# Transcript – Object Recognition using Classical Machine Learning

## Slide 1 – Title

Good morning everyone.  
Today I’ll be presenting Object Recognition using Classical Machine Learning, focusing on Feature Engineering with HOG combined with SVM and KNN, using the CIFAR-10 dataset as our benchmark

## Slide 2 – Introduction

Object recognition is a core task in AI applications like autonomous driving, security systems, and biometric identification. The CIFAR-10 dataset includes 60,000 color images of size 32×32 pixels, categorized into 10 classes. In this project, we use Histogram of Oriented Gradients (HOG) as our feature extractor, and compare two classical models — Support Vector Machine and K-Nearest Neighbor. The goal is to evaluate how well traditional methods perform using hand-crafted features.

## Slide 3 – Exploratory Data Analysis (EDA)

CIFAR-10 is balanced: 6,000 images per class. The images are small, colorful, and highly variable within the same class. Because of this variability and low resolution, raw pixels aren’t informative, so feature engineering is needed to extract useful edge and gradient information.

## Slide 4 – Pre-Processing

Pixel values were normalized to 0–1. Each RGB image was converted to grayscale to reduce complexity. Then, HOG was applied to extract texture and edge information, helping models differentiate shapes effectively.

## Slide 5 – Data Partitioning

All 60,000 images were merged and split into 60% training (36,000), 20% validation (12,000), and 20% testing (12,000). Stratified sampling preserved class balance.

## Slide 6 – Model Architecture

Two models were implemented: SVM with a linear kernel (C=1) and KNN with k=5. Both used HOG features (~5,000 dimensions per image).

## Slide 7 – Training Strategy

HOG feature extraction for all 60,000 images took a few minutes. Models were trained on 36,000 samples using scikit-learn. Classical models train once, without epochs or loss curves, and are compared using validation results.

## Slide 8 – Evaluation (SVM)

SVM achieved 52.15% accuracy on validation, performing well on structured classes like trucks and cars, but struggling with animals. Macro F1 ~0.52.

## Slide 9 – Evaluation (KNN)

KNN reached 46.79% accuracy. It had higher recall for simpler shapes but struggled in high-dimensional space, with confusion mainly among animal classes.

## Slide 10 – Test Results

On the test set: SVM = 52.39% accuracy (best for airplanes, trucks); KNN = 47.05% (best for simple shapes). Overall, HOG + SVM outperformed HOG + KNN.

## Slide 11 – Confusion Matrix (SVM)

SVM recognized about half of samples correctly, doing well on digits 0, 1, and 9, but confusing similar ones like 2, 3, 4, and 5. It shows the need for improved features or models.

## Slide 12 – Confusion Matrix (KNN)

KNN achieved 47% accuracy, with best recall for digits 1 (67%), 6 (74%), and 4 (57%). However, class 3 was mostly misclassified as 4 or 6. High confusion due to overlapping pixel patterns.

## Slide 13 – Single Image Test

For individual images, both models gave reasonable predictions, though SVM was more stable. KNN tended to overfit to visual similarities.

## Slide 14 – Strengths & Weaknesses

Strengths: Simple, interpretable, CPU-efficient, and effective for limited data. HOG captures strong edge/orientation features. Weaknesses: Poor scalability, weak high-level abstraction, fixed feature extraction, no training curves.

## Slide 15 – Comparative Insights

SVM generalizes better in high-dimensional space, while KNN suffers from the curse of dimensionality. Trade-off between interpretability and scalability.

## Slide 16 – Conclusion

Classical ML models can perform decently when paired with strong feature engineering like HOG, but they are limited compared to deep learning, which learns features automatically. Future work could involve dimensionality reduction or CNNs for better accuracy.