Custom Image Classification Dataset and Model Development for Five Animal Classes

CSE445 - Machine Learning
Section 5
Project Group No. 1

Raisul Islam Kabbo (2222542042), Md. Rafawat Islam (2122343642),
Shakil Ahmed (2221453042), Entishar Rashid Chowdhury (2222145042),
Sudipto Roy (2222756042)

***Abstract*—This project aims to build a custom image classification dataset with five animal classes—dog, cow, cat, lamb, and zebra—each containing 100 images, sourced from the internet and mobile phones. Our goal is to train classical machine learning models to achieve at least 90% accuracy in classifying these images. We avoid deep learning approaches as per the course requirements. So far, we have experimented with multiple models, feature engineering techniques, and preprocessing steps to assess baseline performance.**

I. Dataset Overview

- **Classes:** Dog, Cow, Cat, Lamb, Zebra
- **Images per Class:** 100
- **Sources:** Internet and mobile phone captures
- **Preprocessing:** Images resized to 64×64 or 128×128, converted to grayscale or RGB as required

II. Methodology and Experiments

A. K-Nearest Neighbors (KNN) – Entishar Rashid Chowdhury

We used the K-Nearest Neighbors (KNN) model due to its simplicity and effectiveness on small, multi-class datasets. KNN classifies samples based on the majority class of their nearest neighbors, requiring minimal training.

In our first attempt, we used grayscale pixel values from 64×64 images, achieving 36.45% accuracy—limited by noisy and uninformative raw features. Next, we used normalized RGB values with PCA, improving accuracy to 42.99%, showing the benefit of color and dimensionality reduction. In the third attempt, color histograms were used, and after scaling and PCA, accuracy rose to 45.79%, indicating better feature representation.

However, accuracy remains low due to KNN's dependence on input feature quality. Raw pixels and color histograms may miss subtle visual patterns, and KNN is sensitive to scaling and high dimensionality.

Future improvements include combining color histograms with texture or edge features, fine-tuning PCA, using advanced normalization, experimenting with different distance metrics, or using ensemble KNN methods.

B. Support Vector Machine (SVM) – Raisul Islam Kabbo

We used the Support Vector Machine (SVM) model to perform image classification. The SVM model is chosen for its effectiveness in high-dimensional spaces and suitability for binary and multi-class classification tasks.

At first, we loaded images from a folder structure based on class labels. Resized each image to 128×128 pixels and converted each image to grayscale. Then we combined the HOG features and labels into arrays X and y. Split the dataset into training and testing sets and trained an SVM classifier. We got an accuracy of 54% from this model.

Although we got poor accuracy due to poor image quality or inconsistent resolution, insufficient number of samples per class, small dataset size or overly simple model.

We can improve these issues by increasing dataset size or use data augmentation, ensuring images are clean and consistently labeled, trying different models.

C. Decision Tree – Sudipto Roy

We used the Decision Tree classifier because of its simplicity, interpretability, and ability to model non-linear decision boundaries without requiring feature scaling. Decision Trees recursively split the feature space based on feature thresholds.

In our first attempt, we used grayscale pixel values from resized 64×64 images, resulting in 4096 features per image. These raw pixel values were directly fed into the model without additional preprocessing or feature engineering. The model achieved an accuracy of 31.00%.

This low performance is likely due to the noisy and unstructured nature of raw pixel features, which do not capture spatial relationships or patterns in the image. Additionally, Decision Trees tend to overfit high-dimensional data.

Future improvements include:

- Using more meaningful features like color histograms, HOG, or texture descriptors
- Reducing dimensionality with PCA
- Fine-tuning tree hyperparameters such as max_depth and min_samples_split

D. *Random Forest – Md. Rafawat Islam*

We used the Random Forest model for image classification because of its ability to reduce overfitting seen in single decision trees. This model builds multiple decision trees and averages them by voting.

We extracted input features from five animal categories using HOG from grayscale images resized to 128×128. After training on 80% of the dataset, the model achieved an accuracy of 58.13%.

However, the accuracy remains low due to insufficient number of samples, small dataset size, and limited feature discrimination in some classes.

Improvements include dataset expansion, better class balance, image cleaning, and hyperparameter tuning.

E. *Logistic Regression – Shakil Ahmed*

We used logistic regression in our image classification project because it is a simple, interpretable, and widely used model for multiclass classification tasks. Logistic regression allowed us to understand how well a linear classifier performs on visual data when deep learning is not allowed.

To apply the model, we first extracted HOG features from the images of five animal classes—dog, cow, cat, lamb, and zebra—then scaled the features and trained a logistic regression model using a one-vs-rest approach. However, the model achieved only about 35% accuracy on the test set.

This low performance is likely due to logistic regression being a linear model, which struggles to capture the complex and non-linear patterns often found in image data.

The problem could be addressed by enhancing our feature extraction—such as combining HOG with color histograms or texture-based features like LBP—and by performing hyperparameter tuning. Additionally, removing noisy or low-quality images from the dataset could help the model learn more effectively.

III. Summary Table

Model	**Feature Type**	**Accuracy (%)**
KNN	Color Histogram + PCA	45.79
SVM	HOG	54.00
Decision Tree	Grayscale Pixels	31.00
Random Forest	HOG	58.13
Logistic Regression	HOG	35.00

IV. Conclusion and Next Steps

Although classical models provided moderate accuracy, none met the 90% target. Performance was limited by small dataset size, feature quality, and class similarity.

Next Steps:

- Increase and balance the dataset
- Enhance feature extraction (e.g., combining HOG, color histograms, texture)
- Apply data augmentation and noise reduction
- Tune hyperparameters and explore ensemble models
- Consider lightweight deep models if permitted