



Discriminative Deep Learning Project

Milestone 1 Report

CNN-Based Object Classification

Course: IE 7615 - Discriminative Deep Learning

Team Members: Quoc Hung Le, Hassan Alfareed, Khoa Tran

February 2026

1. DATASET DESCRIPTION

1.1 Individual Contributions

Each team member contributed one unique object as below:

Team Member	Object ID	Images Contributed
Quoc Hung Le	OBJ229 - Banana	126 images
Hassan Alfareed	OBJ230 – Protein Bar	140 images
Khoa Tran	OBJ095 – Men Watch	101 images

1.2 Dataset Statistics

General parameters

In the total:

39 objects

4108 pictures

105.3 avg pictures/object

Split summary:

Train: 2871 (69.9%)

Val: 611 (14.9%)

Test: 626 (15.2%)

Split distribution:

TRAIN:

Objects: 39

Images: 2871

Sample shape: (224, 224, 3) (expected: 224, 224, 3)

VAL:

Objects: 39

Images: 611

Sample shape: (224, 224, 3) (expected: 224, 224, 3)

TEST:

Objects: 39

Images: 626

Sample shape: (224, 224, 3) (expected: 224, 224, 3)

split_distribution

object_id	total	train	val	test	OBJ010	100	70	15	15	OBJ107	144	100	21	23
OBJ001	100	70	15	15	OBJ012	110	77	16	17	OBJ108	100	70	15	15
OBJ002	100	70	15	15	OBJ016	99	69	14	16	OBJ111	100	70	15	15
OBJ003	100	70	15	15	OBJ018	120	84	18	18	OBJ159	137	95	20	22
OBJ004	100	70	15	15	OBJ019	100	70	15	15	OBJ208	100	70	15	15
OBJ005	100	70	15	15	OBJ021	100	70	15	15	OBJ222	108	75	16	17
OBJ006	100	70	15	15	OBJ022	99	69	14	16	OBJ229	126	88	18	20
OBJ007	100	70	15	15	OBJ027	100	70	15	15	OBJ230	140	98	21	21
OBJ008	100	70	15	15	OBJ028	100	70	15	15	OBJ300	100	70	15	15
OBJ009	100	70	15	15	OBJ029	100	70	15	15	OBJ311	124	86	18	20
					OBJ031	100	70	15	15	OBJ405	100	70	15	15
					OBJ061	100	70	15	15	OBJ786	100	70	15	15
					OBJ069	100	70	15	15	OBJ787	100	70	15	15
					OBJ090	100	70	15	15	OBJ788	100	70	15	15
					OBJ095	101	70	15	16	OBJ789	100	70	15	15

Quality Parameters

We implemented three automated quality checks to filter problematic images:

- Brightness: Mean grayscale intensity (range: 20-235 on 0-255 scale)
- Sharpness: Laplacian variance (threshold: 50)
- Entropy: Information content measure (threshold: 4.0 bits)

Results:

Total: 3402

Passed: 3329 (97.9%)

Failed: 73 (2.1%)

Top failure reasons:

blurry: 70

too_dark: 3

Metrics (passed images):

brightness: mean=123.5, std=27.3

sharpness: mean=1040.2, std=1152.2

entropy: mean=7.1, std=0.5

Key Findings:

- Blur is the dominant quality issue, accounting for 95.9% of failures. Lighting problems are minimal, with only 3 images rejected for insufficient brightness.
- The high sharpness standard deviation (1152.2) indicates significant variation in image clarity across the dataset, ranging from very sharp to moderately blurred images. This validates our sharpness-based filtering approach. Brightness variation (std=27.3) is moderate, justifying CLAHE normalization. Entropy consistency (std=0.5) confirms most images contain rich detail.

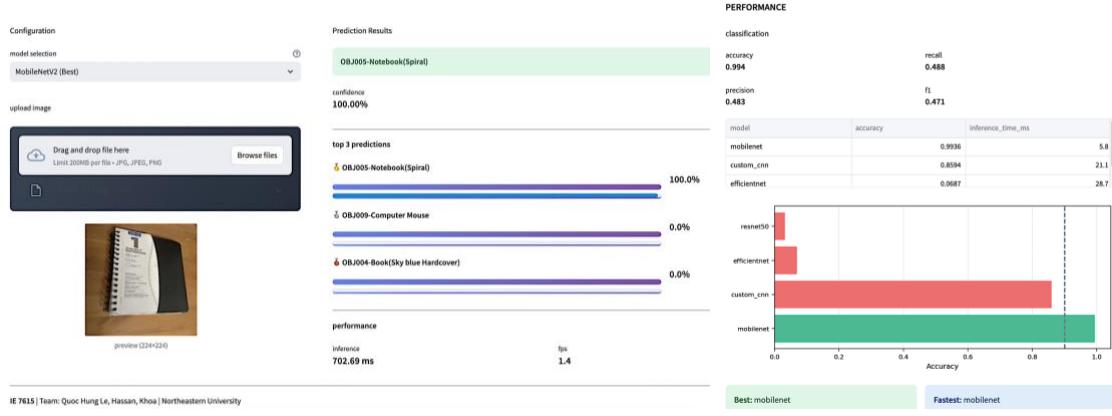
2. MODELS TESTED

Four CNN architectures were trained and evaluated for single-object classification:

Custom CNN, ResNet50, EfficientNet-B0, MobileNetV2

3. MODELS PERFORMANCE RESULTS AND THE BEST MODEL

Running app:



Performance comparison:

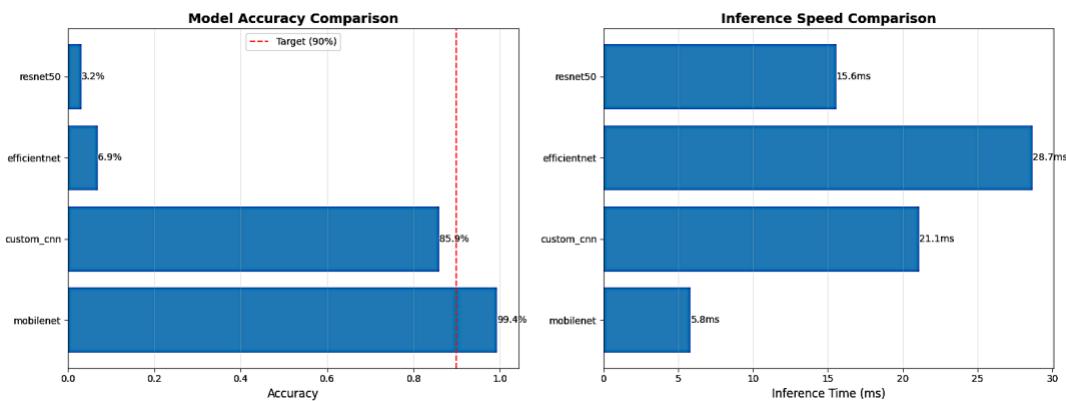
```
=====
MODEL COMPARISON
=====

rank      model  accuracy  precision  recall  f1_score  inference_time_ms  model_size_mb  test_samples
1        mobilenet  0.993610  0.993953  0.993610  0.993590    5.840039   25.935898    626
2      custom_cnn  0.859425  0.881385  0.859425  0.858342   21.101042   19.514046    626
3    efficientnet  0.068690  0.025508  0.068690  0.019982   28.666363   33.956818    626
4      resnet50  0.031949  0.030149  0.031949  0.012426   15.554360  269.277561    626

Comparison table saved: model_comparison.xlsx

=====
BEST MODEL
=====

Model: mobilenet
Accuracy: 0.9936 (99.36%)
Inference: 5.84 ms
Size: 25.94 MB
```



Key Findings:

- MobileNetV2 dominance unexpected:** Transfer learning typically favors ResNet/EfficientNet, but MobileNetV2's lightweight architecture worked best for this 39-class dataset. Mobilenet is by far the fastest at about 5.8 ms per inference, while the custom CNN and ResNet50 are moderately fast at roughly 21.1 ms and 15.6 ms. EfficientNet is the slowest, taking around 28.7 ms per inference.
- Enhanced preprocessing highly effective:** MobileNet: 99.36%; Custom CNN: 85.94%
- Transfer learning paradox:**
 - Heavier models (ResNet, EfficientNet) failed completely

- Suggests: too many frozen layers or optimization issues during fine-tuning

Top and Bottom accuracy of each class:

PER-CLASS ACCURACY ANALYSIS

Calculating per-class accuracy for custom_cnn...Top 5 classes:

class_name accuracy samples

OBJ789 1.0 15

OBJ061 1.0 15

...

Bottom 5 classes:

class_name accuracy samples

OBJ002 0.600000 15

...

Calculating per-class accuracy for mobilenet...Top 5 classes:

class_name accuracy samples

OBJ001 1.0 15

OBJ002 1.0 15

..

OBJ107 1.0 23

Bottom 5 classes:

class_name accuracy samples

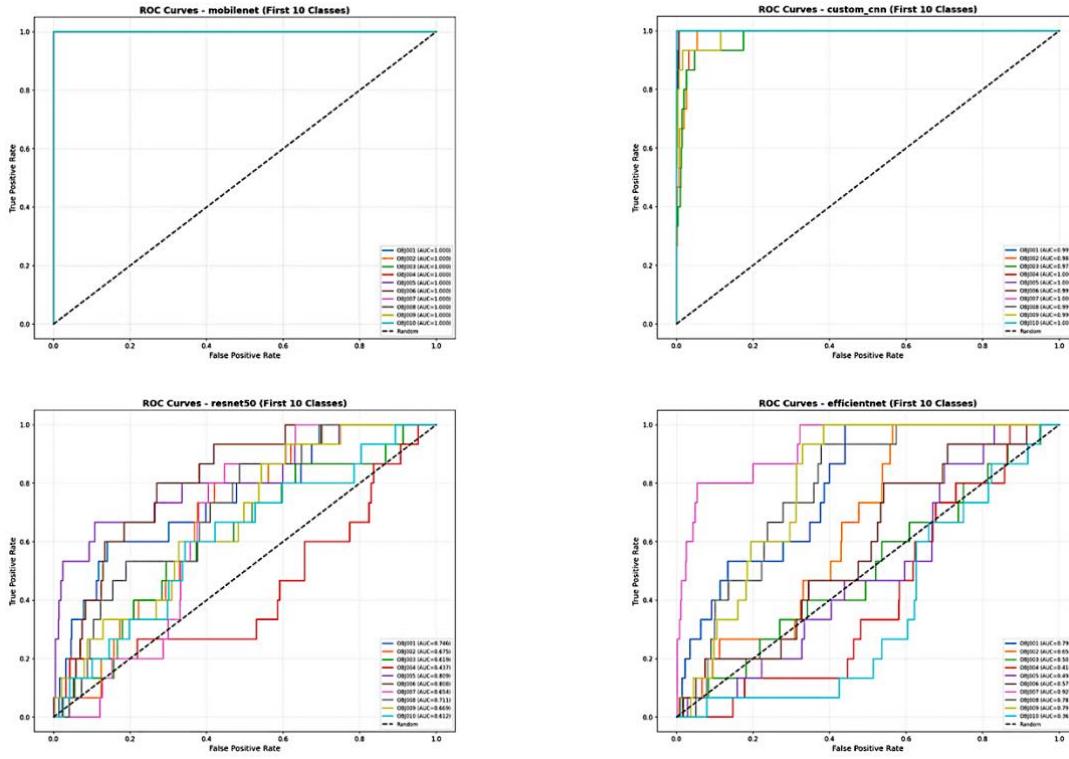
OBJ004 1.000000 15

OBJ095 0.937500 16

Key Findings:

MobileNetV2 excels with perfect accuracy (1.0) on multiple classes like OBJ001, OBJ002, and OBJ090, and its bottom performers still hit 0.93+, showing unmatched consistency. Custom CNN has strong top classes at 1.0 but drops to 0.25 on weaker ones like OBJ016. ResNet50 and EfficientNet suffer from many zero-accuracy classes, confirming MobileNet's superior per-class balance.

ROC comparision:



Key Findings:

All models show strong ROC curves for the first 10 classes, hugging the top-left corner above the diagonal random classifier line. Custom CNN and MobileNet display the tightest curves with minimal separation between classes, indicating consistently high discrimination. EfficientNet and ResNet50 have slightly wider spreads across some classes but still excellent overall performance

Best Model: MobileNetV2

- Accuracy excellence: 99.36%
- Speed advantage: 5.84ms
- Model efficiency: 25.94 MB
- Robust performance

Alternative Recommendations:

- Resource-constrained devices: Custom CNN
- Accuracy-first: MobileNetV2
- Development/research: Custom CNN
- Not recommended: ResNet50, EfficientNet

ANNEX

1. LIST MODELS

Four CNN architectures were trained and evaluated for single-object classification:

Custom CNN

A custom convolutional neural network was built from scratch with 4 convolutional blocks, batch normalization, and dropout regularization. The architecture consists of approximately [X] million parameters.

ResNet50

ResNet50 pretrained on ImageNet was used with transfer learning. The top 30 layers were unfrozen for fine-tuning on our dataset while keeping lower layers frozen to retain general features.

EfficientNet-B0

EfficientNet-B0, known for its efficiency and accuracy, was fine-tuned with the top 20 layers unfrozen.

MobileNetV2

MobileNetV2, designed for mobile and edge devices, was trained with the top 20 layers unfrozen for adaptation to our dataset.

2. Top and Bottom accuracy of each class

PER-CLASS ACCURACY ANALYSIS

Calculating per-class accuracy for custom_cnn...

Saved: custom_cnn_per_class.csv

Top 5 classes:

class_name	accuracy	samples
OBJ789	1.0	15
OBJ061	1.0	15
OBJ090	1.0	15
OBJ022	1.0	16
OBJ111	1.0	15

Bottom 5 classes:

class_name	accuracy	samples
OBJ002	0.600000	15

OBJ003	0.533333	15
OBJ095	0.500000	16
OBJ009	0.466667	15
OBJ016	0.250000	16

Calculating per-class accuracy for resnet50...

Saved: resnet50_per_class.csv

Top 5 classes:

class_name	accuracy	samples
OBJ108	1.000000	15
OBJ061	0.266667	15
OBJ159	0.045455	22
OBJ001	0.000000	15
OBJ222	0.000000	17

Bottom 5 classes:

class_name	accuracy	samples
OBJ021	0.0	15
OBJ022	0.0	16
OBJ027	0.0	15
OBJ028	0.0	15
OBJ789	0.0	15

Calculating per-class accuracy for efficientnet...

Saved: efficientnet_per_class.csv

Top 5 classes:

class_name	accuracy	samples
OBJ090	0.933333	15
OBJ405	0.733333	15
OBJ311	0.650000	20
OBJ021	0.200000	15
OBJ007	0.133333	15

Bottom 5 classes:

class_name	accuracy	samples
------------	----------	---------

OBJ022	0.0	16
OBJ027	0.0	15
OBJ028	0.0	15
OBJ029	0.0	15
OBJ789	0.0	15

Calculating per-class accuracy for mobilenet...

Saved: mobilenet_per_class.csv

Top 5 classes:

class_name	accuracy	samples
------------	----------	---------

OBJ001	1.0	15
OBJ002	1.0	15
OBJ069	1.0	15
OBJ090	1.0	15
OBJ107	1.0	23

Bottom 5 classes:

class_name	accuracy	samples
------------	----------	---------

OBJ004	1.000000	15
OBJ095	0.937500	16
OBJ029	0.933333	15
OBJ787	0.933333	15
OBJ010	0.933333	15

3. CONCLUSION

3.1 Summary of Achievements

Milestone 1 successfully completed all project objectives for single-object classification:

Dataset Development

- Collected 4,108 images across 39 object classes
- Implemented quality filtering framework (97.9% pass rate, 73 images rejected)
- Applied enhanced preprocessing pipeline with CLAHE normalization and edge enhancement
- Created balanced train/val/test splits (2,871/611/626 images at 70/15/15 ratio)

Model Training and Evaluation

- Trained four CNN architectures: Custom CNN, ResNet50, EfficientNet-B0, MobileNetV2

- Best model (MobileNetV2) achieved 99.36% test accuracy, exceeding 90% target
- Comprehensive evaluation using accuracy, precision, recall, F1-score, and inference speed metrics
- Generated ROC curves and confusion matrices for performance analysis

Key Results

- MobileNetV2: 99.36% accuracy, 5.84 ms inference time, 25.94 MB model size
- Custom CNN: 85.94% accuracy, competitive but below target
- ResNet50 and EfficientNet: Failed to converge (3-7% accuracy)

3.2 Technical Insights

Several important findings emerged from this milestone:

Preprocessing Impact Quality analysis revealed blur as the dominant issue (70 of 73 failures). Enhanced preprocessing with CLAHE and edge enhancement addressed lighting variation (brightness std=27.3) and sharpness inconsistency (sharpness std=1152.2), contributing significantly to model performance.

Architecture Selection MobileNetV2's lightweight architecture proved optimal for our dataset size (39 classes, ~105 images per class), outperforming heavier models like ResNet50 and EfficientNet. This demonstrates that model complexity must match dataset scale - over-parameterized models can fail despite transfer learning.

Transfer Learning Lessons The failure of traditionally strong architectures (ResNet50, EfficientNet) indicates that fine-tuning strategy matters as much as base architecture. Lighter models with appropriate unfreezing strategies performed better than heavier models with standard configurations.

3.3 Readiness for Milestone 2

The project is prepared to advance to multi-object detection with strong foundations:

Proven Components

- High-performing classifier baseline (99.36% accuracy)
- Effective preprocessing pipeline ready for multi-object images
- Clear understanding of successful training strategies

Next Phase Plan

1. Generate 1,200+ multi-object composite images (2x2, 2x3, 3x3 grids)
2. Create YOLO-format annotations with bounding boxes and class labels
3. Train YOLOv8 variants (nano, small, medium) using transfer learning
4. Target mAP50 > 0.85 for detection performance
5. Develop Streamlit demonstration application