Machine Learning
Image Classification: From Feature Extraction to Model Evaluation

Vikas Thammanna Gowda

04/03/2025

Context

Machine Learning (ML) and Deep Learning (DL) are rapidly transforming the field of computer vision. Traditionally, students are accustomed to dealing with tabular data (e.g., CSV files with rows and columns). However, images—just like structured data—can be transformed, processed, and leveraged for classification tasks. In computer vision, an image's essential characteristics can be extracted using filters or feature extraction techniques, converting each image into a numerical representation. Once these features are extracted, they can be treated like typical tabular data, enabling the use of both traditional ML algorithms and more advanced Neural Networks.

This project will walk you through the full cycle of an image-based classification workflow:

1. Acquiring a dataset.
2. Pre-processing and extracting features using at least three different filters.
3. Applying a set of classification algorithms, including at least one Neural Network, to predict the labels of unseen images.
4. Evaluating model performance with standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

Objectives

1. **Explore ML/DL for images:** Students will learn that standard classification algorithms (e.g., SVM, Decision Trees, Neural Networks) can work on image data once appropriate features are extracted.
2. **Understand the feature extraction concept:** By experimenting with multiple filters (e.g., edge detection, texture-based features, color histograms), students will gain insight into how images can be numerically described.
3. **Reinforce classification evaluation:** Students will see that evaluation metrics used for tabular data (accuracy, precision, recall, F1-score, confusion matrix) apply directly to image classification.
4. **Practice parameter tuning:** Students will learn how adjusting parameters and arguments in their chosen models (e.g., number of hidden layers in a neural network, the C-parameter in SVM, or max depth in a Decision Tree) impacts performance.

Learning Outcomes

Upon successful completion of this project, students will be able to:

1. **Explain the fundamentals of image representation:** Describe how images are converted into numerical features for ML/DL algorithms.
2. **Implement image pre-processing and feature extraction:** Utilize at least three different types of image filters or transformations to derive features.
3. **Apply multiple classification algorithms, including a Neural Network:** Train and evaluate at least four classifiers (three classic ML algorithms plus one NN).
4. **Evaluate and interpret model performance:** Use classification metrics (accuracy, precision, recall, F1-score) and confusion matrices to interpret results and compare model performance across different feature sets.
5. **Document and communicate findings:** Present the project's methodology, experimental setup, and results in a clear, structured manner.

Requirements

1. Dataset Selection

1. **Choose an image dataset** with:
   - At least **2000 images** in total.
   - **At least three different classes** (e.g., "cats vs. dogs vs. birds," "car models," "flowers," etc.).
   - Publicly available datasets (Kaggle, academic repositories) are acceptable, or you can source your own images if you have permission and resources.
2. **Data organization**:
   - Ensure images are sorted or labeled so you can easily distinguish classes.
   - Most image datasets contain separate folders for training and testing data. If not, split your dataset into training and testing sets.

2. Pre-processing

1. **Image resizing or scaling**:
   - Decide on a standard dimension or resolution for all images (e.g., 128×128 pixels) to keep data consistent.
2. **Normalization or standardization**:
   - Consider normalizing pixel values (e.g., scaling from [0, 255] to [0, 1]) or standardizing them (subtract mean, divide by standard deviation).
3. **Filtering and noise reduction** (optional):
   - Remove artifacts or noise if the dataset images are noisy.

3. Feature Extraction

Use at least three different feature extraction techniques. Some suggested approaches:

1. **Edge-based features**:
    - Apply filters like Canny, Sobel, or Laplacian to capture edge information.
2. **Texture-based features**:
    - Use techniques like Local Binary Patterns (LBP) or Gray Level Co-occurrence Matrix (GLCM).
3. **Color histogram**:
    - Compute color histograms (RGB, HSV, etc.) to capture color distribution.
4. **Global image descriptors**:
    - For example, Histograms of Oriented Gradients (HOG) for shape-based features.

Each extraction method will produce a different numerical feature vector for each image.

4. Classification Algorithms

You must apply **four classification algorithms** to the extracted features. These must include:

1. **Neural Network**:
    - At least a simple feedforward network (e.g., multi-layer perceptron) with configurable hyperparameters (hidden layers, number of neurons, learning rate, etc.).
2. **Three additional ML algorithms**:
    - Possible choices: SVM, Decision Tree, Random Forest, Logistic Regression, k-NN, Naive Bayes, etc.

Implementation tip: For each of the three different feature sets (produced by the different filters), you should train and evaluate all four algorithms.

5. Parameter Tuning (Hyperparameter Optimization)

- Adjust hyperparameters for each model and document how changes affect performance.
- Example hyperparameters to tune:
    - **Neural Network**: Number of hidden layers, number of neurons, learning rate, batch size, epochs.
    - **SVM**: Kernel type, regularization parameter C, gamma.
    - **Decision Tree**: Max depth, min samples split, criterion (Gini vs. entropy).
    - **Random Forest**: Number of estimators, max depth, max features, etc.

6. Evaluation Metrics

For **each experiment**, i.e., each feature extraction method × each classifier (algorithm + hyperparameter), compute the following metrics on the test set:

1. **Accuracy**
2. **Precision** (macro or per-class)
3. **Recall** (macro or per-class)
4. **F1-score** (macro or per-class)
5. **Confusion matrix** (visualize or present numerically)

Present your findings (during presentation) with concise tables, charts, or plots comparing the performance across different combinations of feature extraction and classification models.

Submission Details

Group submission (35% + 35%) Due 04/23/2025:

- Source code with dataset in a compressed format.
- Presentation slides:
  - A 15 minute presentation + 2-3 minutes for Q&A on 04/24/2025.
  - Explain the scope of the project, summarize the project findings including performance comparisons.
  - **Note:** Points will be deducted if the presentation does not last 15 minutes. For example, a 10-minute presentation will result in a 33% deduction.

Individual submission (30%) Due 04/27/2025: Answer the following with minimum 600 words (.pdf)

- Your contributions.
- How did the coursework help you with the project?
- What additional knowledge did you need to gain to complete this project? Explain.
- Your take on traditional data classification and image classification.
- Your experience interacting with AI to assist with the code generation.

Academic Integrity Policy

- You are allowed to use AI to assist you with the coding.
- The use of AI tools (e.g., ChatGPT, Bard, etc.) to generate or assist in writing the individual contributions is strictly prohibited. Any detected use of AI tools for individual contributions generation will result in an automatic **F grade** for the project.
- Proper citations and references must be provided for any external sources consulted. Plagiarism will not be tolerated and will be subject to academic penalties.