**Sign Recognition Web Application**

*Flask, TensorFlow, Keras, MySQL, CNNs*

* Built and deployed a web-based image classification system to predict signs from user-uploaded images using pre-trained Convolutional Neural Networks (CNNs).
* Trained multiple models on diverse datasets and saved them as .h5 files for scalable, inference-ready deployment.
* Used .pkl files to store label encoders for converting numerical predictions into human-readable class labels.
* Implemented modular preprocessing (resizing, normalization) and dynamic model loading to support multiple datasets.
* Integrated with a MySQL backend to handle user interactions and enable future analytics.
* Demonstrated end-to-end machine learning deployment skills, from model training and optimization to user-facing web integration.

**Key files include:**

* app.py: Likely the main application file.
* prediction.py: Possibly contains machine learning or data prediction logic.
* Admin.py: May handle admin-related functionality, possibly for a user interface or access control.

**Use of .h5 Files:**

* The .h5 files in the project store **trained deep learning models** (using **Keras/TensorFlow**).
* Each file contains the complete architecture, weights, and training configuration of a **Convolutional Neural Network (CNN)** trained for image classification.
* These files are loaded at runtime to perform **inference** (i.e., predicting the class of a new input image) without needing to retrain the model.

**Use of .pkl Files:**

* The .pkl (Pickle) files are used to **serialize and store label mappings** (i.e., class label dictionaries).
* They map numeric class indices predicted by the model back to **human-readable labels** (e.g., from 3 to "Stop Sign").
* This enables clear and interpretable prediction results in the application interface.