Car License Plate Detection & OCR Readiness Prediction

This project was developed for the Introduction to Data Science course at the University of Tehran. It focuses on advanced license plate recognition.

Project Goals and Objectives

Primary Goal

Precisely detect car license plates in images.

Secondary Goal

predict suitability for optical character recognition (OCR).

Project Development Phases

Phase 1: Data Collection, and Initial Visualization

A dataset of car images with plate annotations was collected and initially explored using visualizations.

Phase 2: Data Storage and Processing Pipeline

The dataset was structured in an SQLite database and processed through a modular Python pipeline. Preprocessing and feature engineering were automated and integrated with CI/CD using GitHub Actions.

Phase 3: Model Training, Evaluation & Integration

We developed a CNN model for plate detection and a classifier to predict OCR readiness. Both models were integrated into a full pipeline for automated inference and evaluation.

Phase 1

Dataset Overview

- **Dataset Overview:** Contains ~450 car images. Plates are annotated.
- **Data Source:** Public Kaggle dataset. Readily accessible for research.
- **Primary Purpose:** Used for detection and OCR-readiness model training.
- **Image Format:** All are .png files. Mean resolution: 290×423, RGB
- **Annotation Details:** .xml files, Pascal VOC format. Image metadata included.
- **Bounding Boxes:** Coordinates (xmin, ymin, xmax, ymax) define plates. Label is "licence".

Sample Images

Below are sample car images from the dataset, illustrating the variety of inputs for license plate detection and OCR readiness prediction.

Annotated Sample Image

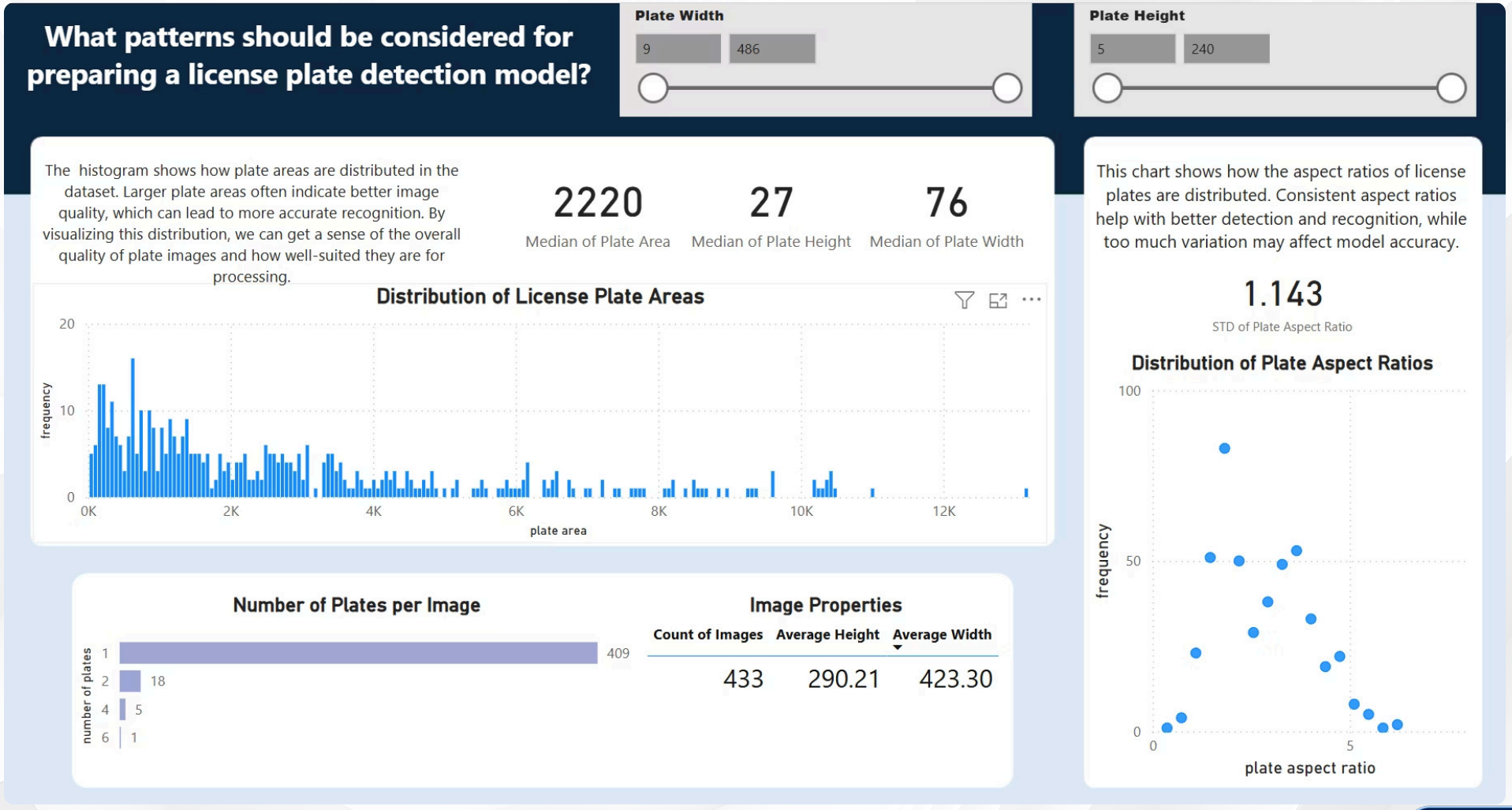
```xml
<annotation>
    <folder>images</folder>
    <filename>Cars0.png</filename>
    <size>
        <width>500</width>
        <height>268</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>licence</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <occluded>0</occluded>
        <difficult>0</difficult>
        <bndbox>
            <xmin>226</xmin>
            <ymin>125</ymin>
            <xmax>419</xmax>
            <ymax>173</ymax>
        </bndbox>
    <ocr_text>KLG1CA2555</ocr_text><worth_ocr>1</worth_ocr></object>
</annotation>
```

Annotation Details

- License plates are precisely defined with bounding boxes.
- Label used for plates: "licence".
- Annotation format: Pascal VOC (.xml files).
- Bounding box coordinates include: xmin, ymin, xmax, ymax.

This visual demonstrates how license plates are localized within each car image for model training.

Initial Visualization by PowerBI



Phase 2

Pipeline Stages

Data Ingestion

Import raw images and annotations to a database.

Data Splitting

Split data into training and test sets, including manual images.

Image Preprocessing

Preprocess images by cleaning and normalizing data.

Feature Engineering

Perform feature engineering, extracting bounding boxes.

Detection Model Training

Train a detection model to find license plates.

OCR Suitability Classifier

Train an OCR model on detected plates for character recognition.

Database Storage & Querying

Embedded Solution

SQLite is file-based, requiring no dedicated server. This simplifies project setup.

Easy Integration

It streamlines development and deployment. Ideal for iterative prototyping.

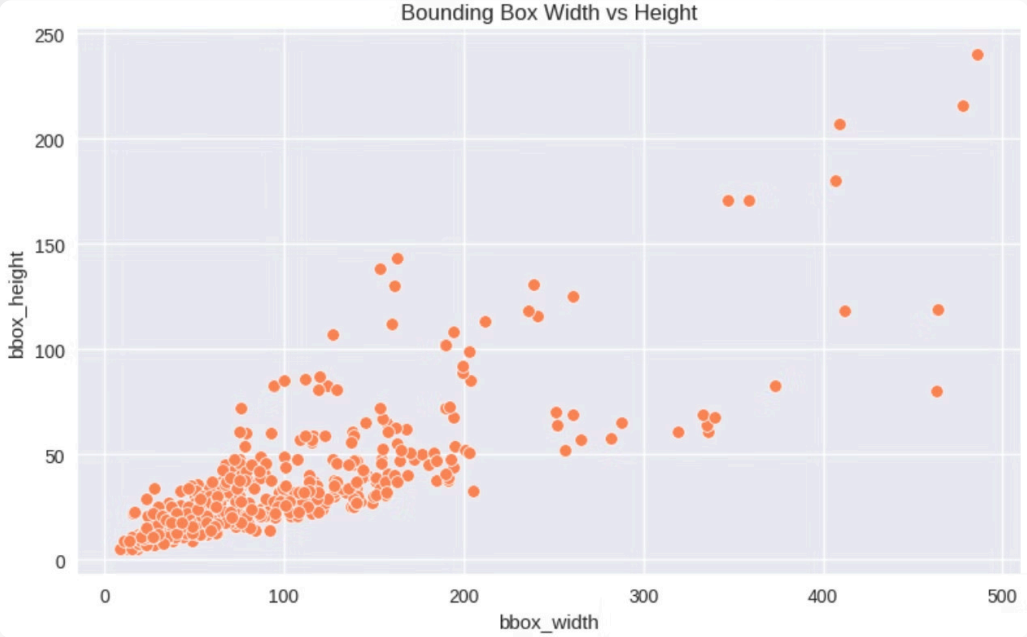
Versatile Storage

Stores raw images, annotations, and processed features efficiently.

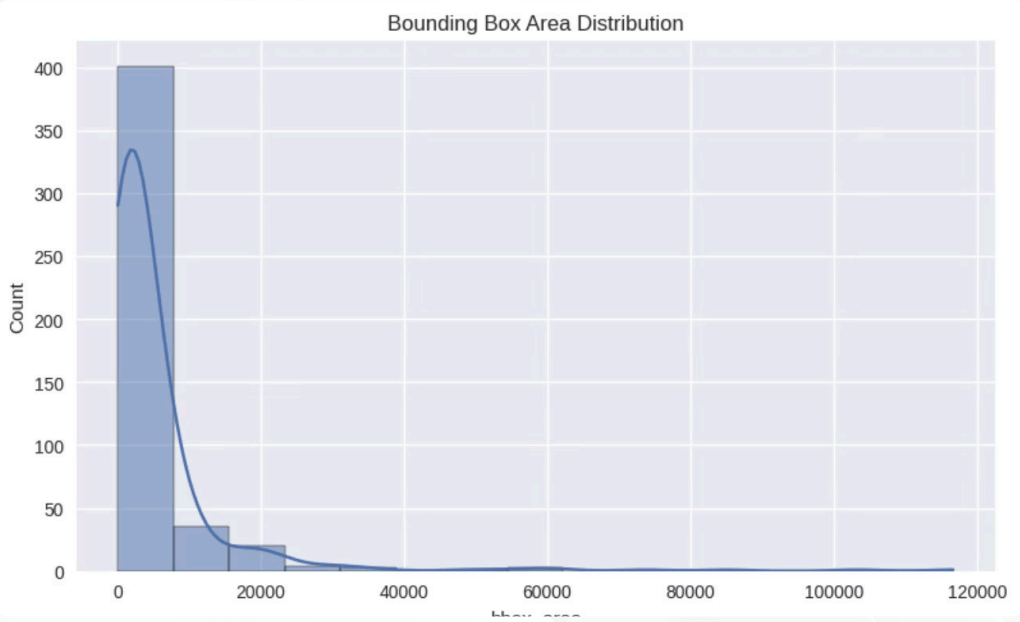
Local Data Handling

Facilitates local data management. Perfect for individual model training.

Exploratory Data Analysis (EDA)



License plate lengths range from 0 to 100 pixels, and widths range from 0 to 50 pixels



The area of most plates is around 1000 pixels² (showing an exponential-like distribution)

Feature Engineering for Worth-OCR Prediction

Key Engineered Features:

- bbox_width, bbox_height – Plate dimensions in pixels.
- bbox_area – Total pixel area of the bounding box.
- aspect_ratio – Width-to-height ratio (format consistency).
- area_fraction – Plate size relative to the full image.
- center_x_norm, center_y_norm – Normalized plate center positions.
- margin_* **(left, right, top, bottom)** – Distance from plate to image borders.

These features collectively provide rich information about the visual and spatial properties of plates. They will support the **OCR preprocessing pipeline** in the next phase by enabling:

1. Selection of consistent plate crops for input into OCR models.
2. Filtering or augmenting images with small, poorly framed, or off-centered plates.
3. Data quality checks for outlier bounding boxes before modeling.

Image Preprocessing & Dataset Finalization

1

Image Preprocessing:

To prepare the cropped plate regions for modeling:

- **Cropping** plates from original images using bounding boxes
- **Resizing** to 128×32 pixels (uniform model input)
- **Grayscale conversion** to reduce noise
- **Normalization** of pixel values to [0, 1]

2

Comprehensive Data Cleaning:

- **Removed invalid entries** with zero-sized bounding boxes
- **Normalized** numerical features using Min-Max scaling
- **Removed highly correlated** features ($\rho > 0.9$) to reduce redundancy
- **Labeled blurry images** using Laplacian variance

3

> Outcome: A clean, compact dataset of uniformly preprocessed plate images, ready for model training and OCR readiness classification.

Blurriness Detection & Labeling

Assessing the visual clarity of license plate crops to filter low-quality samples and support OCR-readiness modeling.

Method: Laplacian Variance

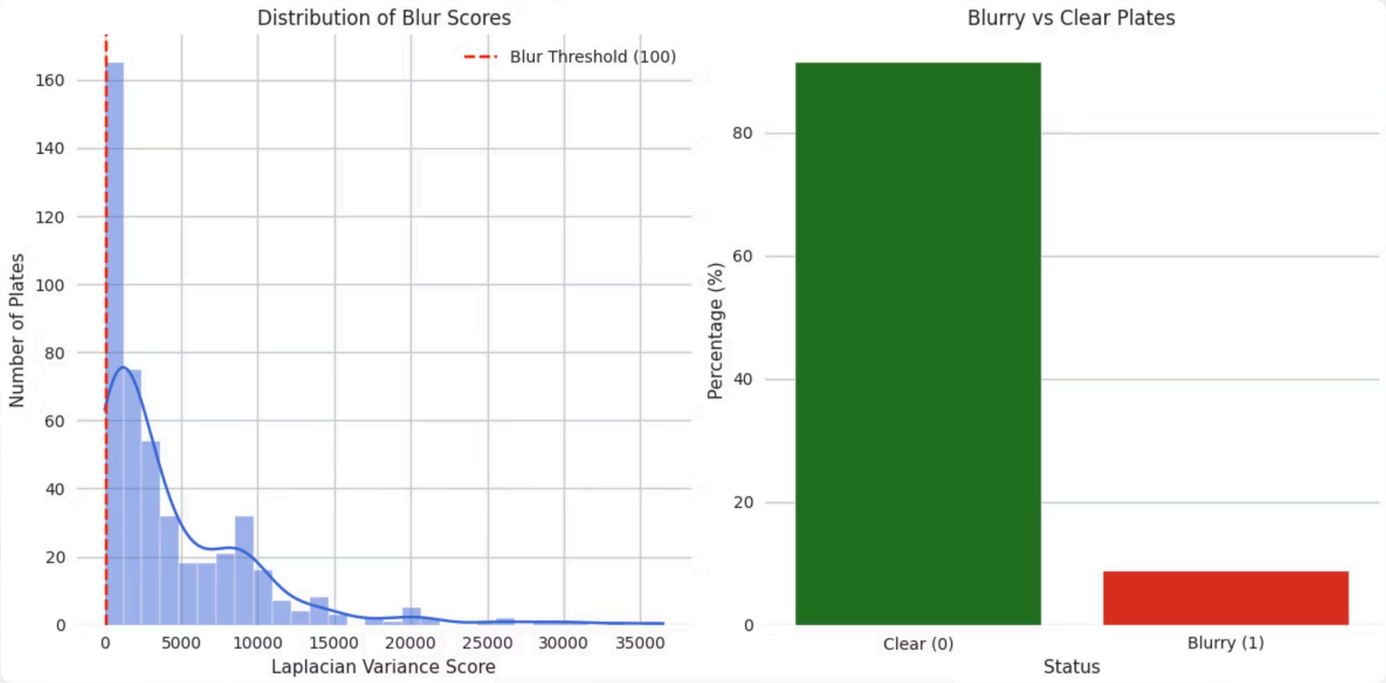
Each **grayscale plate image** was passed through the **Laplacian operator** to detect edges. The **variance** of this output (`blur_score`) was used as a sharpness metric. Images with a `blur_score < 100` were classified as blurry.

Results & Statistics

Out of 450 analyzed plates, **41 (~8.7%) were identified as blurry**. The mean blur score was 4391.7, with a median of 2294.7, and a standard deviation of 5507.1.

Database Integration

Both `blur_score` and `is_blurry` flags were stored in the `engineered_plate_features` table. This enables selective filtering and robust performance evaluation under varying input conditions.



Phase 3

License Plate Detection Models

Traditional CNN

This custom regression-based model predicts bounding box coordinates directly. It features convolutional layers and a 4-value output.

Pros: Simple, lightweight, and offers fast inference. Cons: Less robust to complex images or scale variations.

Faster R-CNN

We used a pretrained object detection network, fine-tuned for plate detection. It includes a Region Proposal Network and a CNN backbone.

Pros: Provides high precision and strong generalization. Cons: Heavier model with slower inference times.

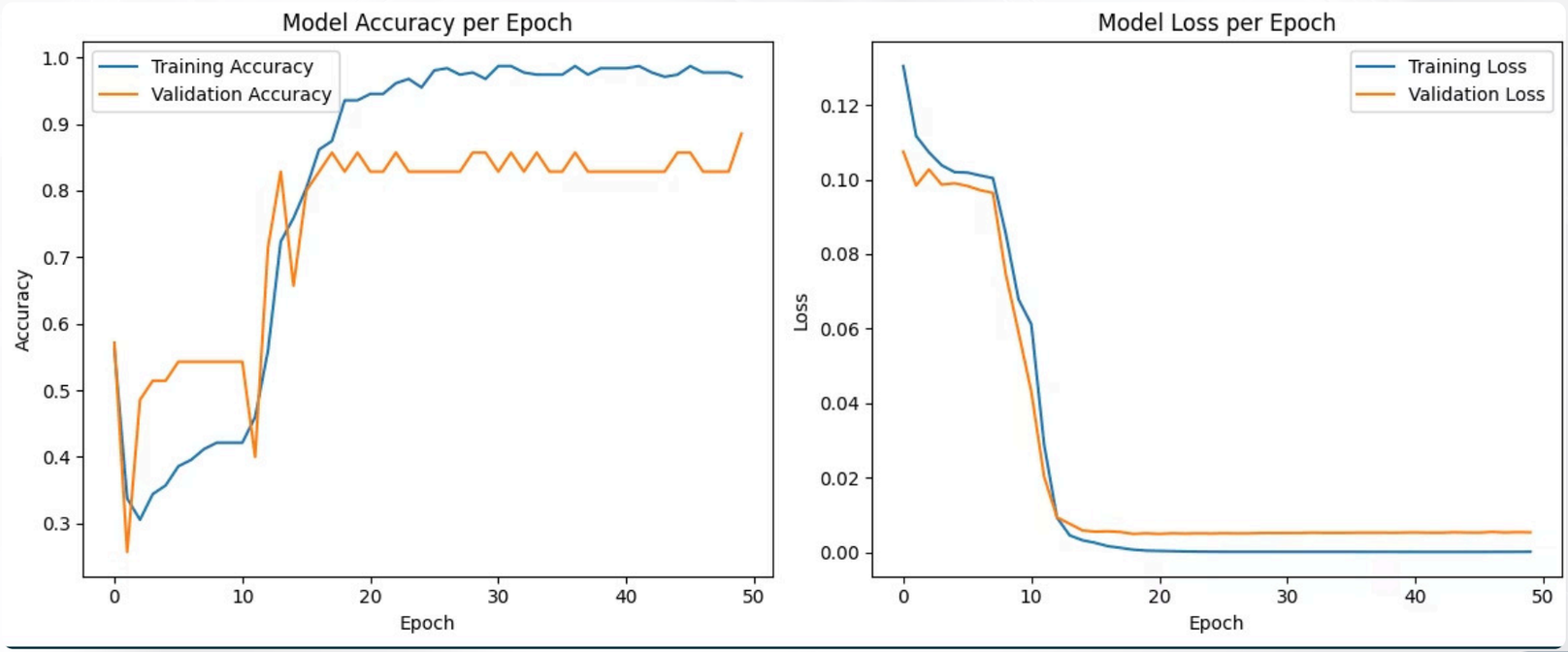
Traditional CNN – Model Architecture

Custom Regression Layers

- Pretrained **VGG16** (ImageNet weights) used as the base (`include_top=False`).
- VGG16 is followed by custom regression layers:
- `Flatten` layer
- `Dense(128, relu)` layer
- `Dense(128, relu)` layer
- `Dense(64, relu)` layer
- `Dense(4, sigmoid)` for predicting normalized bounding box coordinates.

Training & Parameters

- VGG16 weights are **frozen** (non-trainable).
- Only the dense layers are trained.
- **Total Parameters**: ~17.95 million
- **Trainable Parameters**: ~3.24 million
- **Non-trainable Parameters**: ~14.71 million



Faster R-CNN – Performance Overview

94%

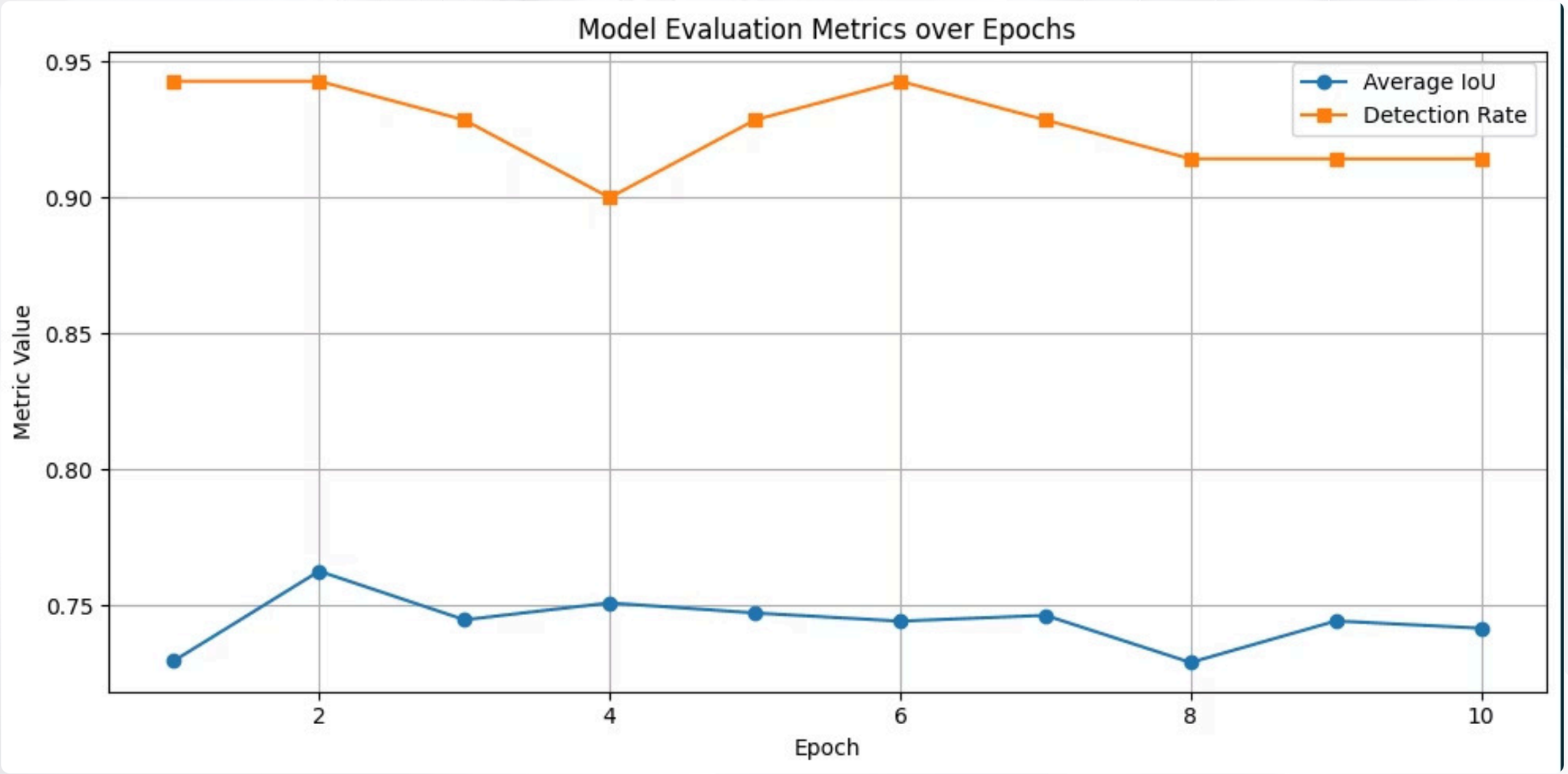
COCO Pre-trained

4

Batch Size

10

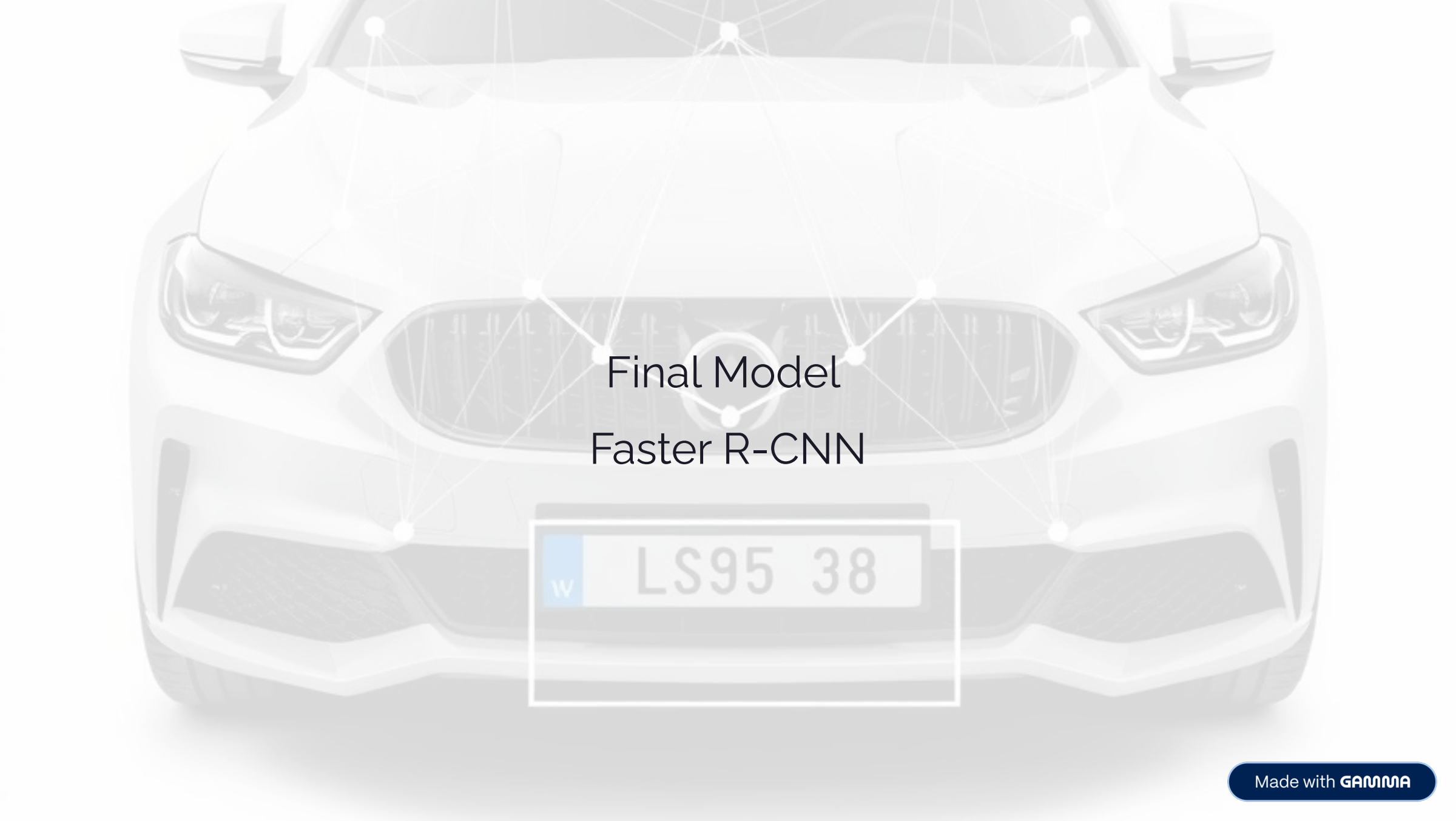
Epochs



Evaluation of Detection Models

Models were evaluated using tailored metrics.

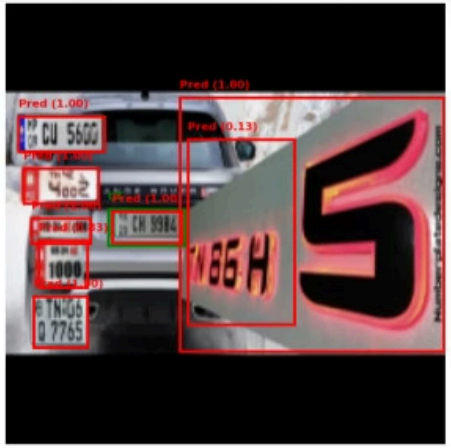
	Traditional CNN	**Faster R-CNN**
Metric	Accuracy (MSE)	IoU
Result	**79.31%**	Average IoU: 0.8082, Detection Rate @ IoU > 0.5: **94.12%**
Insight	No direct spatial overlap evaluation.	Superior localization precision.

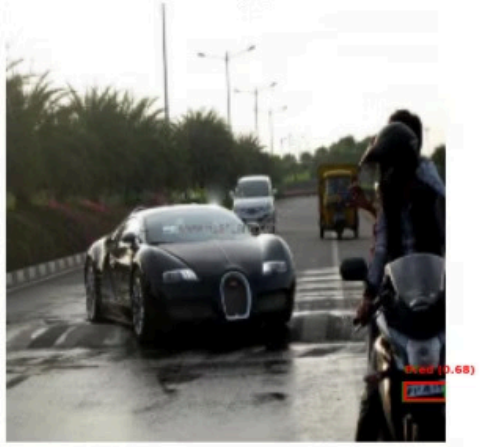


Final Model

Faster R-CNN

Sample Predictions



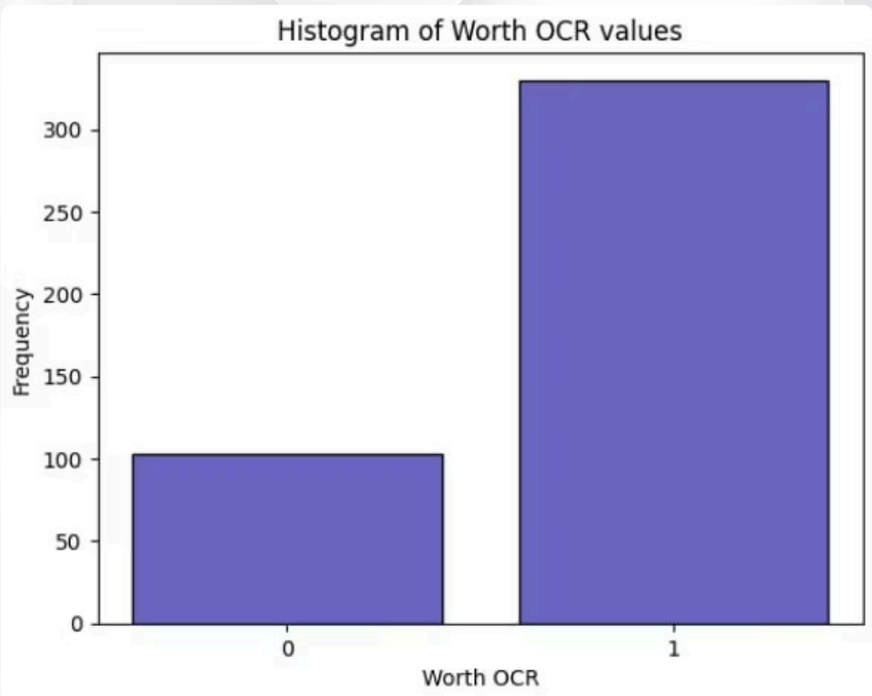


Worth OCR Prediction: Model Training

Models were trained on features like blur score, bounding box size, area, and aspect ratio. We used three classifiers for worth OCR prediction.

Model	Accuracy	F1 (Positive)
Random Forest	0.88	0.93
Logistic Regression	0.79	0.88
XGBoost	0.85	0.91

Random Forest demonstrated the best overall performance for predicting OCR readiness.



OCR Phase Overview

Models Explored

- EasyOCR: Lightweight, CNN + LSTM based, offering real-time speed.
- TrOCR: Microsoft's Transformer-based model, powerful with complex text.

OCR Results Summary

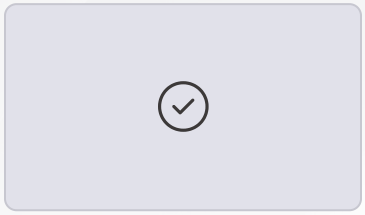
Performance Metrics

Metric	EasyOCR	TrOCR
Exact Match (%)	13%	26%
Normalized Match (%)	22%	50%
Levenshtein Distance	3.19	1.68
Character Accuracy	41%	67%

Key Takeaways

- TrOCR consistently shows superior performance over EasyOCR across all metrics.
- Normalization, like mapping 'O' to '0' or 'I' to '1', significantly enhances accuracy.
- OCR models still face challenges with blurry or highly distorted license plates.

Conclusion and Future Work

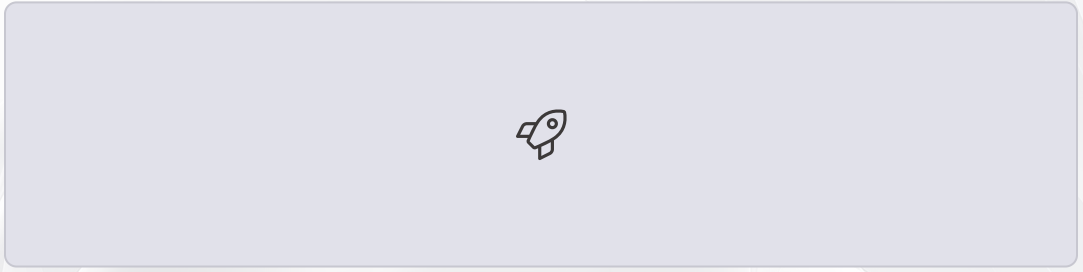


Accurate Plate Detection

Faster R-CNN achieved 94.12% detection rate with 0.8082 average IoU.

Promising OCR Readiness

Random Forest achieved 0.88 accuracy and 0.93 F1 score in worth OCR prediction, demonstrating strong performance for OCR readiness.



Future Enhancements

Focus on OCR , real-time systems, and edge device deployment.

This project successfully achieved its primary goal of robust license plate detection and demonstrated promising capabilities for OCR readiness prediction. Moving forward, efforts will concentrate on further enhancing OCR accuracy, developing real-time detection systems, and enabling model deployment on edge devices for practical, real-world applications.-