A Comparative Study and Hyperparameter Tuning DL Methodology Using Ensemble Learning

Submitted by:
Ganesh Sesha Sai Akhil Koutarapu, (AP21110010931)
Venkatesh Komalli, (AP21110010933)
Venkata Arun Kumar Perla(AP21110010970)
Guide: Dr. Ashu Abdul

Abstract

- Created a Streamlit web app that automates CNN model selection and training.
- Uses Optuna to tune activation functions, optimizers, and other hyperparameters.
- Supports CSV (structured) and ZIP (image) datasets.
- Automatically trains and evaluates models like ResNet50, VGG16, MobileNetV2, DenseNet121.
- Outputs accuracy, precision, recall, F1-score, and confusion matrix in real-time.

Problem Statement And Motivation

- Training deep learning models requires domain expertise, especially CNNs.
- Manual model selection and tuning is time-consuming and error-prone.
- Our goal: Simplify and democratize deep learning through automation.
- Users can experiment without writing code, enabling education and prototyping.

Objectives & Deliverables

Objectives:

- Automate CNN selection, tuning, and evaluation using ensemble learning.
- Build an intuitive UI with real-time metrics and configuration.

Deliverables:

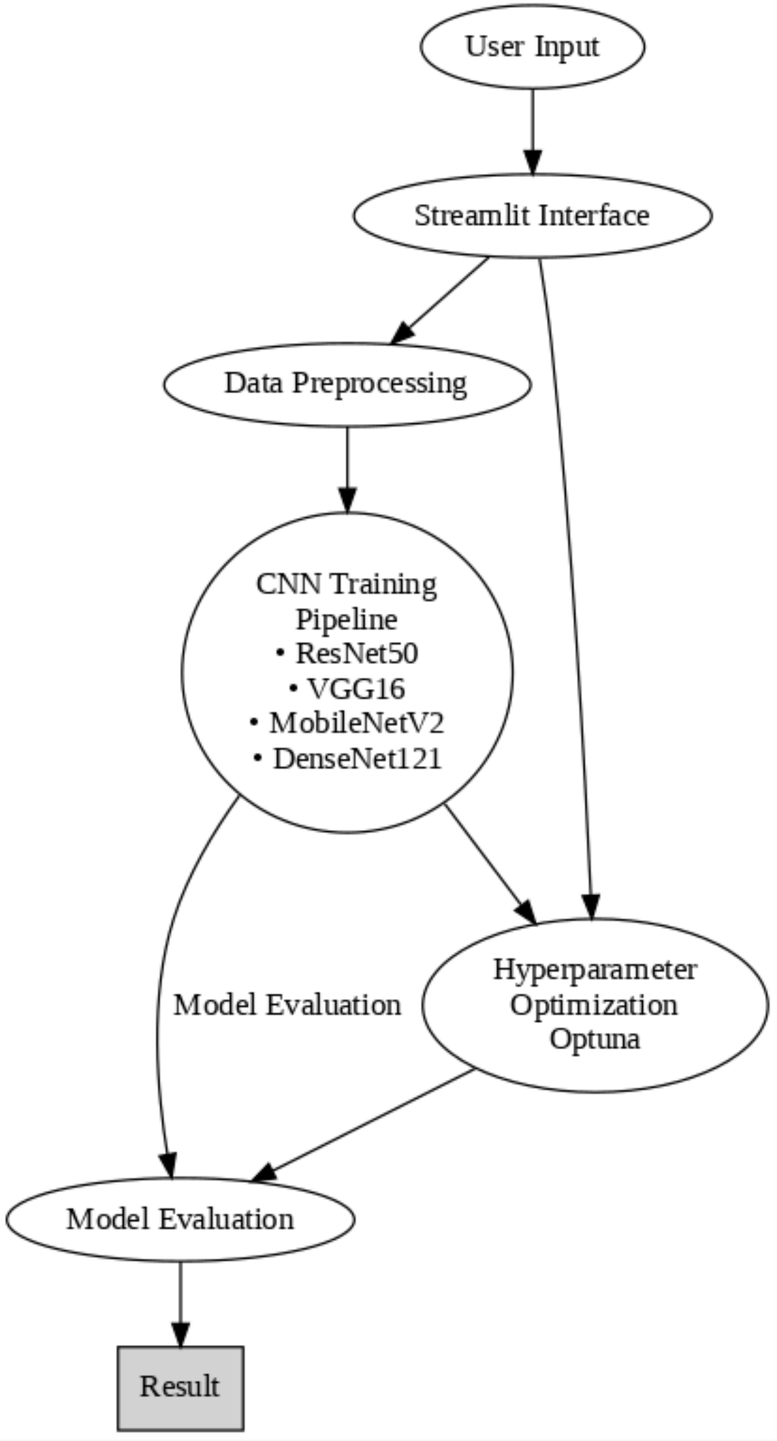
- Streamlit Web Application
- Optuna-powered tuning engine
- Evaluation Dashboard
- Support for multiple dataset formats

Literature Review

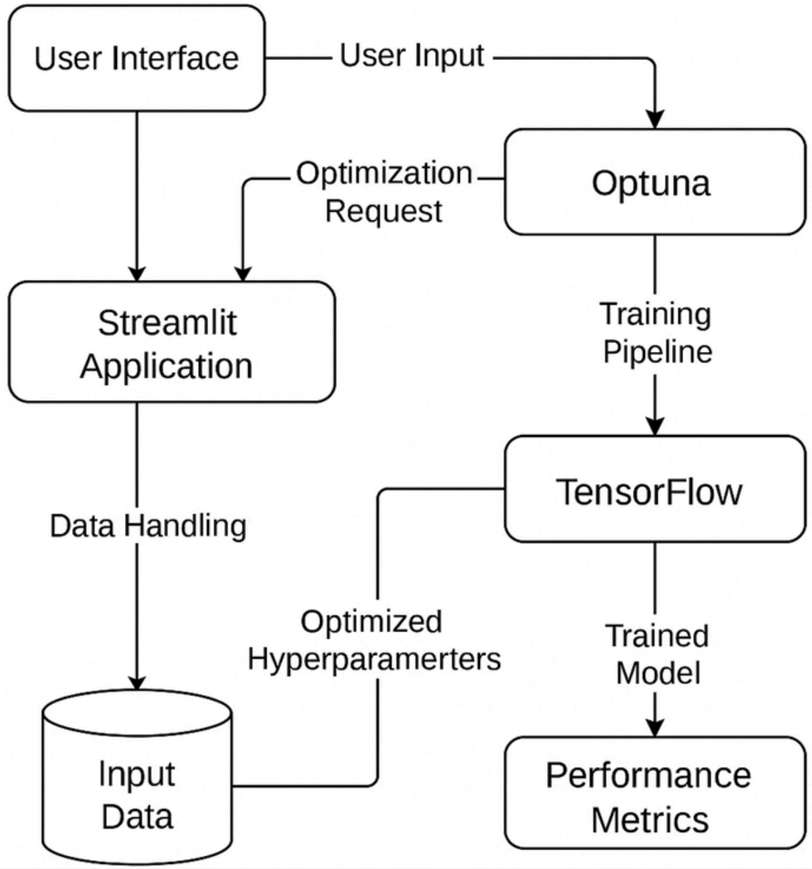
- CNNs like ResNet, MobileNet, VGG, DenseNet used for baseline comparison.
- Studied AutoML tools like AutoKeras, Google AutoML—limited user control.
- Opted for Optuna (efficient, flexible hyperparameter tuner).
- Streamlit selected for rapid, code-free deployment interface.

System Architecture

1. Data Ingestion (CSV or ZIP with images)
2. Preprocessing (resizing, normalization, one-hot encoding)
3. Model Generator (selects CNN model dynamically)
4. Hyperparameter Tuner (Optuna: activation, optimizer, etc.)
5. Model Training & Evaluation
6. UI (Streamlit) with upload, control, and output view



Methodology



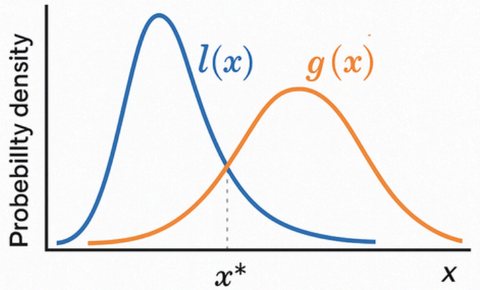
- CSV: Processed using pandas, label encoding, reshaped for image simulation.
- ZIP: Extracted, resized to 224×224, labeled from folder structure.
- Optuna performs 5 trials per dataset, selecting best CNN and hyperparameters.
- Training uses early stopping to avoid overfitting.
- Metrics computed: Accuracy, Precision, Recall, F1-Score, Confusion Matrix.

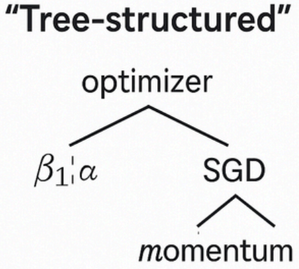
Hyperparameter Tuning with Optuna

Parameters tuned:

- Optimizer: adam, sgd, rmsprop
- Activation: relu, tanh
- Batch size, Epochs (via Streamlit slider)
- Uses Tree-structured Parzen Estimator (TPE) for search.
- Returns best configuration with model evaluation.

Tree-structured Parzen Estimator





$$x = \text{argax}\ \frac{l(x)}{g(x)}$$

- Probability density of good (low-loss) hyperparameter configurations
- Probability density of all other (non-optimal) configurations

Evaluation Metrics

Accuracy: 0.5000

Precision: 0.2500

Recall: 0.5000

F1 Score: 0.3333

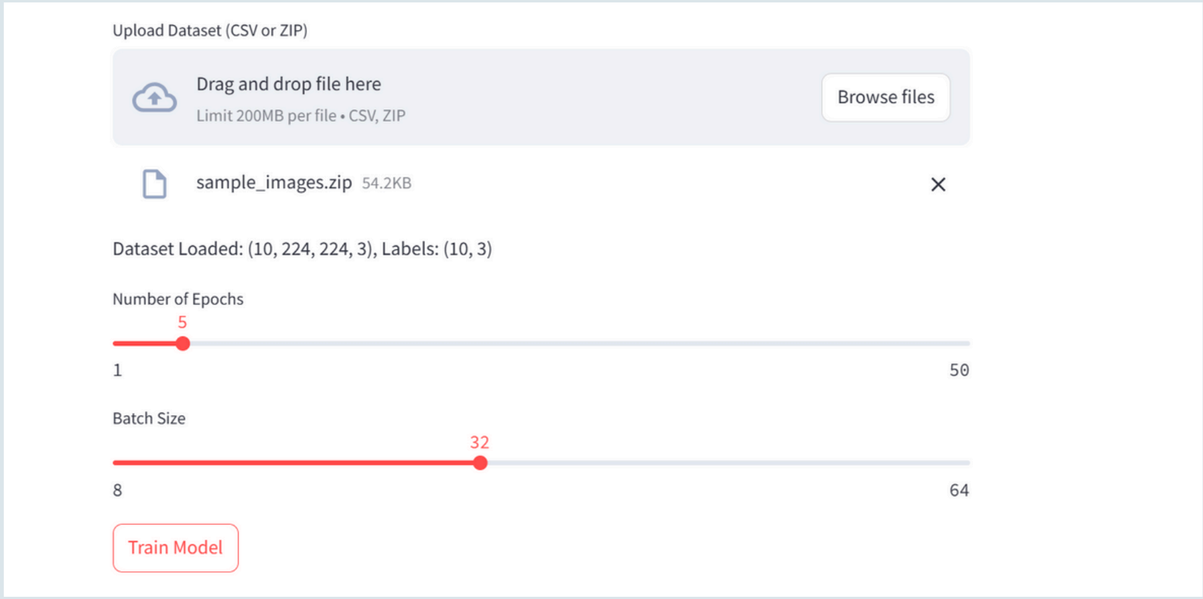
Confusion Matrix:

0	1
0	1
0	1

- Accuracy: Correct predictions / total
- Precision: TP / (TP + FP)
- Recall: TP / (TP + FN)
- F1 Score: Harmonic mean of Precision & Recall
- Metrics calculated using sklearn.metrics
- Confusion matrix displayed as table via st.write().
- TP(True Positive): Correctly predicted positive instances
- .TN(TrueNegative): Correctly predicted negative instances.
- FP(False Positive): Incorrectly predicted positive instances.
- FN(False Negative): Incorrectly predicted negative instances

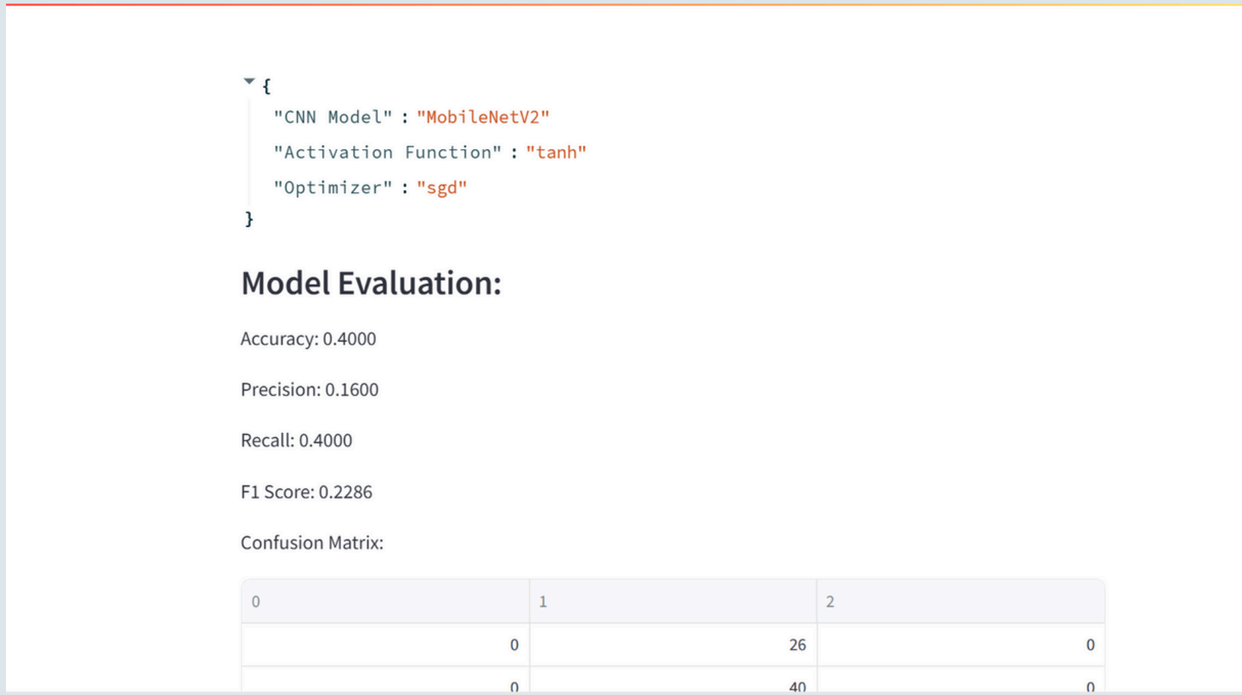
UI Features And Results

- Users upload CSV or ZIP dataset.
- Choose Epochs and Batch Size interactively.
- Click "Train Model" to start pipeline.



Results include:

- Best model architecture
- Hyperparameter config
- Evaluation metrics



Conclusion & Social Impact

- Enables automated CNN training for both technical and non-technical users.
- Reduces barriers in machine learning education and experimentation.
- Ideal for use in academics, research, and prototyping in various domains.
- Brings transparency, reproducibility, and accessibility to deep learning.

Thank you