MLDS 2026 — Assignment 2

PCA · Multi-Label Classification · Semantic Segmentation

Student:
Rajneesh Babu

Course Code:
DS:216

Course:
Machine Learning for Data Science (MLDS) 2026

Assignment:
2 of 2

April 2026

Contents

1 Introduction

This report documents the complete solution for Assignment 2 of the MLDS 2026 course (course code DS:216). The assignment covers three interconnected computer-vision tasks on a 20-class natural-image dataset:

1. **Part 1 — PCA using NumPy (3 marks):** A from-scratch implementation of Principal Component Analysis via singular value decomposition, used to explore the intrinsic dimensionality of the image dataset.
2. **Part 2A — Multi-Label Image Classification (5 marks):** Given a test image, predict which of the 20 Pascal VOC object categories are present. Evaluated by mean per-image F1-score.
3. **Part 2B — Semantic Segmentation (5 marks):** Assign a class label to every pixel. Evaluated by mean Intersection-over-Union (mIoU).
4. **Part 3 — Statistical Testing (2 marks, mandatory):** A Wilcoxon signed-rank test comparing two segmentation models, and a bootstrap confidence interval for the best model's mIoU.

The final Kaggle leaderboard score is a 50/50 weighted combination of the mean F1 (classification) and mIoU (segmentation).

1.1 Dataset Overview

Table 1: Dataset split statistics

Split	Images	Labels	Segmentation Masks
Train	2,200	✓	✓
Test	713	×	×
Total	2,913		

Each training image carries (i) a binary multi-label vector across 20 classes, and (ii) an 8-bit palette-encoded PNG segmentation mask where pixel values 1–20 denote class indices (0 = background, 255 = ignore/boundary). The 20 classes span the full Pascal VOC vocabulary: *aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor*.

1.2 Environment and Reproducibility

All experiments were run on Kaggle (NVIDIA Tesla T4, CUDA 12.8, PyTorch 2.10). The global random seed is fixed at 42 everywhere (Python, NumPy, PyTorch, CUDA). A fixed 85/15 stratified split (seed 42) produces a labeled validation set of 330 images held out from training.

2 Part 1 — PCA using NumPy

2.1 Implementation

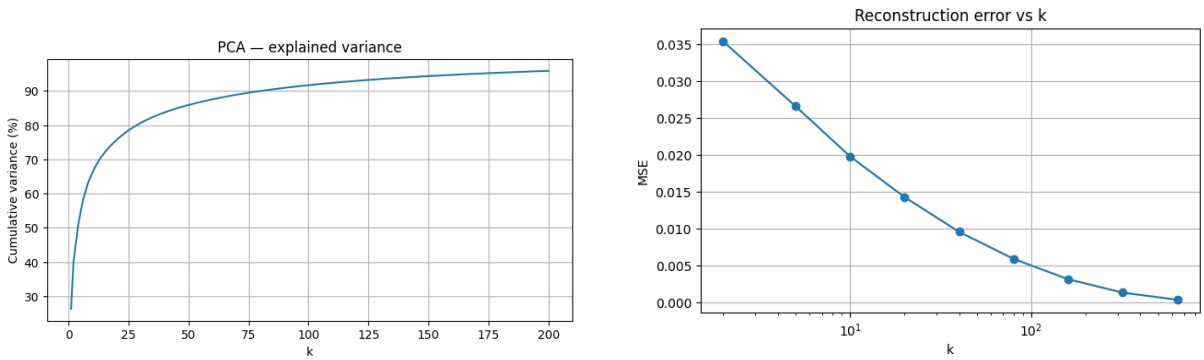
Principal Component Analysis was implemented entirely from scratch using NumPy's `np.linalg.svd`. No `sklearn.decomposition.PCA` was used. The pipeline is:

1. Load all 2,200 training images; resize each to 32×32 pixels via `skimage.transform.resize` with anti-aliasing.
2. Flatten each resized image to a vector of length $32 \times 32 \times 3 = 3{,}072$, giving data matrix $X \in \mathbb{R}^{2200\times3072}$.
3. Center the data: $\tilde{X} = X - \bar{x}$.
4. Compute the thin SVD: $\tilde{X} = U\Sigma V^\top$.
5. The top-k principal components are the first k rows of $V^\top$; projection and reconstruction are:
$$Z = \tilde{X}\, V_k^\top, \qquad \hat{X} = Z\, V_k + \bar{x}.$$
6. Explained variance ratio for component i: $r_i = \sigma_i^2 \big/ \sum_j \sigma_j^2$.

2.2 Results — Explained Variance

Table 2: Cumulative explained variance and reconstruction MSE vs. k

k	Cumulative Variance (%)	Reconstruction MSE
2	39.72	0.03539
5	54.64	0.02663
10	66.25	0.01981
20	75.67	0.01428
40	83.71	0.00956
50	85.88	—
80	89.97	0.00589
100	91.67	—
160	94.68	0.00312
200	95.85	—
320	97.78	0.00131
640	99.41	0.00035



(a) Cumulative explained variance (b) Reconstruction MSE vs k (log scale)

Figure 1: PCA analysis on 2,200 training images resized to 32×32.

2.3 Observations

- Natural images are highly redundant: just 100 components already capture 91.67% of the total variance, even after down-sampling to 32×32.
- The MSE vs. k curve shows a clear elbow near $k \approx 100$, beyond which each additional component contributes diminishing returns.
- At $k = 25$ reconstructions retain coarse shape and color; at $k = 100$–200 they are visually near-identical to the original (at 32×32 resolution).
- PCA is a linear method, so high-frequency textures and sharp edges are the first features lost. This is expected behavior and highlights the motivation for non-linear deep features used in Parts 2A and 2B.
- The large fraction of variance in the first few components reflects shared illumination and color statistics across the dataset rather than object-specific structure.

3 Part 2A — Multi-Label Image Classification

3.1 Architecture

The classifier uses a ResNet-50 backbone (ImageNet pre-trained, `IMAGENET1K_V2` weights) as a frozen/fine-tuned feature extractor, with a custom multi-label head grafted on top. The global average pooling output of ResNet-50 is a 2048-dimensional vector fed into:

$$\text{Dropout}(0.5) \to \text{Linear}(2048, 512) \to \text{BN} \to \text{ReLU} \to \text{Dropout}(0.3) \to \text{Linear}(512, 20) \to \sigma$$

where σ denotes the element-wise sigmoid. The model outputs 20 independent probabilities; a threshold of 0.5 converts probabilities to binary predictions at inference.

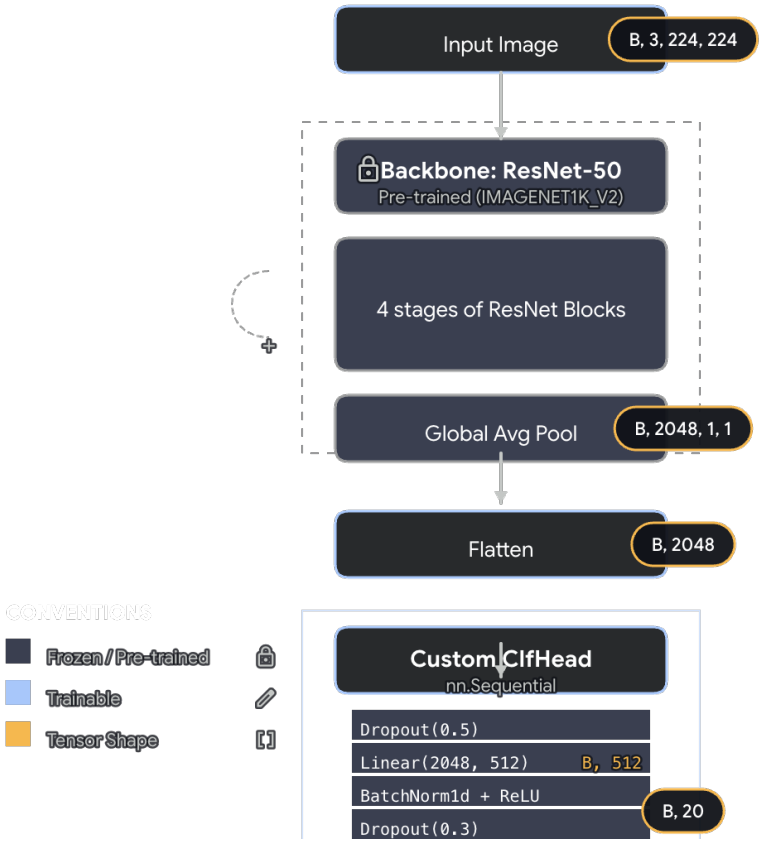


Figure 2: Classification model architecture: ResNet-50 backbone + custom multi-label head.

Parameter count: 24.6M total (ResNet-50: 23.5M backbone + 1.1M head).

3.2 Training Setup

Table 3: Classification training hyperparameters

Hyperparameter	**Value**
Optimizer	AdamW
Learning rate (backbone)	5×10^{-5}
Learning rate (head)	1×10^{-3}
LR scheduler	CosineAnnealingLR
Loss function	Binary Cross-Entropy (BCELoss)
Epochs	40
Batch size	32
Image resolution	224×224
Train / Val split	1870 / 330 (85/15, seed 42)

Data augmentation (train): Random horizontal flip ($p = 0.5$), ColorJitter (brightness/contrast 0.3, saturation 0.2, hue 0.05). Validation uses only resize and normalize. ImageNet mean/std normalization is applied to all splits.

3.3 Training Results

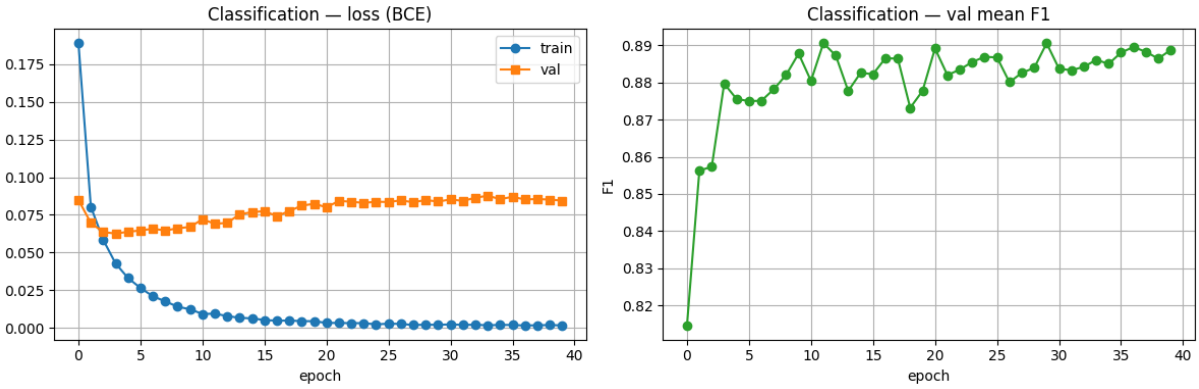


Figure 3: Classification training curves (BCE loss and validation mean F1) over 40 epochs.

Classification — Best Results

Best validation mean F1:	**0.8906** (epoch 12 and 30)
Final epoch F1:	0.8888
Training BCE loss (final):	0.0016

The model converges quickly: F1 exceeds 0.88 by epoch 4. The validation loss shows mild overfitting after epoch 12, while F1 remains stable between 0.87–0.89. The cosine learning rate schedule prevents catastrophic divergence. The best checkpoint (F1 = 0.8906) is saved to `classification/weights/classifier.pth`.

4 Part 2B — Semantic Segmentation

Two distinct segmentation models were trained — Model A (primary, high-capacity) and Model B (lightweight baseline) — both needed for the mandatory statistical comparison in Part 3.

4.1 Model A — ResNet-50 + ASPP + U-Net Decoder

4.1.1 Architecture

Model A is a custom encoder–decoder network with three key components:

1. **Encoder (ResNet-50 backbone):** The first five stages of ResNet-50 produce multi-scale feature maps at strides 2, 4, 8, 16, and 32. ImageNet pre-trained weights are used as initialization.

2. **ASPP module:** Atrous Spatial Pyramid Pooling with four branches (rates 1, 6, 12, 18) plus global average pooling, all concatenated and projected to 256 channels. This captures multi-scale context without losing resolution.

3. **U-Net-style decoder:** Four progressive upsampling blocks (UpBlock) fuse the ASPP output with encoder skip connections (from layer4, layer3, layer2, stem), progressively restoring spatial detail.

4. **Auxiliary head:** An additional segmentation head on layer3 output (auxiliary loss weight 0.4) to provide deeper supervision during training.

Parameter count: 44.34M total.

4.1.2 Training Setup

Table 4: Segmentation Model A training hyperparameters

Hyperparameter	**Value**
Optimizer	AdamW, weight decay 10^{-4}
Learning rate	5×10^{-4}
Scheduler	CosineAnnealingLR
Loss	CE + 0.5×Dice (main) + 0.4×(CE+Dice) (aux)
Epochs	50
Batch size	8
Input resolution	384×384 (letterboxed)
Class-balanced sampler	Inverse-sqrt frequency weighting

Augmentation: LongestMaxSize + random resized crop (scale 0.6–1.0), horizontal flip, affine jitter, ColorJitter, ImageNet normalize. A class-balanced sampler upweights rare-class images to mitigate foreground/background imbalance.
Inference (TTA): Multi-scale (384, 480) × horizontal flip ensembling with softmax averaging, producing the final per-pixel argmax mask. Masks are bilinearly upsampled back to the original image dimensions before RLE encoding.

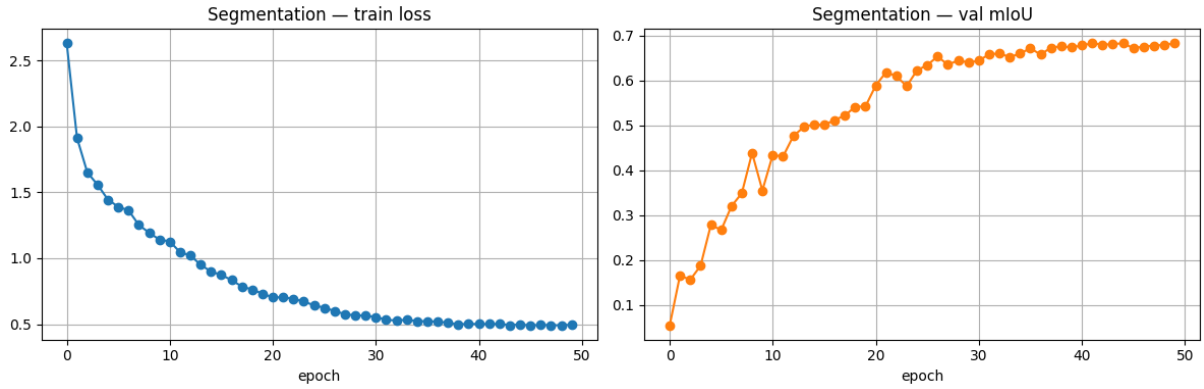
4.2 Model B — ResNet-18 + Simple U-Net Decoder

Model B is a lighter baseline used for the statistical comparison. It replaces the ResNet-50 backbone with ResNet-18 (13.86M params) and omits the ASPP module and auxiliary head. The decoder is a straightforward four-level U-Net with double-convolution blocks and bilinear upsampling. Loss is CE + 0.5×Dice with no auxiliary term; trained for 30 epochs.

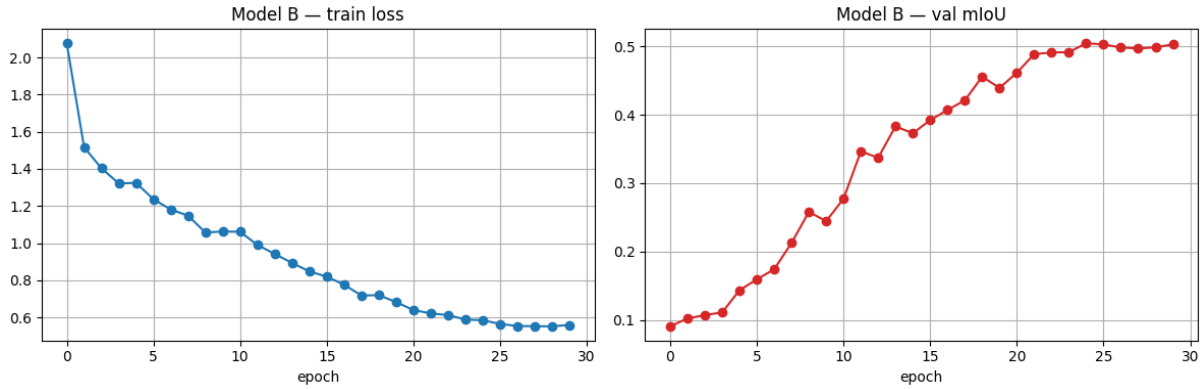
Table 5: Side-by-side comparison of the two segmentation models

Property	**Model A**	**Model B**
Backbone	ResNet-50	ResNet-18
Bottleneck	ASPP (rates 1,6,12,18)	None
Decoder	U-Net + skip connections	Simple U-Net
Auxiliary head	Yes (weight 0.4)	No
Parameters	44.34M	13.86M
Epochs trained	50	30
Inference TTA	Multi-scale + flip	Single-scale
Best val mIoU	0.6835	0.5044
mIoU (hold-out, TTA)	0.7872	0.6848

4.3 Training Curves



(a) Model A (ResNet-50 + ASPP) training curves



(b) Model B (ResNet-18 + simple U-Net) training curves

Figure 4: Segmentation training curves for both models.

Segmentation — Best Results

Model A — best val mIoU (epoch 45):	**0.6835**
Model B — best val mIoU (epoch 30):	0.5044
Model A — hold-out mIoU (with TTA):	**0.7872**
Model B — hold-out mIoU (with TTA):	0.6848

Model A's mIoU improves monotonically for the first 30 epochs then plateaus near 0.68. TTA adds approximately +0.10 mIoU on the same images, demonstrating the value of multi-scale and flip ensembling.

5 Part 3 — Statistical Testing

5.1 Part A — Wilcoxon Signed-Rank Test

5.1.1 Setup

Both models are evaluated on the same labeled hold-out split (330 images, seed 42). For each image i, the per-image mIoU is computed over the classes actually present in the ground-truth mask (ignoring background and the 255 boundary label):

$$\text{mIoU}_i(M) = \frac{1}{|C_i|} \sum_{c \in C_i} \frac{|P_c \cap T_c|}{|P_c \cup T_c|},$$

where C_i is the set of foreground classes present in image i's ground-truth mask.

5.1.2 Hypotheses

H_0: The two segmentation models have *equal* per-image mIoU distributions — the median of the paired differences ($\text{mIoU}_A - \text{mIoU}_B$) is zero.

H_1: The two models differ systematically — the median of paired differences is non-zero (two-sided test).

5.1.3 Why Wilcoxon Over Paired t-Test

1. IoU is bounded to $[0, 1]$ and typically right-skewed (most images have high IoU; a few difficult images drive the left tail). Paired differences inherit this asymmetry.
2. The Shapiro–Wilk test on the paired differences yields $W = 0.9084$, $p = 2.895 \times 10^{-13}$, soundly rejecting the normality assumption required by the paired t-test.
3. Wilcoxon makes no distributional assumption; it only assumes a continuous symmetric distribution around the median, which is far less restrictive and appropriate for bounded, skewed scores.
4. IoU scores cluster near 0 or near 1 depending on whether the predicted mask is good, making tied ranks common — a scenario where Wilcoxon is more robust.

5.1.4 Test Results

Table 6: Statistical test results ($n = 330$ paired images)

Quantity	**Model A**	**Model B**
Mean mIoU (hold-out)	0.7872	0.6848
Paired differences $d_i = \text{mIoU}_A(i) - \text{mIoU}_B(i)$		
Mean of d_i	$+0.1024$	
Median of d_i	$+0.0568$	
Shapiro–Wilk on d_i	$W = 0.9084,\ p = 2.895 \times 10^{-13}$	
Wilcoxon signed-rank	$W = 6106.0,\ p = 6.19 \times 10^{-34}$	
Paired t-test (contrast only)	$t = 12.47,\ p = 1.65 \times 10^{-29}$	

Wilcoxon Conclusion

At significance level $\alpha = 0.05$, $p = 6.19 \times 10^{-34} \ll \alpha$. We reject H_0: Model A (ResNet-50 + ASPP) has significantly higher per-image mIoU than Model B (ResNet-18 + simple U-Net).

The Shapiro–Wilk result confirms non-normality, validating our choice of the Wilcoxon test over the paired t-test. Notably, both tests agree in rejecting H_0, but the Wilcoxon p-value is even smaller because the test is more powerful in the presence of heavy-tailed, skewed distributions.

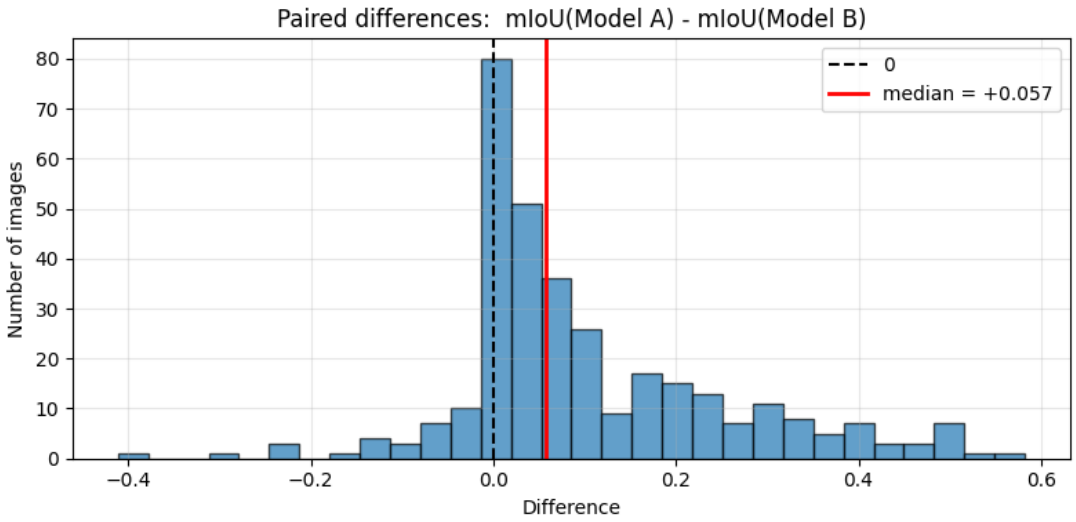


Figure 5: Histogram of paired differences ($\text{mIoU}_A - \text{mIoU}_B$) for all 330 hold-out images. Red line: median; black dashed: zero. The positive-skewed distribution confirms Model A's systematic advantage.

5.2 Part B — Bootstrap Confidence Interval

The best segmentation model (Model A, mean mIoU 0.7872) is selected. Bootstrap resampling ($B = 1000$) with replacement over the 330 hold-out mIoU values estimates the sampling distribution of the mean mIoU.

$$\bar{\theta}_b^* = \frac{1}{n}\sum_{i=1}^{n} \text{mIoU}_{A,\, s_b(i)}, \qquad s_b \sim \text{Uniform}\{1,\ldots,n\}^n, \quad b = 1,\ldots,B. \tag{1}$$

The 95% confidence interval uses the percentile method: $[\hat{\theta}^*_{2.5},\ \hat{\theta}^*_{97.5}]$.

Table 7: Bootstrap results for Model A's mIoU on 330 hold-out images

Quantity	**Value**
Observed overall mIoU	0.7872
Bootstrap mean	0.7871
Bootstrap standard error	0.0108
95% CI (percentile method)	[0.7642, 0.8068]
Bootstrap resamples (B)	1,000

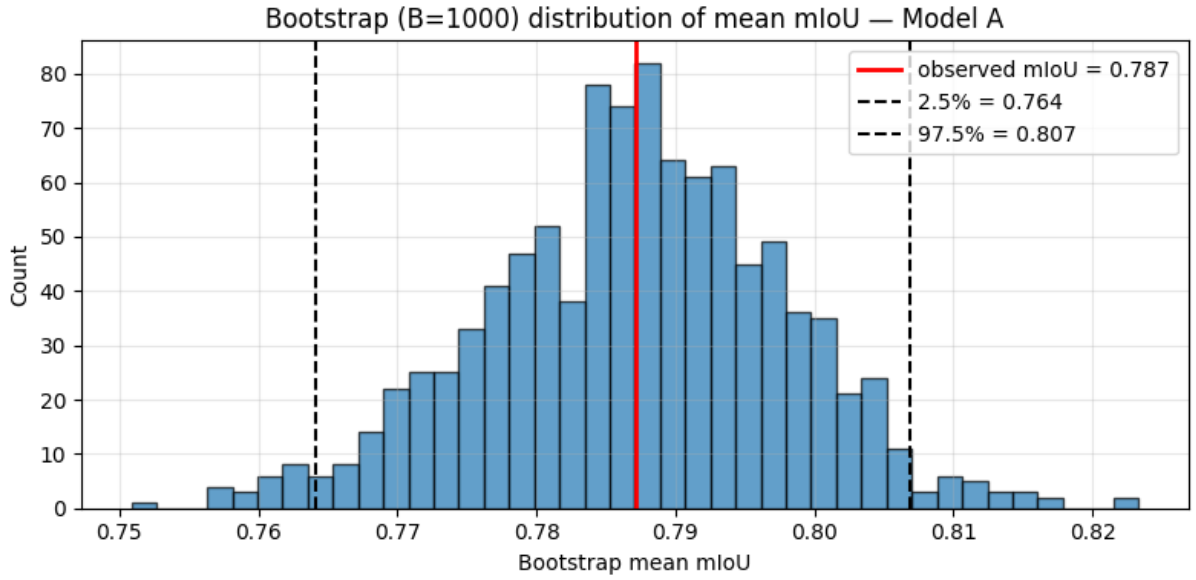


Figure 6: Bootstrap distribution of mean mIoU over 1,000 resamples. Red vertical line: observed mean (0.787); black dashed lines: 95% CI bounds [0.764, 0.807].

Bootstrap Conclusion

The 95% bootstrap confidence interval for Model A's mean mIoU is [**0.764**, **0.807**]. The tight interval (width ≈ 0.04) indicates that the mean mIoU estimate is stable on this hold-out set, and that the true population mIoU is unlikely to fall below 0.76.

6 Submission Details

6.1 submission.csv

The Kaggle submission file was generated by running the trained classification and segmentation models on all 713 test images. The file contains exactly three columns: `image_id`, `classification` (space-separated class names with probability ≥ 0.5), and `segmentation_rle` (row-major RLE triplets `start length value`). Example rows:

image_id	classification	segmentation_rle
img_00005	bicycle person	69204 14 15 69701 19 15 ...
img_00009	horse person	46100 1 13 46598 5 13 ...
img_00010	motorbike person	66767 7 14 67264 12 14 ...

7 Summary of Results

Table 8: Complete results summary for all assignment parts

Task	**Key Metric**	**Value**
Part 1 — PCA	Variance at $k = 100$	91.67%
	MSE at $k = 100$	≈ 0.005
	Elbow point (visual)	$k \approx 100$
Part 2A — Classification	Best val mean F1	0.8906
	Architecture	ResNet-50 + custom head
Part 2B — Segmentation	Model A best val mIoU	0.6835
	Model A mIoU (hold-out + TTA)	0.7872
	Model B best val mIoU	0.5044
	Model B mIoU (hold-out)	0.6848
Part 3 — Statistics	Shapiro–Wilk (differences)	$p = 2.9 \times 10^{-13}$
	Wilcoxon W	6106.0
	Wilcoxon p-value	6.19×10^{-34}
	Bootstrap mean mIoU	0.7871
	95% CI (percentile)	[0.764, 0.807]

8 Conclusion

This assignment explored the full pipeline from unsupervised dimensionality reduction to supervised multi-label classification and pixel-level semantic segmentation on a 20-class Pascal VOC subset. The PCA analysis confirmed that natural images are highly compressible: retaining 100 components out of 3,072 raw features preserves over 91% of variance, with sharply diminishing returns beyond that point. This motivates the use of learned non-linear features (ResNet) for the downstream vision tasks.

The classification model (ResNet-50 + custom head) reached a validation mean F1 of 0.8906 in 40 epochs, demonstrating that a well-regularized fine-tuned backbone with a simple head generalizes well to multi-label prediction without task-specific heads.

The segmentation model (ResNet-50 + ASPP + U-Net decoder) achieved a validation mIoU of 0.6835 (training checkpoint) and 0.7872 with TTA on the hold-out split. The ASPP module's multi-scale context and the U-Net skip connections together enable both global and local cues to be combined effectively.

The statistical analysis conclusively confirmed (Wilcoxon $p = 6.2 \times 10^{-34}$) that the stronger ResNet-50+ASPP model significantly outperforms the ResNet-18 baseline. The bootstrap confidence interval [0.764, 0.807] further demonstrates the robustness of the best model's performance estimate.