Automatic License Plate Recognition Using Decision Tree and SVM

Hossein Arianmehr Pourya Alvani

April 2025

1 Introduction

The objective of this project is to develop an automatic vehicle license plate recognition system using machine learning techniques. The system is divided into multiple phases, each responsible for a part of the recognition pipeline: extracting license plates from full vehicle images, segmenting the characters from the plate, training machine learning models (Decision Tree and SVM), and finally evaluating their performance on test data.

Two datasets are used:

- A labeled dataset for training, containing separate images of Persian digits and letters from 0 to 9 and 'Alef' to 'Ya'.
- A test dataset consisting of real-world car images with plates that need to be detected and processed.

2 Phase 1: License Plate Extraction

In the first phase, the goal is to detect and extract the license plate region from each car image using annotation files. Each image is accompanied by an XML annotation file that includes the bounding box coordinates for the vehicle plate.

The annotations are based on a smaller reference image size (typically 224×224), whereas the original images have a higher resolution (1280×1280). Therefore, proper scaling is necessary to map annotation coordinates accurately onto the high-resolution images.

The process is summarized as follows:

- Set up directory paths for annotations and vehicle images.
- Loop through all annotation files and load their corresponding high-resolution images.
- Read the width and height from the annotation and calculate scaling factors based on the actual image dimensions.
- For each bounding box labeled as "vehicle plate," apply the scale and crop the region from the original image.

- Save the cropped plate image in a separate output directory for use in later phases.

At the end of this phase, a collection of plate-only images is prepared. These images serve as input for character segmentation and recognition in the subsequent stages.

3 Phase 2: Character Segmentation

In the second phase, we focus on isolating individual characters from each extracted license plate image. This step is crucial to prepare clean character images that can later be used for training and testing recognition models.

The main operations in this phase include:

- **Deskewing the Plate Image:** To correct tilted plates, Hough line detection is used to estimate the skew angle and apply an affine transformation to straighten the image.
- **Preprocessing:** Each plate image is converted to grayscale and passed through a Gaussian blur to reduce noise. Adaptive thresholding using Otsu's method is applied to obtain a binary image. Morphological operations (closing and dilation) are performed to enhance character boundaries.
- **Connected Components Analysis:** Both the thresholded and dilated images are used to detect connected components. These components are filtered based on properties like area, height, and width to isolate potential character regions.
- **Character Extraction and Normalization:** Each valid component is cropped, inverted (black on white), resized to a fixed size (23×23), and centered on a 28×28 canvas. These images are saved as individual character files in separate folders for both dilated and thresholded versions.
- **Composite Image Creation:** For visual inspection and debugging, a composite image is generated for each plate. It includes the intermediate processing steps (gray, blurred, thresholded, dilated) and the extracted characters, arranged in two rows.

At the end of this phase, two sets of character images are generated per plate (based on thresholded and dilated sources), along with a composite image for quick verification. These character images are then ready to be used for model training in the next phase.

4 Phase 3: Model Training for Character Recognition

In this phase, we focus on building and training machine learning models to recognize individual Persian characters extracted from license plates. The models used in this phase are Support Vector Machine (SVM) and Decision Tree classifiers.

The steps are summarized as follows:

Data Preparation

All labeled character images are first preprocessed by cropping white borders and resizing them to a uniform size of 28×28 pixels. The dataset is organized into folders per character class, and each image is converted to grayscale.

Data Augmentation

To increase the robustness of the models and reduce overfitting, each image is augmented using three techniques:

- **Random Affine Transformations:** Slight rotations, translations, and scaling.
- **Elastic Transformations:** To mimic natural handwriting variations.
- **Additive Gaussian Noise:** To simulate variations in image quality.

Each augmented image retains the same label as its original.

Feature Extraction and Dimensionality Reduction

Histogram of Oriented Gradients (HOG) features are extracted from both original and augmented images. These features capture edge and shape information which is useful for character classification.

The feature vectors are then:

- Normalized using `StandardScaler`.
- Reduced in dimension using `Principal Component Analysis (PCA)` with 50 components, improving efficiency and reducing noise.

Model Training and Evaluation

Two classification models are trained and evaluated:

- **Support Vector Machine (SVM):** A linear kernel is used. The model is evaluated using 5-fold cross-validation, and trained on the PCA-reduced feature vectors.
- **Decision Tree Classifier:** A basic decision tree model is trained similarly and evaluated with 5-fold cross-validation.

Cross-validation results showed that both models performed well, with SVM generally achieving higher and more stable accuracy than the decision tree. The trained models are used in the next phase to recognize characters extracted from actual license plates.

5 Phase 4: License Plate Character Prediction

In this phase, the trained models from the previous step are used to predict the characters of full license plates. These predictions are based on individual character images extracted from thresholded plates in Phase 2.

Prediction Process

The prediction workflow consists of the following steps:

- Each character image is resized to 28×28 pixels and converted to grayscale.
- HOG features are extracted from the image and passed through the same scaler and PCA transformation used during training.
- Both the SVM and Decision Tree models are used to predict the character class for each image.
- The predicted labels are recorded for each character and grouped by plate ID.

Output and Storage

Each predicted character image is saved with a filename indicating its corresponding plate ID and both model predictions (e.g., `SVM-two_DT-four`). This allows for visual comparison of results from the two models.

Two summary files are generated:

- `svm_predictions.txt` – lists plate IDs and their predicted characters by SVM.
- `tree_predictions.txt` – lists plate IDs and their predicted characters by the Decision Tree model.

Result Visualization

For better visualization and evaluation, the system also constructs composite images that merge all predicted characters for each plate into a single image. Each character tile includes the predicted label from both models and the plate ID for reference. These combined images are stored in a designated folder for manual inspection and comparison.

This phase provides a full pipeline from raw character extraction to readable license plate output, forming the basis for final model evaluation in the next phase.

6 Phase 5: Model Evaluation

In the final phase, we evaluate the performance of the trained models (SVM and Decision Tree) using multiple metrics. The evaluation focuses on both individual character recognition accuracy and full license plate recognition accuracy.

Evaluation Metrics

The following metrics are used:

- **Character-level Accuracy:** Measures the percentage of correctly predicted individual characters.
- **Plate-level Accuracy:** A stricter metric that counts a plate as correctly recognized only if all its characters are correctly predicted.

- **Precision & Recall (Macro-averaged):** Evaluates the model's performance across all classes, treating each class equally.
- **Confusion Matrix:** A detailed visual representation showing which characters were misclassified as others.

Support Vector Machine (SVM)

- Plate-level Accuracy: **0.19**
- Character-level Accuracy: **0.72**
- Precision (macro): **0.39**
- Recall (macro): **0.44**

The SVM model showed moderate performance at the character level, but low performance when evaluated on complete license plates. The low precision and recall indicate that while some characters are predicted well, others are frequently confused or missed.

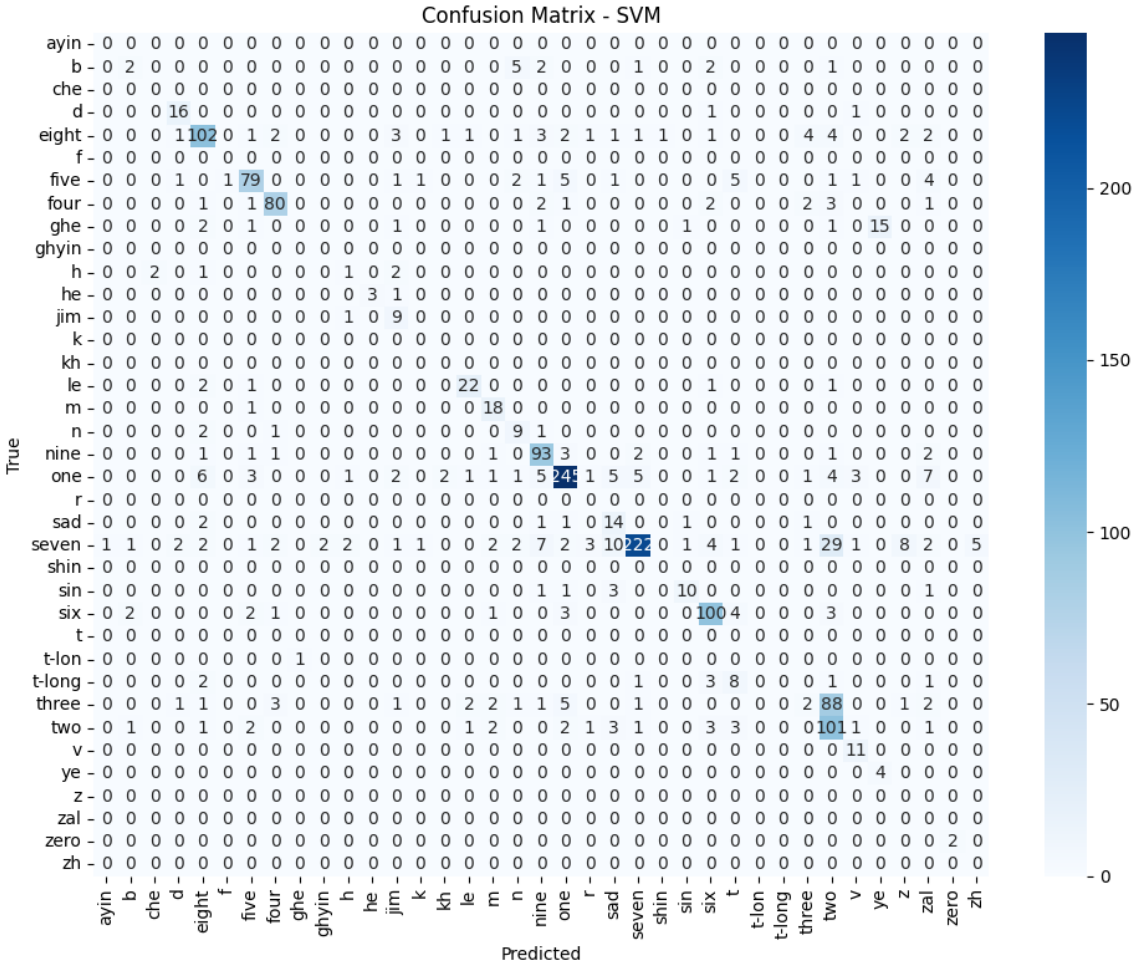

Figure 1: Confusion Matrix for SVM Model

Analysis: Most errors occurred in characters with similar shapes such as "one" and "seven" or "three" and "eight". However, overall prediction distribution remains consistent and centered on the diagonal, indicating high accuracy.

Decision Tree Classifier

- Plate-level Accuracy: **0.01**
- Character-level Accuracy: **0.56**
- Precision (macro): **0.28**
- Recall (macro): **0.30**

The Decision Tree model significantly underperformed in both plate-level and character-level evaluations. Its low scores suggest difficulty in handling the complexity of Persian character classification.

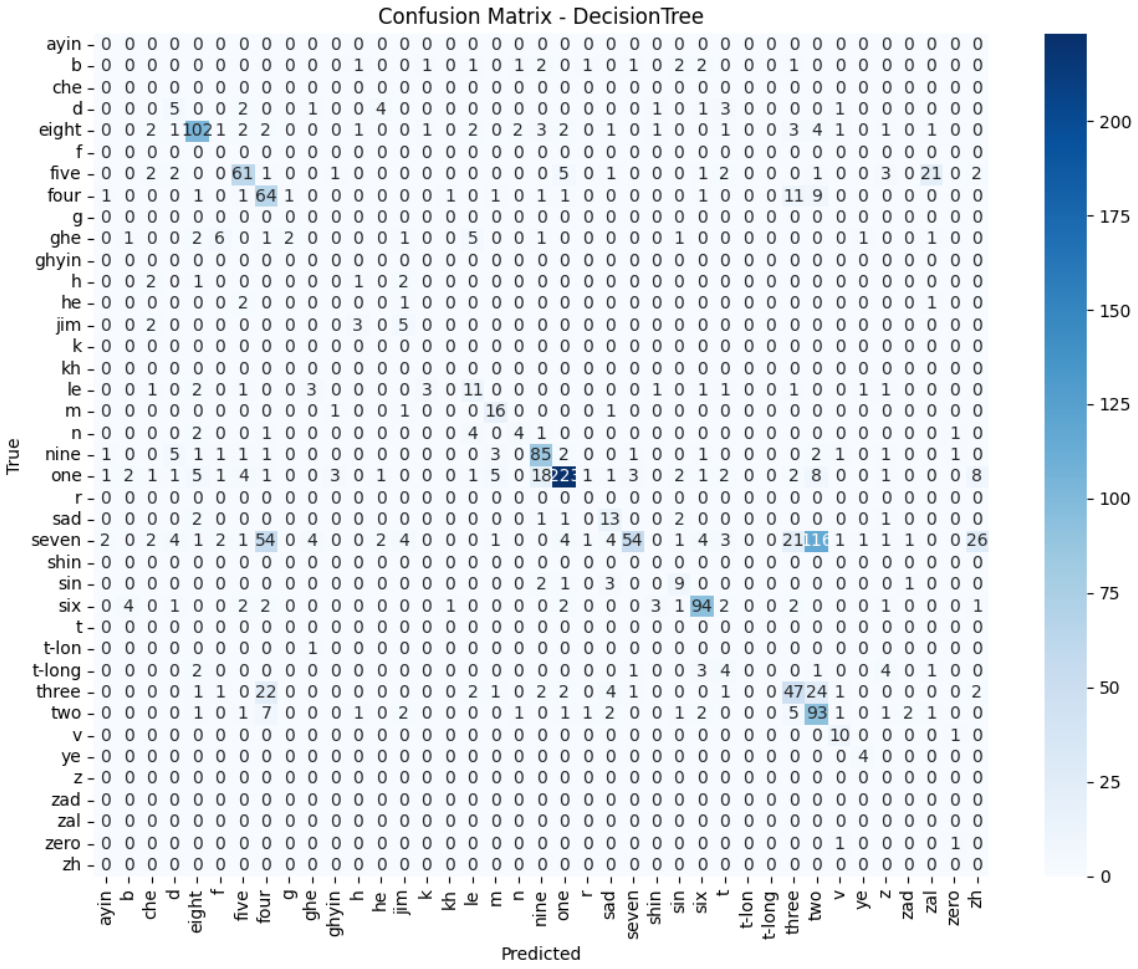

Figure 2: Confusion Matrix for Decision Tree Model

Analysis: This model struggles to generalize, especially under noise and distortion from augmented data. Frequent confusion among many characters and poor diagonal consistency in the confusion matrix reflect its limited capacity to capture distinguishing features.

Summary

Although both models failed to achieve high accuracy at the plate level, the SVM model outperformed the Decision Tree classifier by a notable margin in all metrics. Improving

the feature extraction pipeline, increasing training samples, or employing deep learning models may be necessary for substantial performance gains in future iterations.