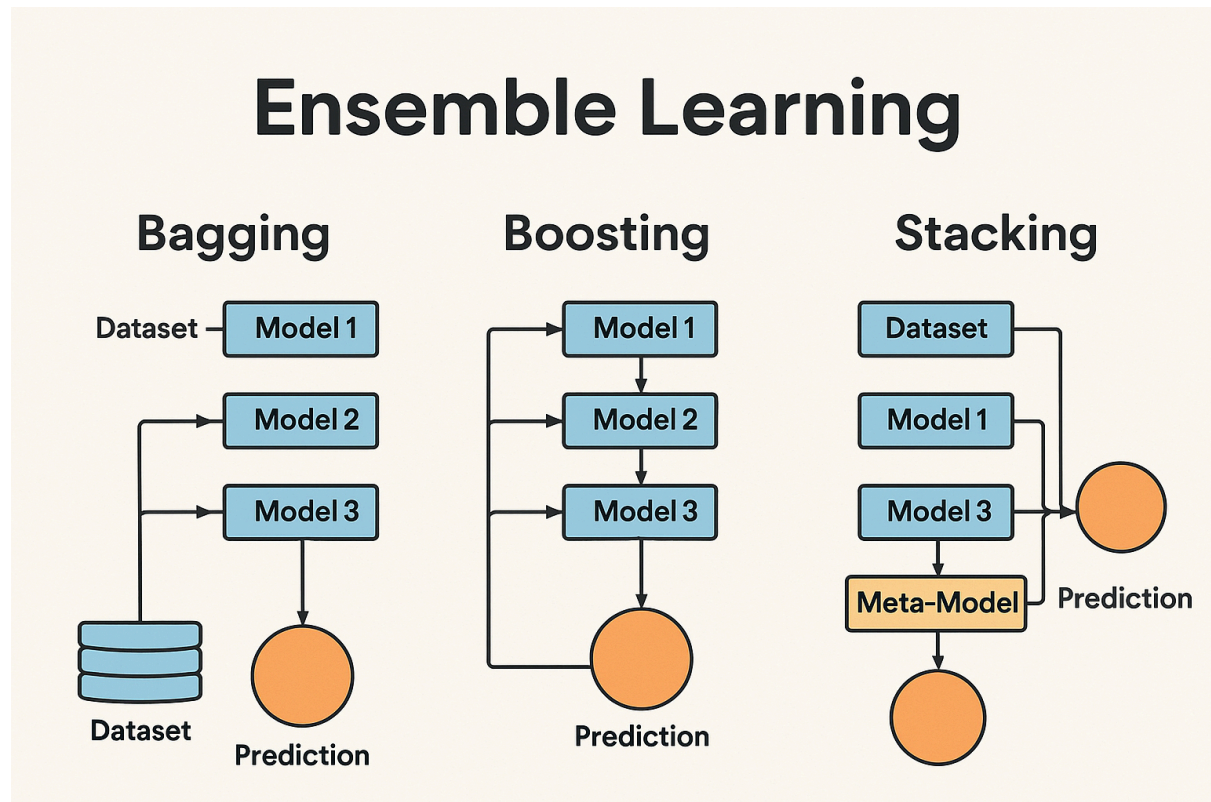


# Ensemble Learning with CIFAR-10 — A Deep Dive into Bagging, Boosting & Stacking

In this post, I'll walk through a computer vision project where I implemented three major ensemble learning techniques on the CIFAR-10 dataset.



## What I Did

1. **Bagging** — Trained multiple LightCNNs independently and combined their outputs.
2. **Boosting** — Used XGBoost on flattened CIFAR-10 images to incrementally improve predictions.
3. **Stacking** — Combined outputs of LightCNN, SmallCNN, and XGBoost to create new meta-features for a logistic regression classifier.

## Challenges & How I Overcame Them

- **Repeated Model Training:** I restructured my code to load only trained models, preventing accidental retraining.
- **Feature Mismatch in XGBoost:** I resized all images to 64x64 and flattened them properly.
- **File Overwrites:** Ensured my training logic was isolated from feature extraction scripts.
- **Torch Warnings:** Acknowledged security issues and planned future use of safer loading parameters.

## **Insights**

- Ensemble methods can significantly improve performance when integrated smartly.
- Clean separation of training vs inference code is essential to avoid duplication or errors.
- Having automated `.npy` exports of meta-features makes stacking easier to scale.

This project forms the backbone for real-time ensemble inference using Flask, with A/B testing, fallback mechanisms, and low-latency aggregation. Stay tuned for the deployment update!