Sinhala Ancient Character and Era Classification

Q Overview

This project aims to develop a robust system that recognizes ancient Sinhala characters and classifies them by the historical era they belong to. Due to the low number of images per class and the high similarity among characters, the system uses transfer learning, data augmentation, and ensemble learning techniques to achieve high accuracy with limited data.

Part 1 – Letter Classification (88 Classes)

Approach 🔷

- Feature extraction was done using pretrained deep learning models (VGG19, InceptionV3, ResNet50, InceptionResNetV2).
- Extracted features were classified using ensemble models:
 - Random Forest
 - Extra Trees
 - XGBoost
- Final predictions were made using majority voting across the three models.

Data Characteristics

- Image shape: (224, 224, 3)
- Number of images: ~1097
- Number of character classes: 88

Part 2 - Era Classification (37 datasets)

Approach

- For each dataset (e.g., dataset_0 to dataset_36), a separate MobileNetV2-based model was trained to classify characters based on their era.
- Pretrained MobileNetV2 was used as a lightweight backbone, suitable for fast training on smaller subsets.

To overcome the lack of data and simulate real-world conditions of ancient carvings, the following augmentation techniques were applied:

Augmentation Technique Purpose

Rotation (±15°) To simulate various carving angles

Width & Height Shift (10%) To mimic misalignments or partial cropping

Shear Transformations To reflect natural deformation of stone or inscriptions

Zoom Range (±10%) To simulate variation in distance

Brightness Adjustment To deal with different lighting or photo conditions

Horizontal Flip Applied selectively (if class-invariant)

Fill Mode (Nearest) To maintain smooth visual transitions in gaps

This improved generalization and helped reduce overfitting.

Model Architecture



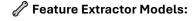
1. Letter Classification Model Architecture



Classify 1097 images into 88 ancient Sinhala character classes.

Architecture Strategy:

Feature Extraction (Transfer Learning) + Classical Ensemble Models



You used ImageNet-pretrained CNNs to extract deep visual features from input images:

Model Name Role in Pipeline Details

VGG19 Feature Extractor 19-layer deep CNN, strong with textures

InceptionV3 Feature Extractor Multi-scale receptive fields

ResNet50 Feature Extractor Skip connections to learn residuals

InceptionResNetV2 Feature Extractor Combines Inception and ResNet styles

Each model was used with:

include_top=False → removes final classification layer.

- pooling='avg' → global average pooling applied to output tensor.
- **Frozen weights** → no fine-tuning, used only for feature extraction.

Feature Vector Output:

Each model outputs a 1D feature vector (e.g., 2048 dimensions for ResNet50).

Classifier Architecture (Ensemble Layer):

Once feature vectors are obtained, they are fed into three different classical classifiers:

Classifier Description

RandomForestClassifier Ensemble of decision trees with bootstrapping

ExtraTreesClassifier Like Random Forest but more randomized

XGBoostClassifier Gradient-boosted decision trees, fast and accurate

Ensemble Output:

- Each model independently predicts the class.
- Final prediction: Based on majority voting among three models.

2. Era Classification Model Architecture

Q Goal:

Train 37 separate models, one per dataset/folder, to identify the era of the character (based on carving style, material, etc.).

Architecture Strategy:

Fine-tuned Convolutional Neural Network (CNN) using MobileNetV2

Input Shape:

(224, 224, 3) RGB image format

Model Design:

Layer Description

MobileNetV2 (Frozen) Pretrained on ImageNet, includes depthwise separable convolutions for speed

GlobalAveragePooling2D() Reduces feature map to a single 1D vector

Layer Description

Dropout(0.5) Regularization to prevent overfitting

Dense(128, ReLU) Fully connected layer to learn compact representation

Dropout(0.3) Additional regularization

Dense(N, Softmax) Output layer with N classes (depends on dataset)

Training Details:

• **Optimizer**: Adam (lr=0.001)

• Loss: Categorical Crossentropy (for multi-class classification)

• Metric: Accuracy

Epochs: 50

Batch Size: 8

■ Model Summary Example (MobileNetV2-based):

Layer Type Output Shape Param #

Input Layer (224, 224, 3) 0

MobileNetV2 (7, 7, 1280) 2.26M

GlobalAvgPooling2D (1280) 0

Dropout(0.5) (1280) 0

Dense (128) (128) ~163K

Dropout(0.3) (128) 0

Dense (N Classes) (N) varies

Total Parameters: ~2.4M

Trainable Parameters: ~180K (only top layers)

Q&A Section

?Q1: Why use transfer learning (pretrained models)?

Because your dataset has **very few images per class**, training a CNN from scratch would overfit. Pretrained models (on ImageNet) already know how to extract generic image features like edges, textures, and shapes—this helps even with Sinhala characters.

Q2: Why not use lightweight models like MobileNet for letter classification?

Letter classification involves 88 very similar characters, which require deep and complex feature extraction. Lightweight models might miss small differences, so you used deeper models like VGG19 and Inception for better accuracy.

?Q3: Why use ensemble classifiers?

- ✓ Each classifier (Random Forest, Extra Trees, XGBoost) has different strengths:
 - Random Forest: good with variance
 - Extra Trees: better speed
 - XGBoost: handles imbalanced data well

Combining them with majority voting improves overall stability and accuracy.

?Q4: Why train a separate model for each era?

The appearance of characters can **change between eras** due to carving style, erosion, or medium (stone, wood, etc.). Training **one model per era** allows better specialization and avoids confusion across styles.

Q5: Why choose MobileNetV2 for era classification?

- ✓ MobileNetV2 is:
 - Lightweight and fast
 - Suitable for small and specialized datasets
 - Offers good accuracy with fewer parameters

Perfect for training 37 separate models without requiring large GPU resources.

?Q6: How was overfitting handled?

✓ Overfitting was addressed through:

- Data Augmentation
- Transfer learning with frozen layers
- **Dropout layers** in the classifier head
- Train-validation split with stratification

?Q7: What challenges were there?

- ✓ Challenges included:
 - Very few samples per class
 - Class imbalance
 - Very similar characters (difficult to separate)
 - Style variations across different eras
 - Need to handle large number of small datasets (37 total)

All were addressed with augmentation, model separation, and ensemble strategies.

?Q8: How are the results stored and used later?

- For each dataset:
 - Trained model saved as .h5
 - Label encoders saved using joblib
 - This makes future loading and inference easy

?Q9: What is the final result?

- ✓ You successfully built:
 - A letter recognizer with ensemble classification from pretrained CNN features
 - An era classification pipeline using MobileNetV2 for 37 ancient datasets
 - A robust pipeline that performs well despite limited data and high visual similarity

Summary

Task	Model	Features	Output
Sinhala Character Classification	VGG19, InceptionV3, ResNet50, InceptionResNetV2 → RF, ET, XGB	Transfer learning features	88 classes
Era Classification	MobileNetV2	Fine-tuned per dataset	1 model per dataset (37 total)

Deliverables (in your notebook)

- Letter classification via ensemble learning
- Z Era classification via 37 MobileNetV2 models
- Z Data augmentation for every image
- Saved models and encoders for later use