Object Detection: A Computer Vision Project

Welcome to our presentation on Object Detection, a key area in computer vision that involves identifying and locating objects within images or videos. This project was developed by Lakshay Gupta, Yash Verma, and Anshuman Mathur under the guidance of Dr. Yajnaseni Dash at Bennett University.

Our system uses deep learning models such as CNN, SVM, and SVM+MobileNet to process visual data, draw bounding boxes around detected objects, and label them accordingly. Join us as we explore the capabilities of object detection models through practical implementation.

Project Objectives and Applications

Build a Reliable Detection Model

Create an efficient object detection system using traditional ML techniques capable of identifying objects with high accuracy

Real-world Applications

Develop a robust solution applicable to diverse scenarios such as security surveillance, autonomous vehicles, and inventory management

Explore Deep Learning Capabilities

Investigate the effectiveness of various models including CNN, SVM, and hybrid approaches for object detection tasks

Tools and Technologies

Python

Primary programming language used for implementation and model development

Libraries

cv2, scikit-learn, PIL, numpy, flask, tensorflow, os, joblib for data processing and model building

ML Models

CNN, SVM (poly), and SVM+MobileNet for object detection and classification tasks

Deployment

Flask for backend development and HTML for creating a user-friendly interface

Data Collection and Preprocessing

CIFAR-10 Dataset

We utilized the CIFAR-10 dataset from the University of Toronto, which contains images divided into 10 distinct categories, creating a balanced and diverse dataset suitable for image classification.

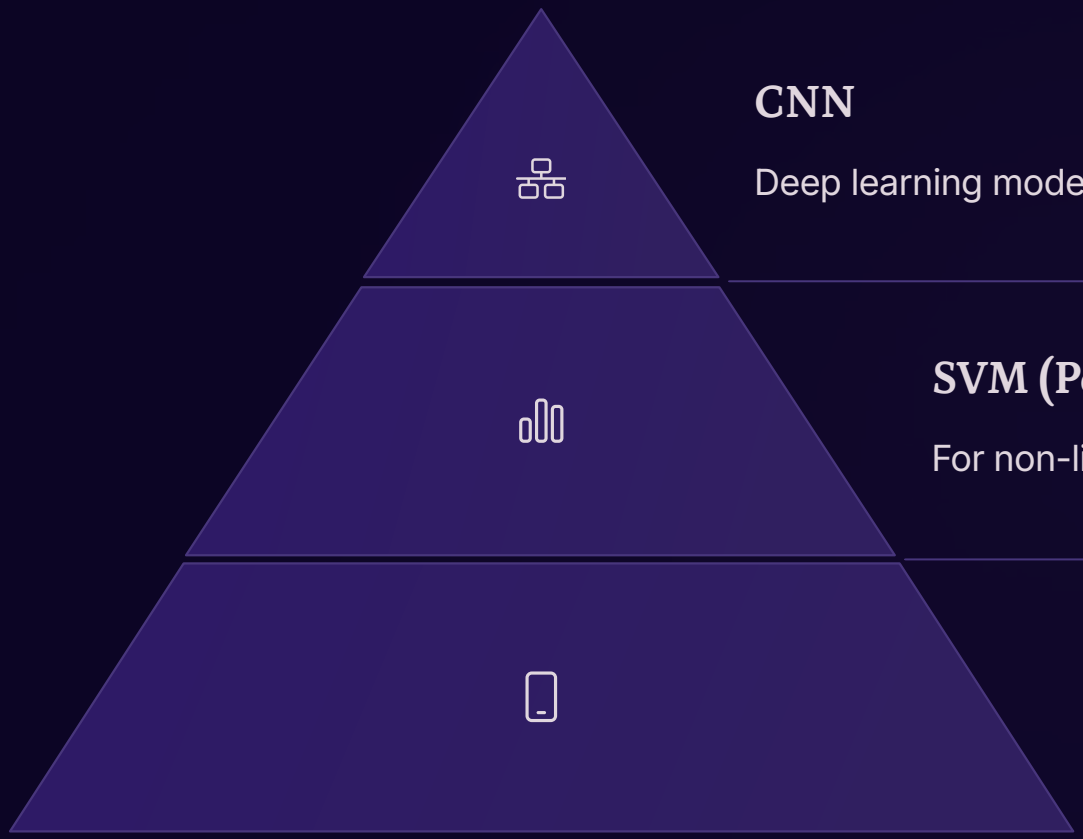
This comprehensive dataset provided the foundation for training our models to recognize various objects accurately.

Preprocessing Techniques

- Resizing images to uniform dimensions
- Normalizing pixel values
- Formatting labels for training
- Data augmentation (flipping, rotation)

These techniques improved model robustness and helped extract meaningful features from raw image data.

Model Development



CNN

Deep learning model for pattern recognition

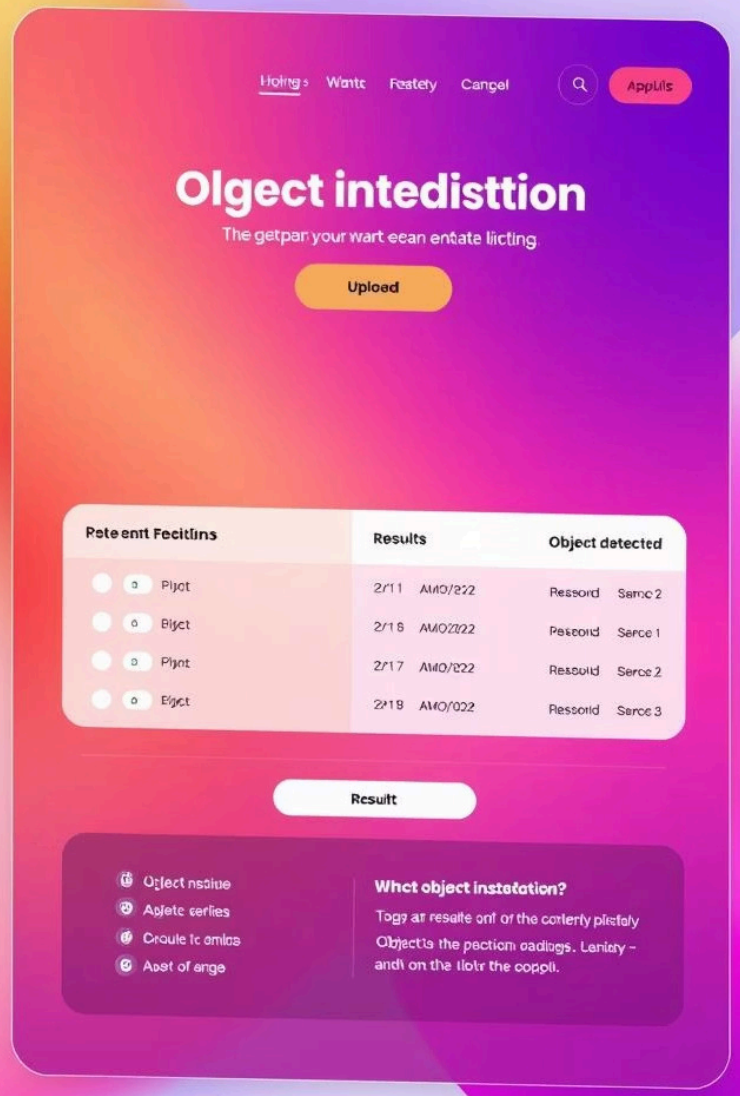
SVM (Polynomial Kernel)

For non-linearly separable data classification

MobileNetV2 + SVM (RBF)

Hybrid approach combining efficient feature extraction with strong classification

Our approach involved experimenting with multiple models to identify the most effective solution. The CNN provided consistent accuracy, while the hybrid MobileNetV2 + SVM approach offered a balance between efficiency and performance.



Deployment and User Experience

Flask Backend Development

We built a robust backend using Flask to load the trained model, handle user inputs, process them efficiently, and return accurate predictions.

HTML Frontend Creation

A simple yet effective frontend was designed using HTML, allowing users to easily upload images and view detection results in an intuitive interface.

Integration and Testing

The system was thoroughly tested to ensure seamless interaction between frontend and backend components, providing a smooth user experience even for non-technical users.

Challenges and Results

Challenges Faced

- Overfitting in SVM(RBF) + MobileNetV2
- Low testing accuracy in SVM(poly) model
- Balancing model complexity with performance

Performance Metrics

- CNN: Highest consistent accuracy
- SVM: Faster but less accurate
- Evaluated using accuracy and validation graphs

User Experience

- Interactive Flask interface
- Understandable for non-technical users
- Efficient real-time detection capabilities

Learnings and Future Scope

1

Key Learnings

Gained in-depth understanding of feature extraction techniques, traditional ML algorithms, and collaborative coding practices. Developed problem-solving skills through continuous model improvement based on test results.

2

Conclusions

The project demonstrated that deep learning approaches, when paired with powerful feature extraction techniques, can deliver reliable results in object classification domains. CNN outperformed other models in terms of consistency and accuracy.

3

Future Scope

Improve speed using quantization and pruning, expand the dataset with more real-world objects, and extend the project to handle multiple object detection from complex scenes.