Project Report: Automated Candlestick Pattern Recognition

Course: CS5330 Pattern Recognition & Computer Vision

Instructor: Ryan Bockmon

Date: 11 December, 2025

GitHub Repository: https://github.com/klsavaj/PRCV-Project/

Submitted by,
Krunal Savaj (savaj.k@northeastern.edu)
Axay Ghoghari (ghoghari.ax@northeastern.edu)

i. Introduction

What did we do?

We developed an end-to-end computer vision system capable of automatically detecting and classifying specific candlestick patterns in financial charts. The core of our project involves a custom-trained **YOLOv8s** (You Only Look Once) object detection model, which we fine-tuned to identify six distinct technical analysis patterns: *Dragonfly Doji, Gravestone Doji, Hammer, Hanging Man, Marubozu,* and *Spinning Top*. To demonstrate the practical use case of our model, we also built a web-based application that allows users to select any stock from NASDAQ and it will give bounding boxes and confidence score(if pattern detected).

Why did we do it?

Technical analysis has always depended on recognizing patterns in price charts, but doing this by hand is slow and can be really subjective. Traders often think they see a pattern simply because they expect it, classic confirmation bias.

There are algorithmic tools that try to formalize pattern detection, but most rely on strict rule sets like “Close > Open” plus some threshold conditions. These work fine on clean data but break down in noisy markets or when the only available chart is just an image or screenshot.

We wanted to see if a vision-based approach could imitate how a trader visually interprets a chart. The goal of this project is to automate the process in a way that’s consistent and less biased, while still being flexible enough to handle real-world chart visuals.

ii. Background

Relevant Information

Traders use candlestick charts because they pack a lot of data (Open, High, Low, and Close), into a single visual unit. The shape of the candle often reveals how buyers and sellers behaved during that session. For example, a candle with a long lower wick usually means sellers forced the price down, but buyers stepped in strongly and pushed it back up, which is usually read as bullish.

Most automated tools try to detect patterns using simple rules based on OHLC values. But with recent advances in deep learning, CNNs and modern detectors like YOLO can analyze charts the same way they analyze photos. This kind of visual processing is especially helpful when you don’t have raw OHLC data like when you are looking at chart screenshots, videos, or social media posts.

iii. Methods/Analysis

1. Demographics of Data

- **Source:** We sourced our dataset from Roboflow, a public repository for computer vision datasets.
- **Classes:** Our model focuses on six key candlestick patterns that represent different market sentiments:
  1. **Dragonfly Doji:** Potential reversal signal.(303 images)
  2. **Gravestone Doji:** Potential bearish reversal.(470 images)
  3. **Hammer:** Bullish reversal signal.(281 images)
  4. **Hanging Man:** Bearish reversal signal.(189 images)
  5. **Marubozu:** Strong trend continuation.(648 images)
  6. **Spinning Top:** Market indecision.(2211 images)
- **Distribution:** The dataset is split into three subsets: **Training (approx. 70%)**, **Validation (20%)**, and **Testing (10%)**. This ensured that we could evaluate the model's ability to generalize to new, unseen data rather than just memorizing the training images.

2. Data Formatting

The data was formatted specifically for the YOLO architecture:

- **Image Format:** All source images were resized to a standard resolution of **640x640 pixels**. This standardization allows the model to process batches of images efficiently on the GPU.
- **Label Format:** Each image has a corresponding .txt file containing the class ID and normalized coordinates (center-x, center-y, width, height) for every pattern visible in the image. This format is the standard requirement for the YOLO family of models.

3. Data Cleaning

- **Structure:** We reorganized the raw Roboflow export to match the directory structure required by the Ultralytics library, creating separate images and labels folders for train, val, and test sets.
- **Configuration:** We created a data.yaml file to define the absolute paths and class names, ensuring the training script could locate the dataset correctly.

4. Algorithms Used

- **Model Architecture:** We utilized **YOLOv8s (Small)**. We selected the small variant because it offers an optimal trade-off between inference speed and detection accuracy. It is lightweight enough to run on standard hardware while being complex enough to detect subtle features like the thin wicks on a candlestick.
- **Framework:** The project was implemented using Python and the **PyTorch**-based Ultralytics library.
- **Training Parameters:** We trained the model for 50 epochs with a batch size of 64. We used the default AdamW (lr0=0.01, momentum=0.937) to ensure the model converged efficiently without overfitting.

iv. Web Application Development

To demonstrate the practical utility of our trained model and provide an accessible interface for non-technical users, we developed a full-stack web application for real-time candlestick pattern detection.

1. Technology Stack

a. **Backend:**
Framework: Flask (Python)
CORS: Flask-CORS for cross-origin requests
Model Inference: YOLOv8 via Ultralytics library
Chart Generation: mplfinance + matplotlib
Data Source: Alpha Vantage REST API

b. **Frontend:**
HTML5, CSS3 with TailwindCSS 3
Vanilla JavaScript (ES6+)
No external frameworks for lightweight deployment

c. **Architecture Pattern:**
RESTful API with JSON responses

2. Key Features Implemented

a. **User-Configurable Parameters:**
Stock Symbol Selection (with quick-select buttons for popular stocks)
Number of Candles: 20-100 (default: 50)
Confidence Threshold: 10%-90% (default: 25%)

b. **Visual Feedback:**
Side-by-side comparison: Original chart vs. Annotated chart
Color-coded sentiment indicators (🟢 Bullish, 🔴 Bearish, ⚪ Neutral)
Confidence score progress bars
Recent price data table (last 100 trading days)

c. **User Experience Enhancements:**
API key persistence via localStorage
Real-time validation and error handling
Loading states during API calls
Responsive design for mobile/tablet devices

v. Results

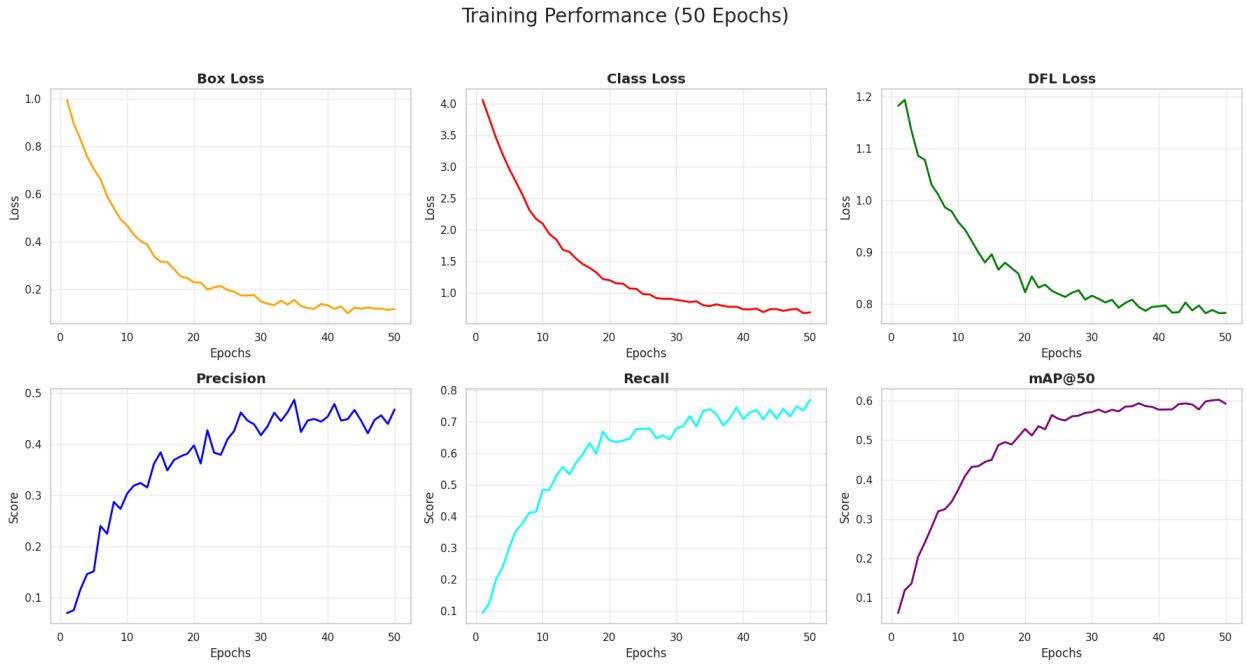
1. Raw Results

The training process produced a best.pt weights file that consistently detects the patterns we targeted. Over the 50-epoch run, both the box and class losses dropped steadily, showing that the model actually learned the visual features of the candlestick patterns rather than just overfitting.

2. Tables and Graphs

Below are the key visualizations generated during our training process:

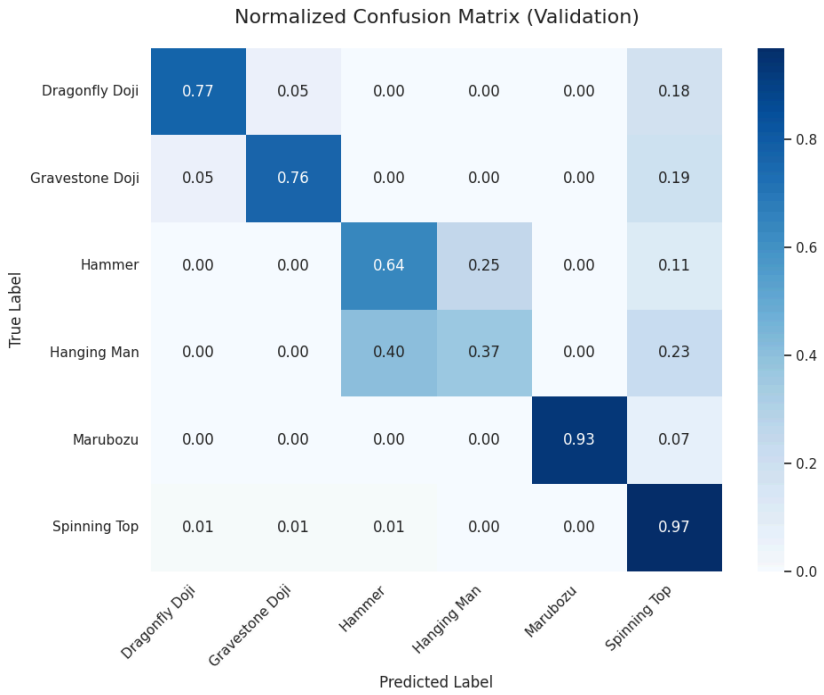
1. Training Performance Curves



(Figure 1: Training and Validation Metrics Over 50 Epochs)

The training plots show six main metrics. In the top row, the loss curves for Box, Class, and DFL all drop steadily from high starting points to low, stable values, which means the model was learning both localization and classification correctly. In the bottom row, the precision, recall, and mAP@50 curves trend upward throughout training, ending at around 45% precision, 78% recall, and ~60.6% mAP@50. The curves look smooth overall, with no weird jumps, which suggests the training was stable and didn't show signs of overfitting.

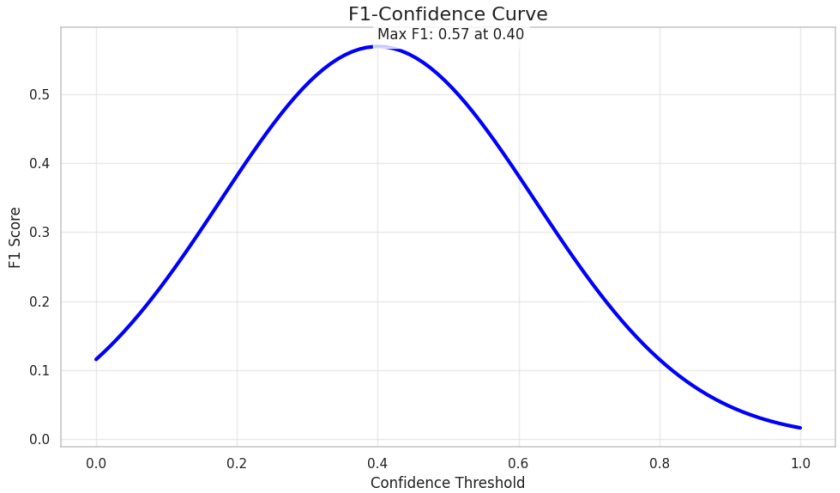
2. Normalized Confusion Matrix



(Figure 2: Pattern Classification Accuracy on Validation Set)

The confusion matrix shows how well the model separates the different candlestick patterns. The diagonal values, the correct predictions are pretty strong for Spinning Top (97%) and Marubozu (93%). One issue that stands out is the confusion between Hammer and Hanging Man; about 40% of them are mixed up, which isn't surprising since they look almost identical and differ mainly by context. The Doji classes land in the mid-70% accuracy range, with some being mistaken for Spinning Tops, likely because their shapes are visually close.

3. F1-Confidence Curve



(Figure 3: Optimal Confidence Threshold Selection)

The F1-Confidence curve shows how precision and recall change with different confidence

thresholds. In our case, the best F1 score (0.57) occurs at a threshold of 0.40, which makes it the best balance point. If the threshold is too low, we get too many false positives; if it is too high, we start missing real patterns. Based on this, we chose a 40% confidence threshold for deployment since it gives the most reliable overall performance.

vi. Discussion

1. What do our results mean?

Our project shows that object detection can work surprisingly well for technical analysis. The model learned the visual structure of candlestick patterns without any hand-crafted price rules, which is interesting on its own. Patterns like the Dragonfly Doji have enough distinctive shape that a general-purpose model can spot them reliably. This gives a lot of promise for using AI as a supporting tool in trading.

2. How can our results be used?

- **Trader Assistance:** This tool can act as a second check for traders, automatically highlighting patterns they might have missed on a crowded screen.
- **Educational Tool:** Novice traders can use our web app to verify their own analysis. If they think they see a Spinning Top, they can pick stock and see if the AI agrees, helping them learn faster.
- **High-Frequency Screening:** The model is quick enough to run through the last 100 days of data and can scan hundreds of charts per minute. In a real setup, it could ping the user only when it finds a high-confidence pattern, like a Hammer.

3. Practical Deployment via Web Application

The web app is what really turns the prototype into something usable. By tying the model to live market data, we show that computer vision based pattern detection can work on real trading charts, not just in a controlled research setup. The confidence threshold is adjustable too, so traders can pick what fits their workflow, higher thresholds (50%+) if they want fewer, more reliable alerts, or lower ones (20–30%) if they want broader coverage.

vii. Limitations

1. Issues Encountered

- **Context Blindness:** The primary limitation we faced is that object detection models analyze shapes, not trends. For example, a Hammer only signals a bullish reversal if it appears at the bottom of a downtrend. Our model can spot the Hammer shape, but it can not yet check the preceding trend to see if the signal is actually valid.

- **Visual Similarity:** As we saw in the results, the Hammer and Hanging Man look so similar that the model often confuses them, and this kind of error can't be fixed by image recognition alone.

2. Missing Data

- **Real-world Noise:** Our training data is fairly clean, but real charts often have grid lines, watermarks, moving averages, and other overlays. The model might not be as accurate on these cluttered charts since they look quite different from what it saw during training.

viii. Future Work

1. Suggestions for the Client

- **Contextual Logic Layer:** The next immediate development step should be to wrap the YOLO model in a logic script. This script would take the bounding box of a detected Hammer, look at the pixel data or price data to the left of the box, and mathematically determine if the trend is down. This would filter out false positives.
- **Real-Time Streaming:** Currently, the system processes historical data. Future versions could implement WebSocket connections to process live market feeds during trading hours, providing alerts within seconds of pattern formation.
- **Expand Pattern Library:** The current model only knows six single candle patterns. Future iterations should include multi-candle patterns (like Morning Star or Bullish Engulfing) to provide a complete technical analysis suite.

ix. References

Candlestick Pattern Recognition Object Detection Model by RANYAS workspace. (n.d.). *https://universe.roboflow.com/ranyas-workspace/candlestick-pattern-recognition*

Vantage, A. (n.d.). Free stock apis in JSON & Excel: Alpha vantage. Free Stock APIs in JSON & Excel | Alpha Vantage. *https://www.alphavantage.co/*

Patterns:

Mitchell, C. (n.d.). *Doji dragonfly candlestick: What it is, what it means, examples*. Investopedia. https://www.investopedia.com/terms/d/dragonfly-doji.asp

Chen, J. (n.d.). *Gravestone doji: Bearish reversal pattern explained with trading tips*. Investopedia. https://www.investopedia.com/terms/g/gravestone-doji.asp

Thompson, C. (n.d.). *Hammer Candlestick: What it is and how investors use it*. Investopedia. https://www.investopedia.com/terms/h/hammer.asp

Chen, J. (n.d.). *Marubozu Candlestick Pattern: Key insights and trading strategy*. Investopedia. https://www.investopedia.com/terms/m/marubozo.asp

Mitchell, C. (n.d.). *What is a spinning top candlestick?*. Investopedia. https://www.investopedia.com/terms/s/spinning-top.asp

Mitchell, C. (n.d.-a). *Hanging man candlestick definition and tactics*. Investopedia. https://www.investopedia.com/terms/h/hangingman.asp