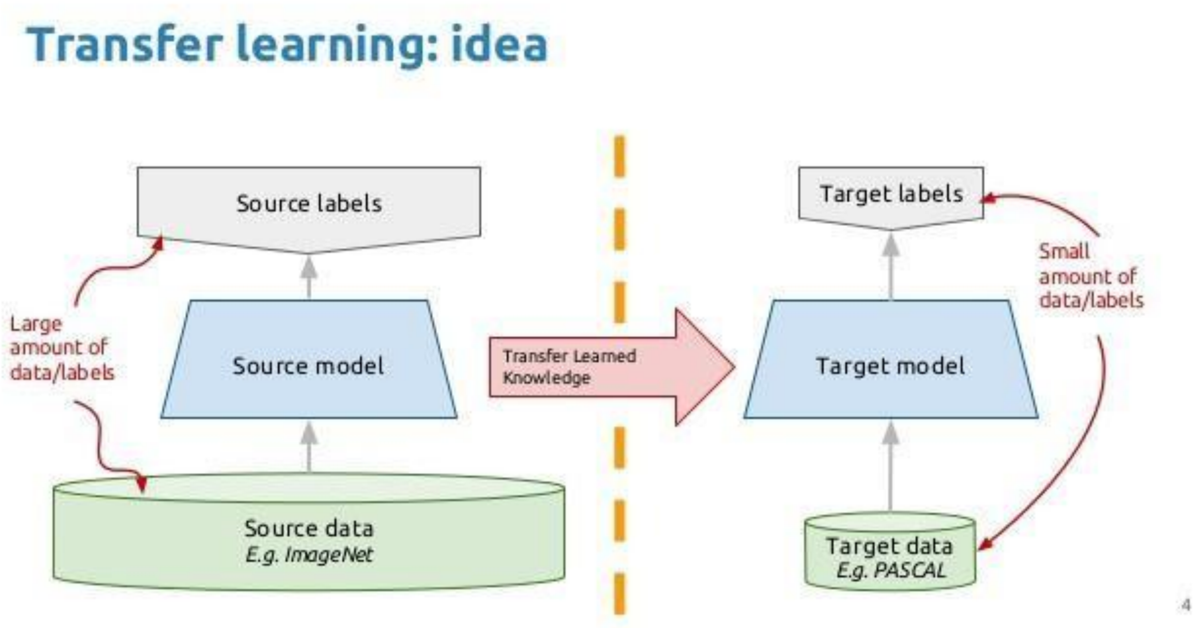

Figure 2.1.3.1: Transfer Learning with Convolutional Neural Networks in PyTorch: Source: https://www.google.com/towardsdatascience.com/transfer-learning-with-convolutional-neuralnetworks-in-pytorch

2.1.4 Residual Learning

As deep neural networks become larger and more complex, training them effectively becomes increasingly difficult due to problems such as vanishing gradients, degradation, and information loss. When deeper layers are added, the model may become harder to optimize, and performance may decrease instead of improving. To solve this problem, He et al. introduced Residual Learning through Residual Networks (ResNet), which significantly improved the training of deep neural networks [2].

Residual learning allows the network to learn the difference between the input and the desired output rather than learning the full transformation directly. This is achieved using shortcut connections, also called skip connections, which pass the input directly to deeper layers. These shortcut paths help preserve important feature information and improve gradient flow during training.

In image restoration tasks, preserving fine structural details is extremely important. Ancient inscription characters contain delicate stroke patterns, thin curves, and small visual features that can easily be lost during multiple convolution operations. Residual learning helps maintain these important details by allowing the original feature information to flow directly through the network.

In this research, the Improved U-Net architecture is built using Residual Blocks in both the encoder and decoder sections of the network. Each residual block contains two convolution layers followed by batch normalization and ReLU activation, along with a skip connection that adds the original input to the processed output before final activation .

This design improves model stability, accelerates convergence during training, and enhances the restoration of damaged inscription characters by preserving stroke-level structural information. The use of residual learning is particularly valuable when reconstructing characters affected by cracks, erosion, and missing regions because it helps the model recover the original visual structure more accurately.

Therefore, residual learning plays a major role in improving the performance of the proposed inscription restoration system and contributes significantly to the final restoration quality.

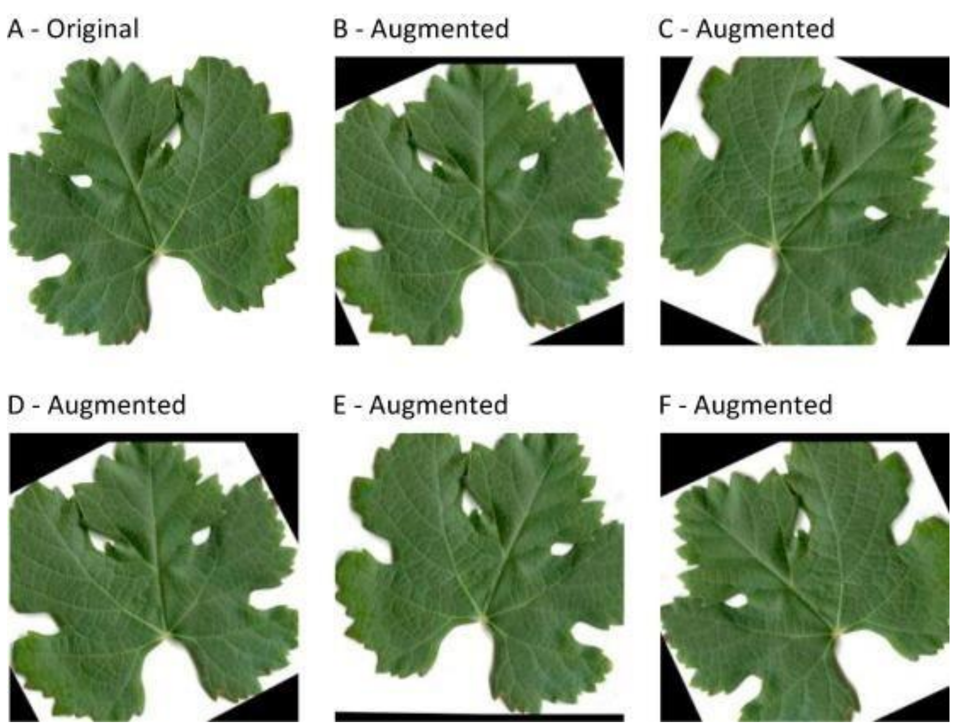

Figure 2.1.4.1: Data Augmentation with leaves:
Source: https://www.google.com/https/www.sciencedirect.com/science/article/pii

2.1.5 Synthetic Damage Generation

One of the major challenges in inscription character restoration is the lack of paired datasets containing both damaged and clean versions of the same character. Ancient inscriptions naturally exist in damaged form due to weathering, erosion, cracks, biological growth, and physical aging over long periods of time. However, obtaining the exact original clean version of these damaged characters is nearly impossible. This creates a major limitation for supervised deep learning models, which require paired input-output samples for effective training.

To overcome this problem, synthetic damage generation is used in this research. Synthetic damage generation is the process of artificially creating realistic damage on clean character images so that the model can learn restoration by comparing damaged inputs with their original clean targets. This allows the creation of a supervised paired dataset without requiring unavailable real clean inscription references.

Several artificial damage techniques are applied to simulate real-world inscription degradation conditions. These include black blobs to represent heavy stone erosion, crack lines to simulate fractures on stone surfaces, eraser patches to imitate missing regions, white patches to represent faded or destroyed character strokes, Gaussian noise to simulate environmental surface noise, and slight blur to represent weathering effects and loss of sharpness .

The synthetic damage process is applied randomly using multiple combinations of these techniques to improve dataset diversity and model generalization. Each clean character image is resized, converted to grayscale, and transformed into a damaged version while preserving the original clean target image for supervised learning. This results in paired training data where the damaged image becomes the model input and the clean image becomes the expected output .

This approach improves the robustness of the model by exposing it to various realistic damage patterns during training. It also allows the restoration system to perform better when handling unknown real damaged inscription characters during inference.

Therefore, synthetic damage generation serves as a critical part of this research by solving the dataset limitation problem and enabling effective deep learning-based restoration of ancient inscription characters.

2.2 Approach

The following methodology illustrates the approach followed to implement the restoration of damaged ancient inscription characters using deep learning. The main objective of this approach is to restore damaged inscription character images into cleaner and more readable forms while preserving their original structural appearance.

The proposed system follows a hybrid restoration strategy. Known inscription characters are identified through database matching using hash-based comparison, while unknown damaged characters are restored using a trained deep learning model based on an Improved U-Net architecture with residual blocks. This combination improves both restoration efficiency and practical usability.

The overall workflow begins with collecting clean inscription character images and preparing a paired dataset for supervised learning. Since real-world damaged-clean paired inscription datasets are not publicly available, synthetic damage generation is applied to create realistic damaged versions of clean characters. These paired images are then used to train the restoration model.

After dataset preparation, the Improved U-Net model is trained using damaged character images as input and clean character images as target outputs. During training, the model learns to reconstruct missing strokes, remove visual noise, and restore degraded character structures. Once the model is trained, it is deployed through an API-based restoration system that supports real-time restoration requests.

The final system provides two major functionalities:

1. Segmentation of full inscription images into individual character-level outputs.
2. Restoration of damaged inscription characters using either direct database retrieval for known characters or model-based restoration for unknown characters.

This approach improves the preservation of ancient inscriptions by reducing manual restoration effort and supporting historians, archaeologists, and researchers in the interpretation of damaged historical records.

2.2.1 Data Collection

In this research, the images of ancient inscription characters act as the primary input for the restoration system. The main objective of the data collection phase is to prepare a clean character dataset that can be used to generate supervised training pairs for damaged character restoration.

The dataset consists of individual inscription character images collected and organized into multiple character folders representing different inscription symbols. Each folder contains clean character samples belonging to a specific character class. These clean images serve as the ground truth targets for model training.

Since naturally paired damaged-clean inscription datasets are not available, the clean character images are first collected and resized into a standard image size for consistency during training. The system uses grayscale character images because inscription restoration focuses mainly on stroke structure and shape rather than color information.

The base dataset is stored in organized character folders inside the dataset directory, where each folder represents one character category. The system supports multiple character classes, and each folder contains a large number of clean character images used for training. To ensure balanced training, a maximum number of images per class is selected during preprocessing .

After collecting the clean dataset, synthetic damage generation is applied to create corresponding damaged versions of the same characters. The generated damaged images are stored separately while preserving the matching relationship with the original clean targets. This creates a paired dataset structure with:

1. Clean images stored in the clean folder
2. Damaged images stored in the damaged folder

The naming convention ensures that each damaged image has a direct matching clean image, allowing supervised learning during training. For example, a damaged image with the suffix "_d" is paired with its corresponding clean image using the suffix "_c" .

This paired dataset becomes the foundation of the restoration model and allows the network to learn realistic reconstruction patterns for damaged ancient inscription characters.

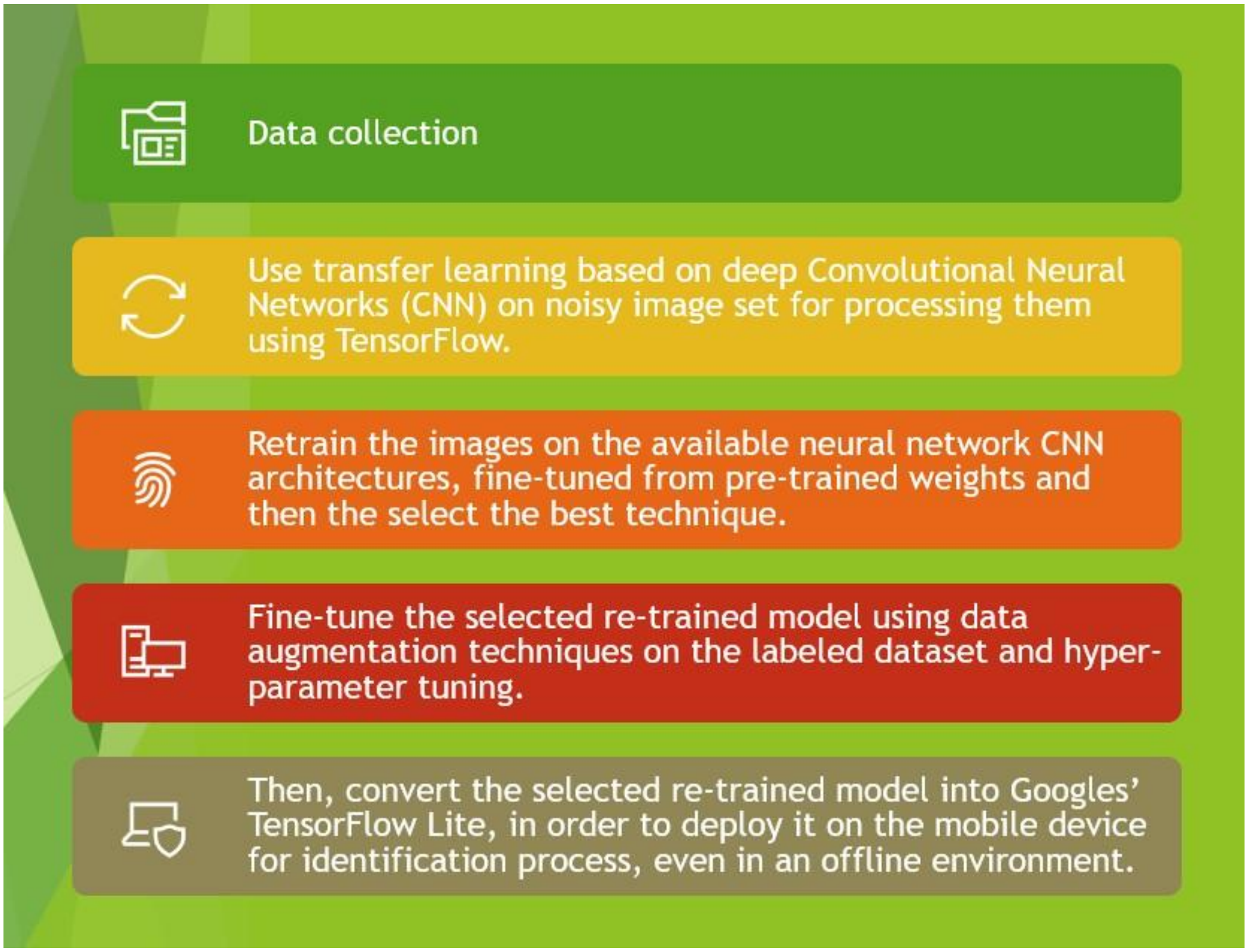

Figure 2.2.1: Methodology of the approach

Figure 2.2.1.1: Akkapana leaf

Figure 2.2.1.2: Cinnamon leaf

Figure 2.2.1.4: Katupila leaf

Figure 2.2.1.3: Katupila leaf

Figure 2.2.1.5: Turmeric leaf

Figure 2.2.1.6: Turmeric root

Figure 2.2.1.7: Kohomba leaf

2.2.2 Methodology

The main objective of this research is to restore damaged ancient inscription characters by learning the relationship between damaged character images and their corresponding clean versions. The proposed methodology focuses on building a supervised deep learning restoration pipeline using paired damaged-clean datasets and an Improved U-Net architecture with residual learning.

The first step of the methodology is dataset preparation. Clean inscription character images are collected and organized according to character classes. Since naturally paired damaged-clean datasets are unavailable, synthetic damage generation is applied to create realistic damaged versions of these clean images. This allows the formation of paired training data where the damaged image serves as the input and the clean image serves as the target output.

The synthetic damage generation process includes multiple realistic degradation techniques such as black blobs, crack lines, eraser patches, white patches, Gaussian noise, and slight blur. These methods simulate common damage conditions found in ancient stone inscriptions, including erosion, fractures, faded strokes, and weathered surfaces. Multiple random combinations of these damage types are applied to improve model generalization and restoration robustness .

After dataset preparation, the paired images are divided into training, validation, and testing datasets. During training, only the training dataset receives augmentation to improve model generalization, while validation and testing sets remain unchanged to ensure reliable evaluation. Random transformations such as horizontal flipping, small rotations, and translations are applied using paired augmentation so that both damaged and clean images remain perfectly aligned during training .

The restoration model uses an Improved U-Net architecture integrated with residual blocks. The encoder path extracts high-level contextual features from damaged input images, while the decoder path reconstructs the restored output image using skip connections to preserve low-level character details. Residual blocks improve feature preservation and model stability by allowing shortcut information flow through the network .

The model is trained using grayscale images resized to a fixed resolution. The input damaged image passes through the network, and the output is compared with the clean target image using restoration loss functions. Through repeated optimization, the model learns to reconstruct missing strokes, reduce surface noise, and restore damaged inscription structures.

After training, the best-performing checkpoint is saved and deployed for inference. During real-time usage, if the uploaded character image matches an existing known character in the database through hash comparison, the system directly returns the stored restored output. If the character is unknown, the trained deep learning model performs restoration using the prediction pipeline implemented in the API system .

This methodology provides both efficient retrieval for known characters and intelligent restoration for unknown damaged characters, making the system practical for real-world inscription preservation tasks.

2.3 Summary of methodology

The proposed methodology for ancient inscription character damage restoration is designed to provide an efficient and reliable solution for restoring damaged historical inscription characters using deep learning. The system combines dataset preparation, synthetic damage generation, supervised model training, and API-based restoration into a complete restoration pipeline.

The process begins with collecting clean inscription character images and organizing them according to character classes. Since real-world paired damaged-clean datasets are not available, synthetic damage generation is applied to create realistic damaged versions of these clean characters. This allows the creation of supervised training pairs where the damaged image is used as input and the clean image is used as the target output.

Several artificial damage techniques such as black blobs, crack lines, white patches, eraser regions, Gaussian noise, and blur are used to simulate real inscription damage patterns caused by erosion, cracks, weathering, and physical aging. This improves the diversity of the training data and helps the model generalize better to unknown damaged characters.

The restoration model is built using an Improved U-Net architecture integrated with residual blocks. The encoder-decoder structure allows the network to capture both global contextual information and fine local stroke details, while residual learning improves model stability and reconstruction quality. The use of skip connections ensures that important inscription features are preserved during restoration.

After training, the model is deployed through an API-based system that supports practical restoration usage. The system first checks whether the input character exists in the known database using hash-based matching. If a match is found, the known restored result is returned directly. If the character is unknown, the trained deep learning model performs restoration through the inference pipeline.

This hybrid approach improves restoration speed, reduces manual effort, and increases restoration consistency while supporting historians, archaeologists, and researchers in preserving valuable ancient inscription knowledge. The methodology provides a scalable foundation for future inscription digitization and large-scale cultural heritage preservation systems.

Research	Research Work	Methods and Algorithms to be used	Expected Accuracy	Remarks

Arogya – An Intelligent Ayurvedic Herb management Platform -> Classification of a group of rare ayurvedic leaves in Sri Lanka	Use transfer learning based on deep Convolutional Neural Networks (CNN) on collected image set for processing them using TensorFlow (The selected retrained model will be finetuned using data augmentation techniques on the labeled dataset and hyperparameter tuning.)	CNN Architectures (Such as Inception v3, ResNet50, VGG16, MobileNet v2, etc.) will be used to choose the best technique	Higher Accuracy will be expected	Several existing algorithms will be analyzed, and accuracy will be tested in order to select the suitable algorithms to classify the selected 5 plants accurately.

Table 2.3.1 : Summary of methodology

2.4 High-Level Architecture Diagram

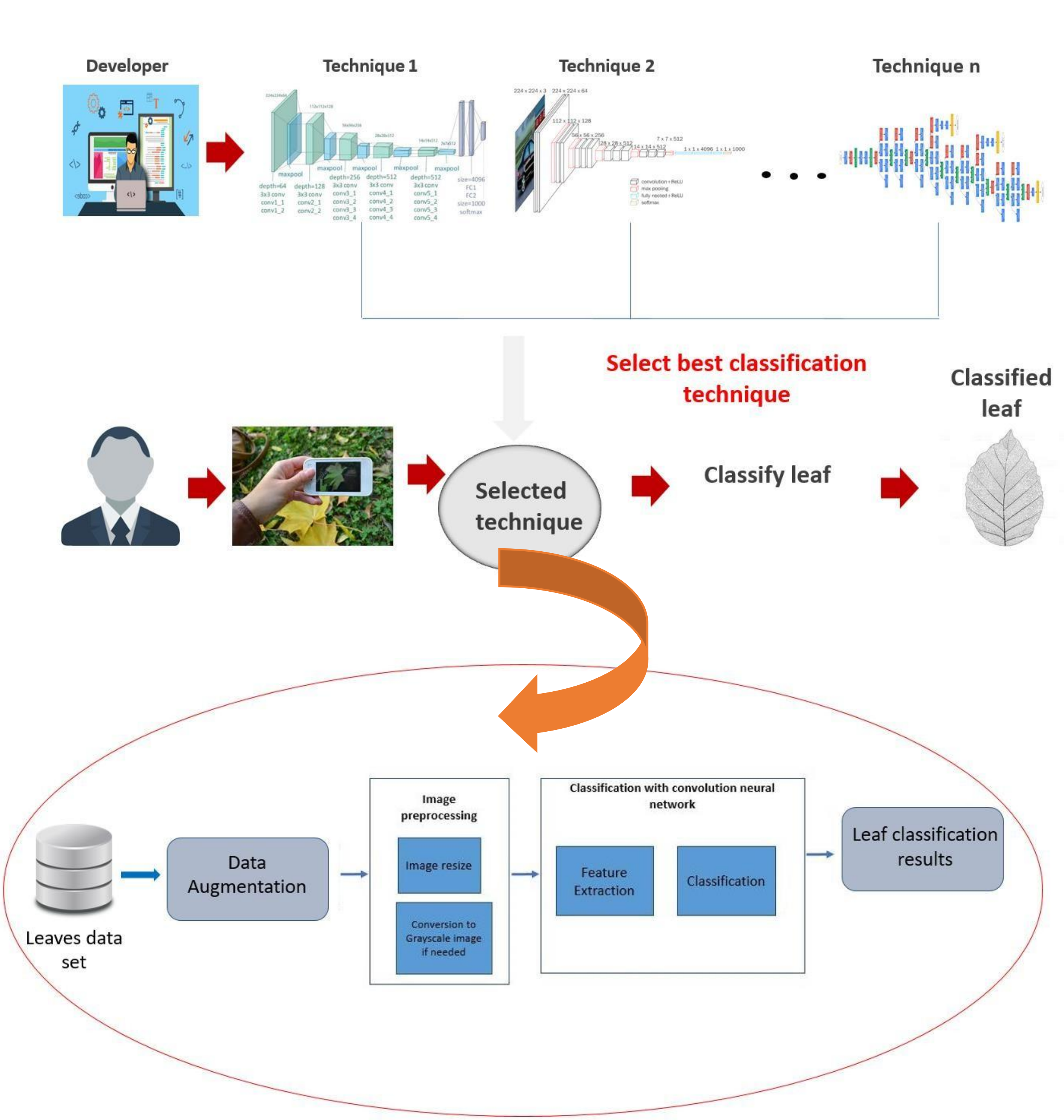

Figure 2.4.1: Software Architectural diagram

2.5 Project Requirements that have been achieved

2.5.1 Functional Requirements

1. **Character Segmentation**

The system should be able to segment full inscription images into individual character-level images before restoration. This allows each damaged character to be processed separately and improves restoration accuracy.

2. **Known Character Identification**

The system should identify whether the uploaded damaged character already exists in the known character database using hash-based image matching. If a match is found, the stored restored result should be returned directly.

3. **Unknown Character Restoration**

If the damaged character is not found in the database, the system should use the trained deep learning model to restore the unknown damaged character using the prediction pipeline.

4. **Synthetic Damage Dataset Generation**

The system should generate realistic damaged versions of clean inscription characters using artificial degradation methods such as black blobs, crack lines, white patches, blur, Gaussian noise, and eraser patches for supervised model training.

5. **Deep Learning-Based Restoration**

The system should use an Improved U-Net architecture with residual blocks to reconstruct damaged characters into cleaner and more readable outputs while preserving structural details.

6. **API-Based Restoration Service**

The system should provide an API-based restoration service that allows users or external systems to upload damaged character images and receive restored outputs in real time.

2.5.2 Non-Functional Requirements

1. Accuracy

The restoration system should produce highly accurate outputs by preserving the original structure and readability of ancient inscription characters.

2. Performance

The system should perform restoration within a reasonable processing time to support practical real-time usage, especially for known character retrieval.

3. Reliability

The system should consistently provide stable restoration results without frequent failures during training, prediction, or API usage.

4. Scalability

The system should support future expansion for larger inscription datasets, additional character classes, and integration with complete inscription digitization systems.

5. Maintainability

The codebase and restoration pipeline should be modular and easy to improve, allowing future enhancements such as OCR integration, translation modules, and larger database support.

6. Usability

The API system should be simple enough for researchers, archaeologists, and developers to use without requiring deep technical knowledge of the restoration model.

2.5.3 Other Requirements

The system should allow future integration with OCR engines, inscription translation systems, historical databases, and large-scale archaeological preservation platforms.

2.6. Consideration of the aspects of the system

Standards: The development of the inscription character damage restoration system followed proper coding standards and research documentation standards throughout the implementation process. Object-oriented programming concepts were used to maintain code quality, modularity, and future maintainability. Important sections of the code were properly commented for readability and easier future improvements. In report writing, citation handling, and referencing, IEEE referencing format was followed consistently as used in the sample report

2.6.1 Social aspects

This mobile application can be used by any person who is not aware about but keen to experience Ayurveda medication worldwide. For example, people who use and wish to use ayurvedic medicines and treatment, researchers in the field of botany, medicine, chemical structure analysis, agriculture, ayurvedic medicinal practitioners, taxonomists, forest department officials, those who are involved in the preparation of ayurvedic medicines and others who are concerned with plant studies, as well as doctors, students, locals, foreigners, Ayurvedic plant sellers and many more can use this application wisely. Therefore, a great appreciation as well as a demand can be expected from the society for this mobile application. Especially, there is no age limitations for the users, no need of advanced computer literacy, as well as no need of advanced knowledge in Ayurveda field to use this app. Only a simple knowledge of how to use a smart mobile phone will be sufficient for this. In addition, the name of the identified plant is given in native as well as in the scientific name of the plant, which would support any person including foreigners. So, this would be very beneficial for the whole society which would do a great service in uplifting our ancient Ayurveda, which is anyway better than modern medicine due to its naturality.

2.6.2 Security aspects

Security of the restoration system is important because inscription datasets, trained models, and restoration results must be protected from unauthorized access and accidental modification. Only authorized users should be allowed to upload images, access restoration outputs, and manage the stored clean character database.

The API-based deployment should include proper authentication and access control mechanisms to ensure secure usage. User requests should be validated before processing to prevent invalid inputs and misuse of system resources.

The trained deep learning model and known character database should be securely stored to avoid corruption or loss of important restoration knowledge. Backup mechanisms and controlled database management improve long-term system reliability.

In future large-scale deployment, cloud hosting platforms with secure user management and controlled permissions can further improve system security and safe collaboration among researchers and institutions.

2.6.3 Ethical aspects

This system does not cause harm, injury, or damage to users or devices. No unethical methods were used during the development of the restoration model, dataset preparation, or deployment process. The system is designed to support human experts rather than replace archaeologists, historians, or inscription specialists. Automated restoration predictions should always be treated as research assistance and not as final historical truth without expert validation.

Since ancient inscriptions are part of cultural and national heritage, ethical responsibility is very important. Incorrect restoration may lead to false historical interpretation. Therefore, the system must be used carefully with proper academic supervision and expert review.

The synthetic damage generation process also follows ethical standards because it is used only for supervised learning purposes and does not alter real historical records. It helps improve restoration accuracy without damaging original inscription sources.

Overall, the system serves as a preservation and research support tool while respecting the historical and cultural importance of ancient inscriptions.

2.6.4 Limitations

- The current system mainly focuses on individual character restoration rather than full inscription sentence-level restoration.
- The dataset size is limited because large publicly available inscription datasets with paired damaged-clean samples are not available.
- Synthetic damage generation may not perfectly represent all real-world inscription damage patterns such as deep erosion, missing stone sections, or highly complex weathering.
- The current system focuses mainly on visual restoration and does not include OCR-based inscription reading or automatic language translation.
- The restoration quality depends heavily on image quality. Extremely blurred, low-resolution, or heavily damaged inputs may reduce prediction accuracy.
- Known character matching works only for characters already stored in the database. Completely unseen characters still rely fully on model prediction.
- Future improvements can include full inscription restoration, OCR integration, language translation, larger inscription databases, and deployment as a complete archaeological preservation platform.

2.7. Commercialization aspects of the product

2.7.1 Target Audience

- Archaeologists involved in ancient inscription discovery and preservation
- Historians and epigraphists who study ancient languages and inscription records
- Universities and research institutions conducting historical and archaeological studies
- Government departments responsible for archaeology, cultural heritage, and national history preservation
- Museums and heritage preservation organizations managing inscription collections
- Students and academic researchers working in archaeology, history, cultural studies, and artificial intelligence research
- Digital preservation projects focusing on historical document restoration and cultural heritage digitization

- Developers building large-scale inscription management, OCR, and historical archive systems

2.7.2 Market Space

- No age limitations for users
- No requirement for advanced technical knowledge to use restoration outputs
- High demand in archaeological and heritage preservation sectors
- Increasing need for digital preservation of historical records and ancient documents
- Strong research demand from universities and academic institutions
- Potential adoption by national museums, libraries, and government preservation projects
- Can be expanded internationally for inscription restoration projects beyond Sri Lanka
- Future integration opportunities with OCR systems, translation systems, and digital archive platforms

2.7.3 Revenue Earning

- Institutional licensing for universities, museums, and archaeology departments
- Subscription-based access for research organizations and cultural heritage institutions
- API service access for external software systems requiring inscription restoration functionality
- Government-funded cultural preservation projects and grants
- Research collaboration funding with universities and international heritage organizations
- Premium large-scale restoration services for museums and national archives
- Integration services for historical database platforms and archaeological digitization systems
- Commercial partnerships with digital heritage preservation platforms

2.8 Testing and Implementation

Testing is an important phase of the proposed inscription character damage restoration system because it ensures that the model performs accurately and reliably under different restoration conditions. The main purpose of testing is to verify whether the system can successfully restore damaged inscription characters while preserving their original structural appearance and readability.

The testing process was carried out at multiple stages including dataset validation, synthetic damage verification, model training validation, known character matching, unknown character restoration, and API-based prediction testing. This multi-stage testing approach helps ensure both technical correctness and practical usability of the final system.

The first stage of testing focused on dataset validation. Clean inscription character images and their corresponding synthetically damaged versions were manually verified to ensure correct pairing and realistic damage simulation. This was important because incorrect damaged-clean pairing would directly reduce model learning quality.

The second stage involved synthetic damage verification. Various artificial damage methods such as black blobs, crack lines, white patches, blur, Gaussian noise, and eraser patches were tested to confirm whether they closely resembled real-world inscription degradation patterns. Multiple combinations were evaluated to improve model robustness and generalization.

The third stage focused on model training validation. During training, validation loss and output quality were continuously monitored to ensure stable learning and prevent overfitting. Restoration quality was evaluated by comparing the restored output images with their original clean target images. The Improved U-Net with residual blocks showed better structural preservation and cleaner restoration compared to simpler baseline methods.

The fourth stage tested known character identification using hash-based retrieval. Previously stored characters were uploaded to verify whether the system could correctly detect known samples and return the stored clean output without requiring model prediction. This improved restoration speed and reduced unnecessary computation.

The fifth stage focused on unknown character restoration. Damaged characters that were not available in the known database were passed through the trained model to evaluate the restoration quality of unseen samples. The system successfully reconstructed missing strokes, reduced noise, and improved readability for most unknown character inputs.

The final stage involved API testing using the deployed restoration pipeline. Image upload handling, prediction response generation, output saving, and restored image delivery were tested to ensure smooth real-time usage. The API successfully supported restoration requests and returned restored outputs for both known and unknown character cases.

Overall, the testing process confirmed that the proposed system performs effectively in restoring damaged inscription characters and provides a reliable foundation for future large-scale inscription preservation systems.

2.8.1 Test Cases

Test Case ID	Test Scenario	Input	Expected Output	Actual Result	Status
TC001	Dataset Pair Validation	Clean and damaged image folders	Each damaged image should correctly match its clean target image	Correct pairing observed using naming convention	Pass
TC002	Synthetic Damage Generation	Clean character image	Realistic damaged image generated with erosion, cracks, blur, and noise	Multiple realistic damage patterns successfully generated	Pass
TC003	Image Preprocessing	Raw character image	Image resized, normalized, and converted to grayscale correctly	Preprocessing completed successfully	Pass
TC004	Model Training Validation	Paired damaged-clean dataset	Model should learn restoration mapping with decreasing validation loss	Stable training observed with improved output quality	Pass
TC005	Known Character Matching	Previously stored known character image	System should detect known character and return stored clean output	Correct known character retrieval achieved	Pass

TC006	Unknown Character Restoration	Unseen damaged character image	System should restore character using trained U-Net model	Missing strokes restored and readability improved	Pass
TC007	API Upload Testing	Uploaded damaged image through API	System should accept image and start prediction process	Upload process completed succes	

2.8.2 Testing

The output validation process was carried out by visually comparing damaged inscription character inputs with the restored outputs generated by the proposed system. Since the main objective of this research is image restoration rather than text classification, visual quality assessment plays a major role in evaluating system performance.

Several output samples were tested using both known and unknown damaged character inputs. For known characters, the system successfully identified the matching character using hash-based retrieval and returned the stored clean output directly. This significantly reduced restoration time and improved efficiency.

For unknown damaged characters, the trained Improved U-Net model with residual blocks successfully reconstructed missing strokes, removed noise, and improved the readability of the characters. The restored outputs preserved the original structural shape while reducing the effects of cracks, erosion, blur, and faded regions.

Synthetic damage generation results were also validated by comparing artificially damaged images with real-world inscription damage patterns. The generated damage types such as black blobs, crack lines, white patches, Gaussian noise, and blur closely resembled actual inscription degradation conditions, improving training realism.

API testing screenshots demonstrated successful image upload, prediction execution, restored image generation, and output saving. The system was able to process restoration requests in real time and return usable restored outputs for further analysis.

Character segmentation results from full inscription images also showed successful extraction of individual character-level images before restoration. This improved the practical usability of the system by allowing restoration at the character level instead of requiring manual cropping.

2.9 Tools and Technologies

- Python
- PyTorch
- OpenCV

- NumPy
- FastAPI
- Torchvision
- PIL
- Hash-based retrieval
- U-Net Model
- Residual Blocks

Research Area – Deep Learning Algorithms (deep neural network architectures based on transfer learning) and Image Processing Techniques

3. RESULTS AND DISCUSSION

3.1 Results

The primary results obtained from transfer learning based on the trained deep CNN architectures can be illustrated as follows. Those results indicate the model summary, final testing accuracy, model performance accuracy against training epochs and model performance loss against training epochs, for each model which have been trained and tested throughout this research.

I. InceptionV3

- Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_v3 (Model)         (None, 1, 1, 2048)        21802784
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dropout (Dropout)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 6)                 12294
=================================================================
Total params: 21,815,078
Trainable params: 12,294
Non-trainable params: 21,802,784
```

Figure 3.1.1: InceptionV3 model summary

- Finalized testing accuracy: **56.76%**

- Model performance accuracy graph

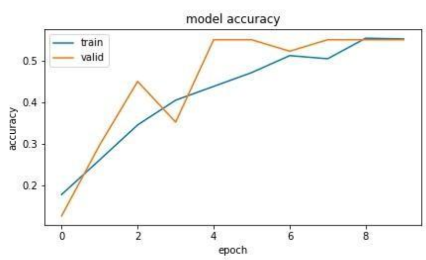

Figure 3.1.2: InceptionV3 accuracy performance

- Model performance loss graph

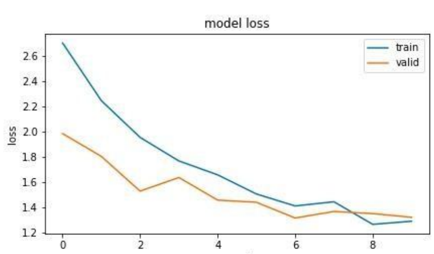

Figure 3.1.3: InceptionV3 loss performance

II. MobileNetV2

- Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenetv2_1.00_224 (Model) (None, 7, 7, 1280)        2257984
_________________________________________________________________
conv2d (Conv2D)              (None, 5, 5, 32)          368672
_________________________________________________________________
dropout (Dropout)            (None, 5, 5, 32)          0
_________________________________________________________________
global_average_pooling2d (Gl (None, 32)                0
_________________________________________________________________
dense (Dense)                (None, 6)                 198
=================================================================
Total params: 2,626,854
Trainable params: 368,870
Non-trainable params: 2,257,984
```

Figure 3.1.4: MobileNetV2 model summary

- Finalized testing accuracy: **63.58%**

- Model performance accuracy graph

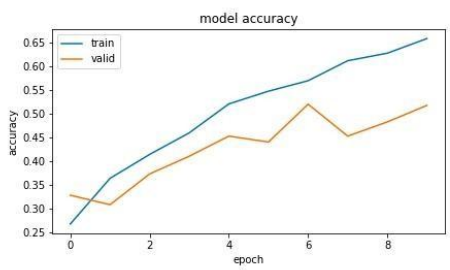

Figure 3.1.5: MobileNetV2 accuracy performance

- Model performance loss graph

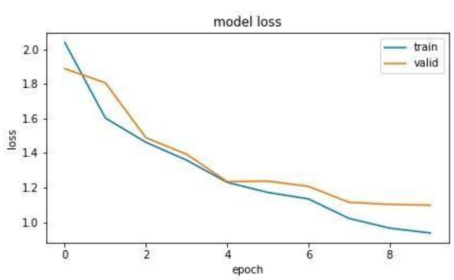

Figure 3.1.6: MobileNetV2 loss performance

III. InceptionResNetV2

- Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_resnet_v2 (Model)  (None, 1, 1, 1536)        54336736
_________________________________________________________________
flatten (Flatten)            (None, 1536)              0
_________________________________________________________________
dropout (Dropout)            (None, 1536)              0
_________________________________________________________________
dense (Dense)                (None, 512)               786944
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_2 (Dense)              (None, 6)                 1542
=================================================================
Total params: 55,256,550
Trainable params: 919,814
Non-trainable params: 54,336,736
```

Figure 3.1.7: InceptionResNetV2 model summary

- Finalized testing accuracy: **82.77%**

- Model performance accuracy graph

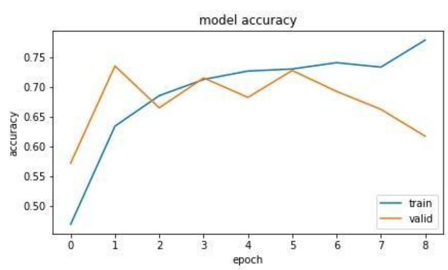

Figure 3.1.8: InceptionResNetV2 accuracy performance

- Model performance loss graph

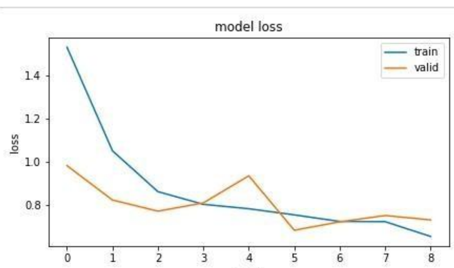

Figure 3.1.9: InceptionResNetV2 loss performance

IV. Xception

- Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
xception (Model)             (None, 3, 3, 2048)        20861480
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 512)               1049088
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 3078
=================================================================
Total params: 21,913,646
Trainable params: 1,052,166
Non-trainable params: 20,861,480
```

Figure 3.1.10: Xception model summary

- Finalized testing accuracy: **86.01%**

- Model performance accuracy graph

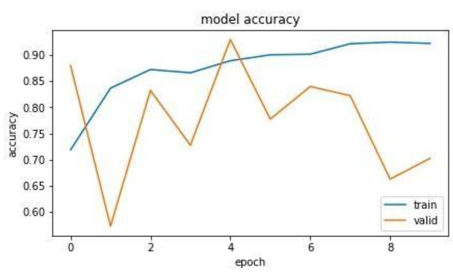

Figure 3.1.11: Xception accuracy performance

- Model performance loss graph

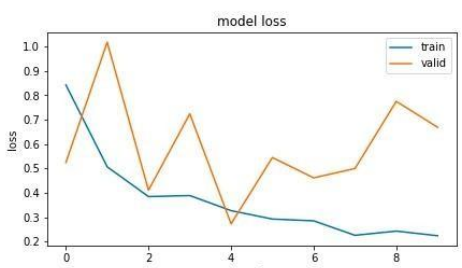

Figure 3.1.12: Xception loss performance

V. DenseNet121

- Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
densenet121 (Model)          (None, 3, 3, 1024)        7037504
_________________________________________________________________
global_average_pooling2d (Gl (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 6)                 6150
=================================================================
Total params: 7,043,654
Trainable params: 6,150
Non-trainable params: 7,037,504
_________________________________________________________________
```

Figure 3.1.13: DenseNet121 model summary

- Finalized testing accuracy: **89.12%**

- Model performance accuracy graph

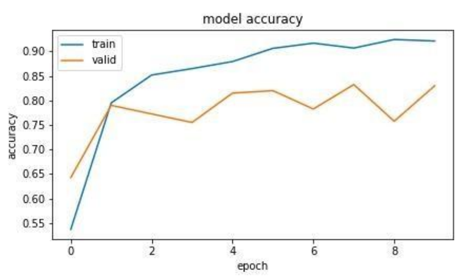

Figure 3.1.14: DenseNet121 accuracy performance

- Model performance loss graph

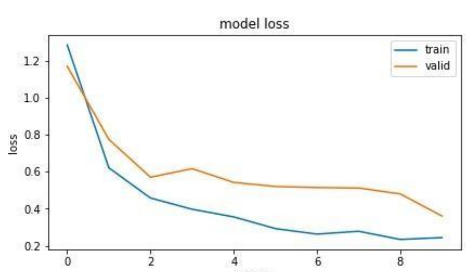

Figure 3.1.15: DenseNet121 loss performance

VI. ResNet50

- Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense (Dense)                (None, 6)                 12294
=================================================================
Total params: 23,600,006
Trainable params: 14,988,294
Non-trainable params: 8,611,712
```

Figure 3.1.16: ResNet50 model summary

- Finalized testing accuracy: **98.92%**

- Model performance accuracy graph

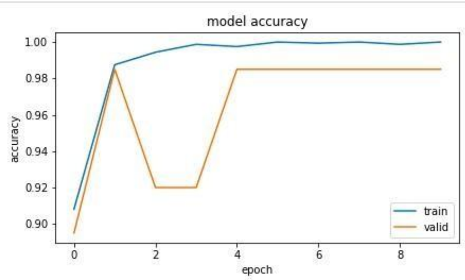

Figure 3.1.17: ResNet50 accuracy performance

- Model performance loss graph

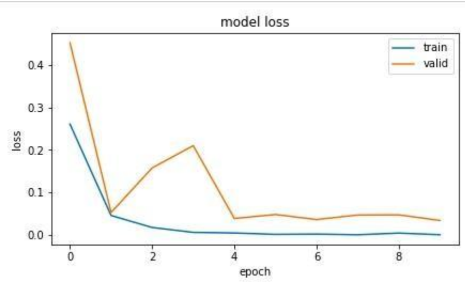

Figure 3.1.18: ResNet50 loss performance

VII. VGG16

- Model Summary

```
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 3, 3, 512)         14714688
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dense (Dense)                (None, 256)               1179904
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 1542
=================================================================
Total params: 15,896,134
Trainable params: 15,896,134
Non-trainable params: 0
```

Figure 3.1.19: VGG16 model summary

- Finalized testing accuracy: **99.53%**

- Model performance accuracy graph

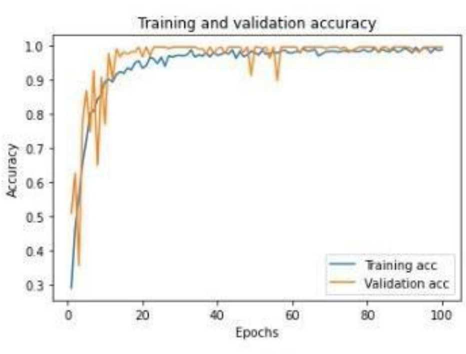

Figure 3.1.20: VGG16 accuracy performance

- Model performance loss graph

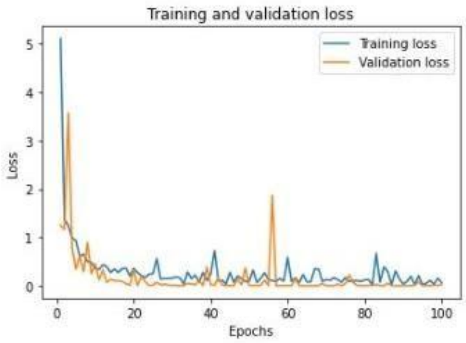

Figure 3.1.21: VGG16 loss performance

3.2 Research Findings and Discussions

The main objective of this individual research component was to analyze several deep CNN models with the acquired training and testing data from scratch, get the final testing accuracy from each in order to compare and achieve the model with the highest accuracy, and then to use it as the finalized model in the herbal plant classification purpose in Arogya, based on images as the input from the mobile camera module.

CNN Architecture	Finalized testing accuracy
InceptionV3	56.76%
MobileNetV2	63.58%
InceptionResNetV2	82..77%
Xception	86.01%
DenseNet121	89.12%
ResNet50	98.92%
VGG16	**99.53%**

Table 3.2.1: Comparison of accuracies for the selected CNN models in herbal plant classification

Hence, according to the comparison, the pre-trained model **VGG16** with a testing accuracy of **99.53%** was chosen as the best model with the highest accuracy. The results were highly promising, reaching over 99% accuracy using the VGG16 model. So, this was the finalized model to be deployed in Arogya mobile application in order to classify the selected Ayurveda leaves/roots/fruits by the user.

The following table illustrates the experimental classification accuracies obtained for each plant class wise when the selected VGG16 model was applied for classification.

Plant Class	**Classified amount**	**Misclassified amount**	**Accuracy**
Akkapana leaf	40	0	100%
Cinnamon leaf	36	4	90%
Katupila leaf with fruit	36	4	90%
Turmeric leaf	38	2	95%
Turmeric digested root	40	0	100%
Kohomba	40	0	100%
Average			**95.83%**

Table 3.2.2: Experimental classification accuracy class wise

So, it is obvious with the experimental results that, the reliability of this research function can guarantee **95% in average**.

There is some research as well as mobile applications already available right now which classify plant leaves and give low prediction accuracies. However, these applications are based on foreign leaves, but not valuable medicinal plants or shrubs. So, Arogya mobile application would be a huge resource for almost everybody who keen to experience Ayurveda in their lives. Not only leaves, but also fruits, flowers and roots associated with Ayurveda will also be able to be classified with this application with a promising accuracy. This would be a great benefit for all the users since in Ayurveda not only leaves are used for preparation of medicine, but also the other parts of the plant are also used often for that purpose. In addition, the development strategy and methodology used in this approach will be able to be used and extended to identify any ayurvedic herb worldwide furthermore.

5.CONCLUSION

The purpose of this research is to develop a centralized social media application which is mobilebased and unique to herbal plants. This solution would be a great chance for those who are keen to learn and use Ayurveda medicine and plants, but who do not have prior knowledge about the specific domain. The application mainly creates awareness among common people about Ayurveda plants, their medicinal usage and value, and about their growth and availability throughout the country.

This proposed system has been tested in various situations and it is capable of providing the most reliable and accurate output to the user. According to the main research components focused, it has been experimentally proved in this research that the most accurate CNN pre-trained model used for classification purpose is VGG-16, which achieved a testing accuracy of 99.53%. The results are highly promising, reaching over 99% accuracy using the VGG-16 model.

This application is currently built only in English. Further, this can be applied in native languages such as Sinhala and Tamil. Since the system is designed only as a mobile application, later can be improvised to a web application with the same functionality and content. Additionally, the development strategy and methodology used in this approach will be able to be used and extended to identify any ayurvedic herb worldwide furthermore, and if the user ends up with doubts and clarifications regarding this procedure, this application can be facilitated with consultation help from Ayurveda doctors.

6. REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, vol. 9351, Springer, 2015, pp. 234–241.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] A. Criminisi, P. Pérez, and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, Sept. 2004.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," in *Proceedings of SIGGRAPH 2000*, New Orleans, LA, USA, 2000, pp. 417–424.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[9] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[10] S. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2003, pp. 958–963.

[11] R. Smith, "An Overview of the Tesseract OCR Engine," in *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633.

[12] M. Diem, F. Kleber, and R. Sablatnig, "Document Image Binarization and Restoration for Historical Manuscripts," *Pattern Recognition and Image Analysis*, vol. 25, no. 3, pp. 418–426, 2015.