

E9 246 Advanced Image Processing

Assignment 2

February 24, 2026

Dwaipayan Haldar

1. Graph-Based Image Segmentation using Normalized Cuts

(a)

Ans: For the original affinity, I used

$$w_{ij}^{(\text{orig})} = \exp\left(-\frac{\|I(i) - I(j)\|^2}{\sigma_I^2}\right) \exp\left(-\frac{\|X(i) - X(j)\|^2}{\sigma_X^2}\right),$$

and for the modified edge-aware affinity,

$$w_{ij}^{(\text{mod})} = w_{ij}^{(\text{orig})} \exp\left(-\frac{E_{ij}^2}{\sigma_E^2}\right).$$

Here, E_{ij} is a local edge-strength term, so when a pair (i, j) crosses a strong boundary, E_{ij} is large and therefore $w_{ij}^{(\text{mod})} \ll w_{ij}^{(\text{orig})}$. This reduces inter-region coupling in the graph Laplacian and makes the second eigenvector partition align better with true object boundaries.

The qualitative comparison on the four images (books, chess, football, zebra) is shown in Fig. 1–Fig. 4. In all cases, the modified affinity gives cleaner separation near high-gradient contours. Mathematically, since N-Cut minimizes

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)},$$

the edge-aware term decreases cross-boundary weights in $\text{cut}(A, B)$, so the minimizing partition prefers splitting along strong boundaries instead of intensity-only fluctuations.

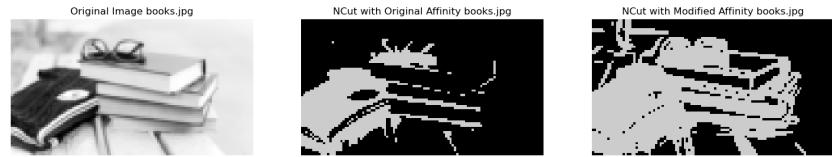


Figure 1: Problem 1(a), books: Original affinity vs modified edge-aware affinity.

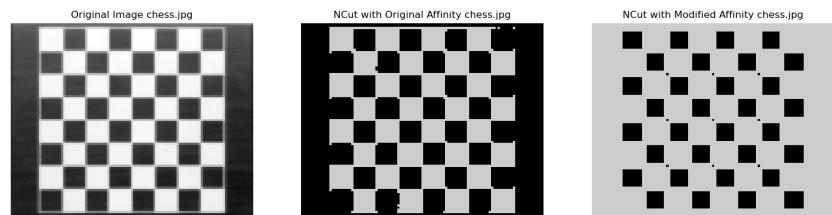


Figure 2: Problem 1(a), chess: Original affinity vs modified edge-aware affinity.

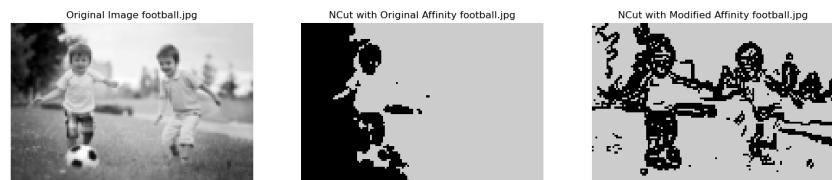


Figure 3: Problem 1(a), football: Original affinity vs modified edge-aware affinity.

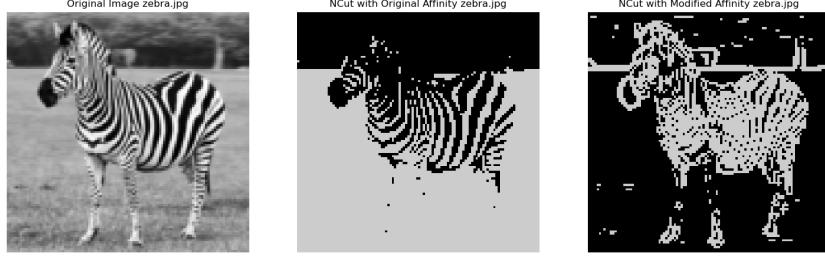


Figure 4: Problem 1(a), zebra: Original affinity vs modified edge-aware affinity.

(b)

Ans: I varied one parameter at a time with the other fixed and compared segmentations for zebra and football. Representative outputs are shown below as individual figures.

When σ_I is small, the factor $\exp(-\|I(i) - I(j)\|^2/\sigma_I^2)$ decays rapidly, so even moderate intensity differences suppress affinity. This increases local fragmentation and causes over-segmentation. As σ_I increases, intensity contrast is down-weighted and many pixels remain strongly connected, which moves toward under-segmentation. Empirically, $\sigma_I \in [0.1, 0.2]$ produced better boundary localization than the extreme values.

For this implementation, the observed trend with σ_X is the opposite: small σ_X gives poorer splits and behaves as under-segmentation, while larger σ_X gives cleaner and more correct partitions. This is consistent with the specific graph construction in code: a full pairwise affinity is built and then hard-thresholded ($W_{ij} < 10^{-3} \rightarrow 0$). With very small σ_X , most spatially separated connections are exponentially suppressed and then removed, so the graph becomes too sparse and the bipartition is less informative. Increasing σ_X preserves more meaningful affinities, improves connectivity structure, and yields better boundary-consistent segmentation in the shown results.

For the modified affinity, the edge term is weighted by

$$\exp\left(-\frac{\|G_i - G_j\|^2}{2\sigma_E^2}\right),$$

with G the normalized gradient magnitude. If σ_E is too small, this factor becomes dominant relative to intensity and spatial terms, and many pairwise

weights collapse toward zero; the effective graph then becomes unstable/noisy. This explains the near-noise behavior visible in Fig. 7. For moderate σ_E , the three components remain comparable in strength and the segmentation stays more consistent while still respecting boundaries.

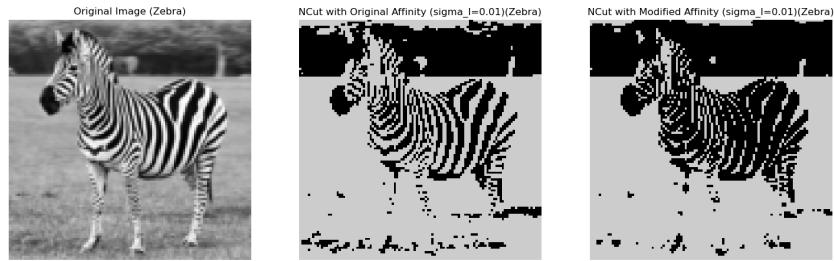


Figure 5: Problem 1(b): Effect of varying σ_I with fixed σ_X, σ_E ($\sigma_I = 0.01$, zebra).

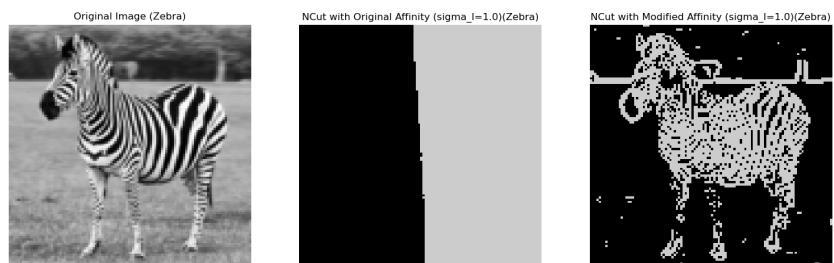


Figure 6: Problem 1(b): Effect of varying σ_I with fixed σ_X, σ_E ($\sigma_I = 1.0$, zebra).

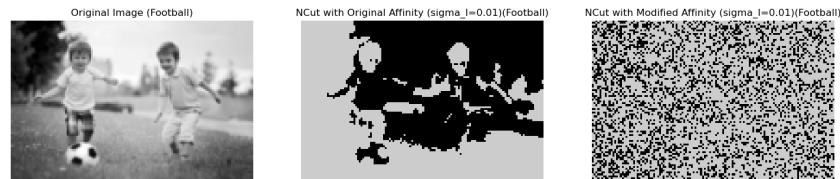


Figure 7: Problem 1(b): Effect of varying σ_I with fixed σ_X, σ_E ($\sigma_I = 0.01$, football).

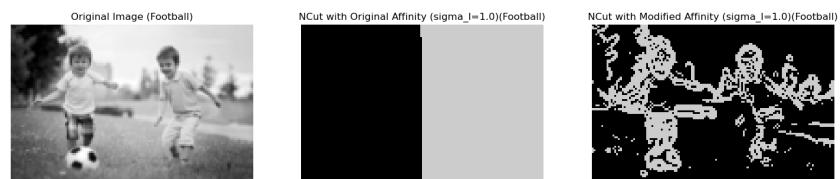


Figure 8: Problem 1(b): Effect of varying σ_I with fixed σ_X, σ_E ($\sigma_I = 1.0$, football).

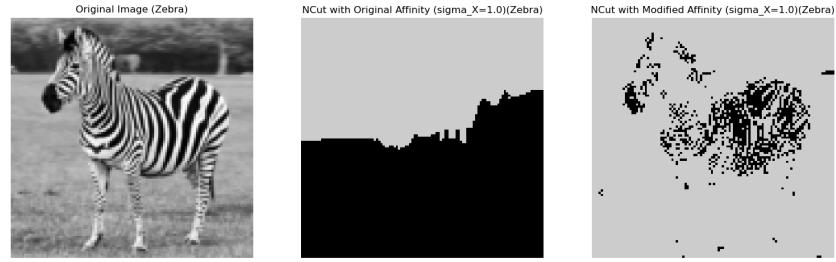


Figure 9: Problem 1(b): Effect of varying σ_X with fixed σ_I, σ_E ($\sigma_X = 1.0$, zebra).

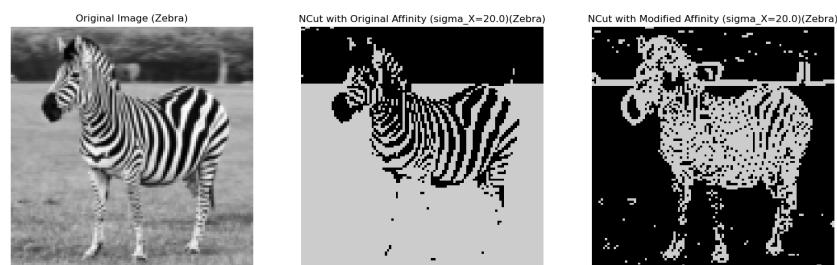


Figure 10: Problem 1(b): Effect of varying σ_X with fixed σ_I, σ_E ($\sigma_X = 20.0$, zebra).

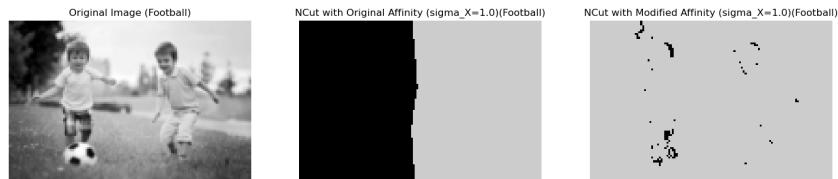


Figure 11: Problem 1(b): Effect of varying σ_X with fixed σ_I, σ_E ($\sigma_X = 1.0$, football).

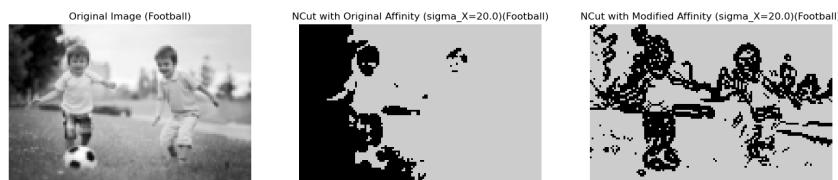


Figure 12: Problem 1(b): Effect of varying σ_X with fixed σ_I, σ_E ($\sigma_X = 20.0$, football).

(c)

Ans: I implemented two multi-way strategies: recursive two-way splitting and simultaneous partitioning using multiple eigenvectors followed by discretization. In recursive splitting, the partition tree is built greedily by repeatedly solving a binary cut; in simultaneous partitioning, the embedding uses the first k non-trivial eigenvectors and assigns labels jointly.

The comparisons are shown in Fig. 13–Fig. 16. In addition to recursive vs simultaneous partitioning, the figures also show original affinity vs modified affinity. The modified affinity produces many thin edge fragments (edge segments), while the original affinity gives fewer and larger regions. This indicates that the edge term is strongly influencing the graph weights and increasing boundary sensitivity.

Mathematically, with

$$w_{ij}^{(\text{mod})} = w_{ij}^{(\text{orig})} \exp\left(-\frac{\|G_i - G_j\|^2}{2\sigma_E^2}\right),$$

large gradient differences suppress cross-edge connections very aggressively. In multi-way splitting, this can create several small boundary-driven components that visually look like noise. This behavior is likely because the edge component has comparable or stronger scale than the intensity/spatial components for the chosen parameters, so edge contrast dominates the partition objective. Recursive splitting still tends to produce more local fragments, whereas simultaneous partitioning is relatively more consistent globally.

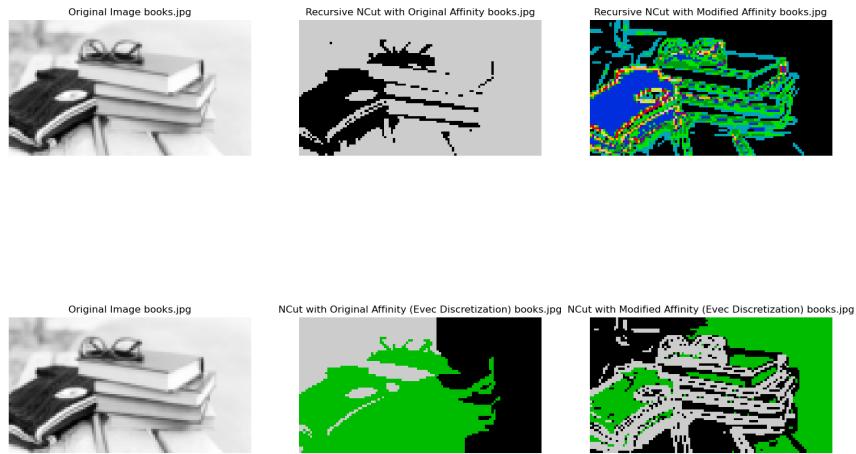
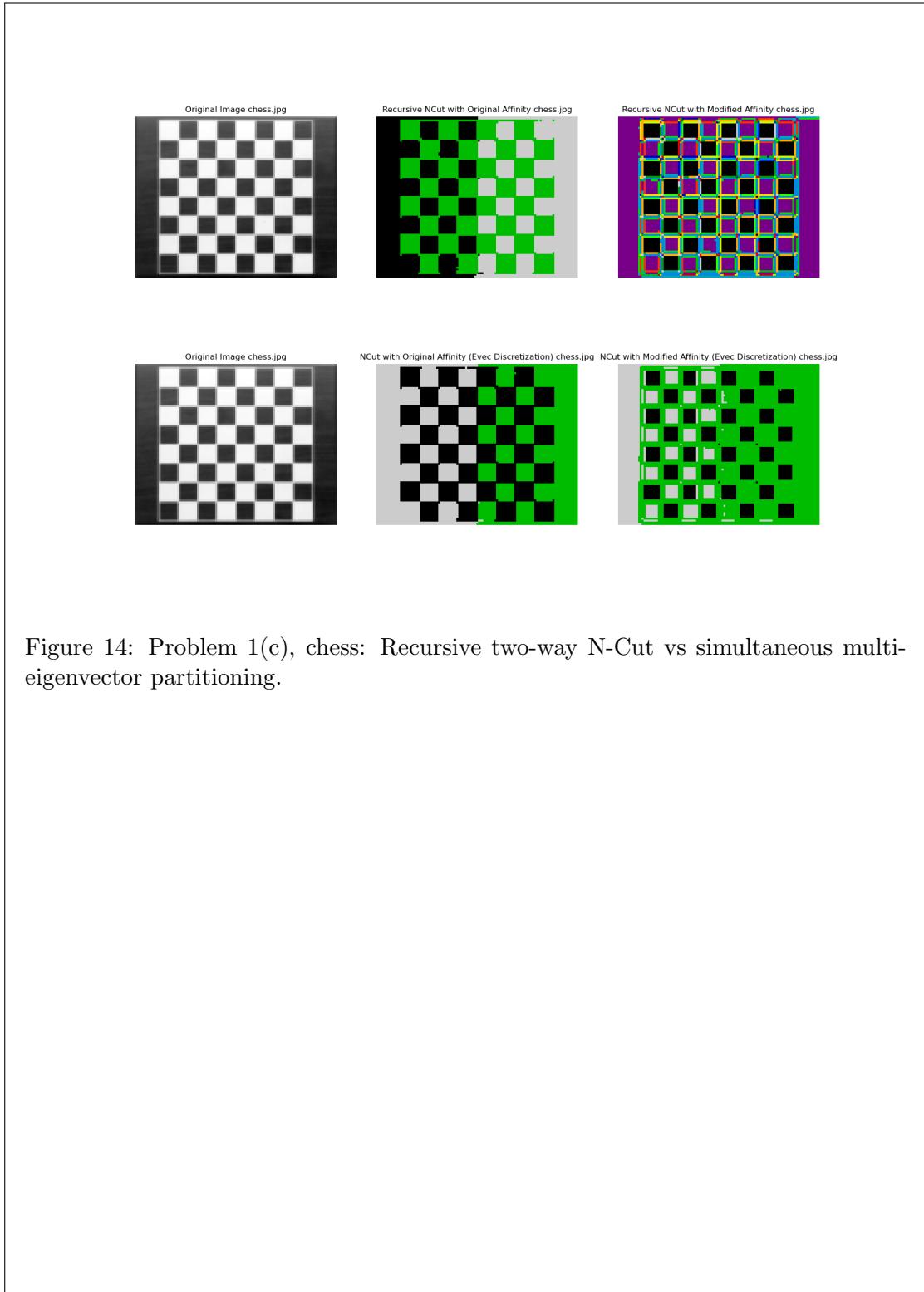


Figure 13: Problem 1(c), books: Recursive two-way N-Cut vs simultaneous multi-eigenvector partitioning.



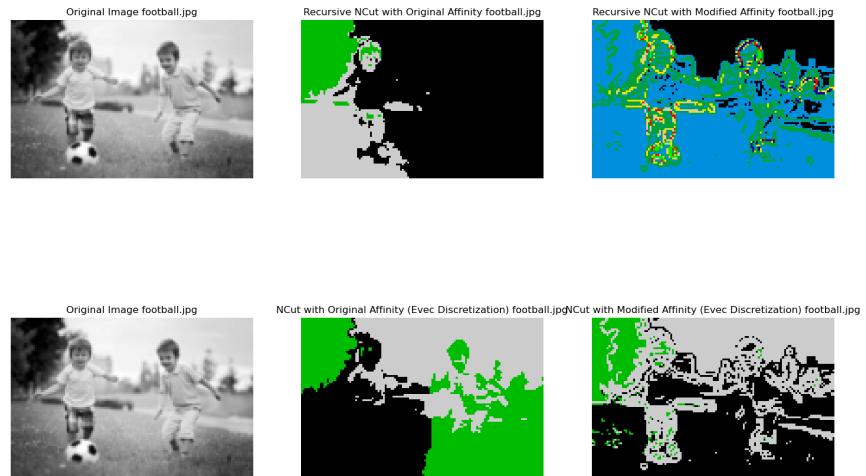


Figure 15: Problem 1(c), football: Recursive two-way N-Cut vs simultaneous multi-eigenvector partitioning.

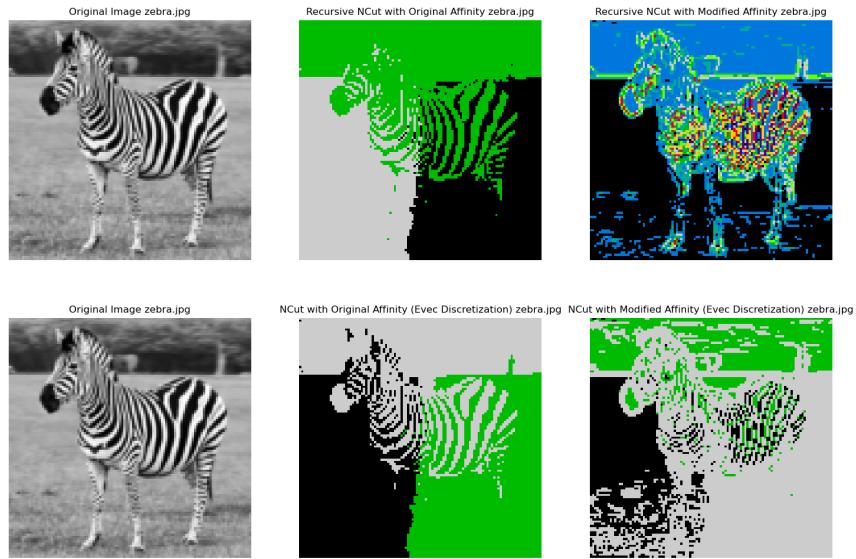


Figure 16: Problem 1(c), zebra: Recursive two-way N-Cut vs simultaneous multi-eigenvector partitioning.

2. Semantic Segmentation with Different Decoder Designs

(a)

Ans: I implemented FCN-32s with a ResNet18 encoder and a single-stage decoder. The decoder upsamples the deepest feature map directly to full resolution, without intermediate skip features. The model tensor dimensions are

$$[1, 3, 512, 512] \rightarrow [1, 2, 512, 512],$$

with 11,185,730 trainable parameters.

After training (15 epochs), test performance is:

$$\text{Pixel Accuracy} = 0.9778, \quad \text{mIoU} = 0.6650.$$

Per-class IoU is 0.9776 (background) and 0.3524 (person). The low person IoU relative to background indicates that direct coarse-to-fine upsampling loses fine boundary detail for the minority foreground class.

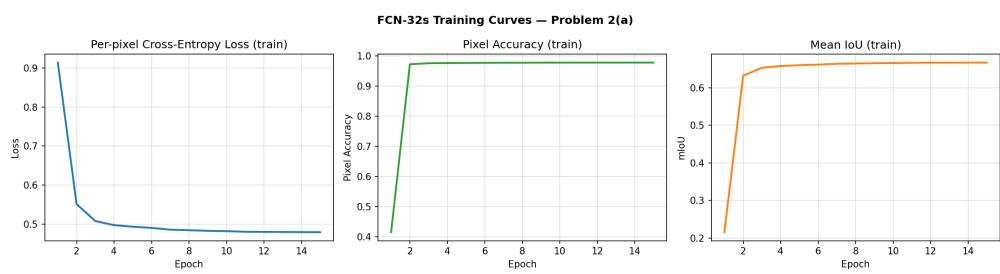
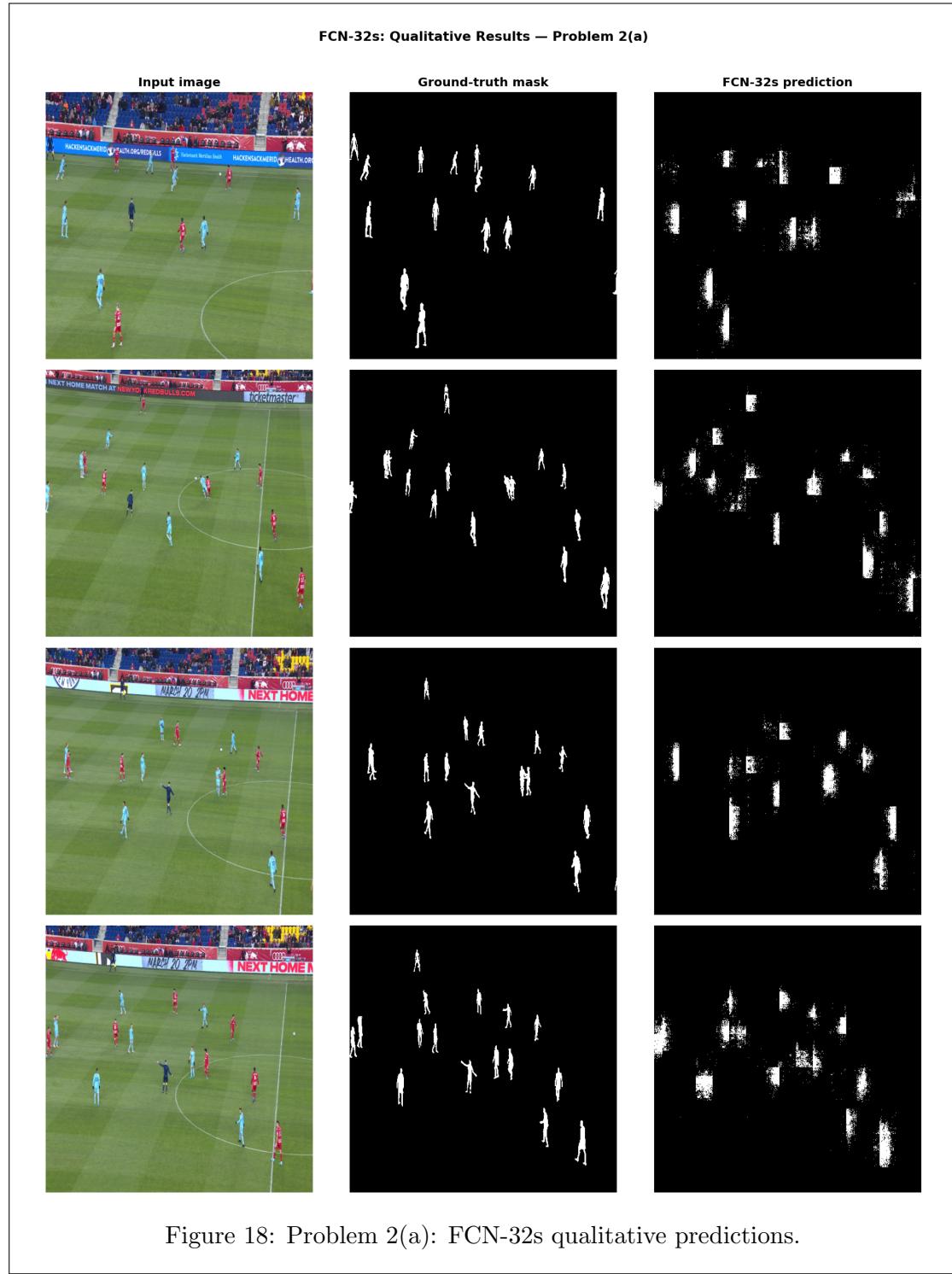


Figure 17: Problem 2(a): FCN-32s training curves.



(b)

Ans: I implemented FCN-8s with progressive multi-stage upsampling and skip connections from two shallower encoder depths. The model output size is again $[1, 2, 512, 512]$, with 11,178,886 trainable parameters.

The encoder is ResNet18 split at three scales:

$$\begin{aligned} \text{pool3} &: [B, 128, H/8, W/8], \\ \text{pool4} &: [B, 256, H/16, W/16], \\ \text{pool5} &: [B, 512, H/32, W/32]. \end{aligned}$$

After 1×1 class scorers on each level, the decoder performs skip-fusion as

$$\hat{S}_{16} = \text{Up}_2(S_5) + S_4, \quad \hat{S}_8 = \text{Up}_2(\hat{S}_{16}) + S_3, \quad \hat{Y} = \text{Up}_8(\hat{S}_8),$$

where S_5, S_4, S_3 are class-score maps from pool5, pool4, and pool3. Also, Up_2, Up_8 are transposed-convolution upsamplers. For 512×512 input, this gives

$$\begin{aligned} [B, C, 16, 16] &\rightarrow [B, C, 32, 32] \\ &\rightarrow [B, C, 64, 64] \\ &\rightarrow [B, C, 512, 512]. \end{aligned}$$

After training (20 epochs), test performance is:

$$\text{Pixel Accuracy} = 0.9960, \quad \text{mIoU} = 0.9056.$$

Per-class IoU is 0.9960 (background) and 0.8152 (person). Compared to FCN-32s, the person IoU gain is substantial, confirming that skip-fused higher-resolution features preserve shape and boundary cues that are absent in single-stage decoding.

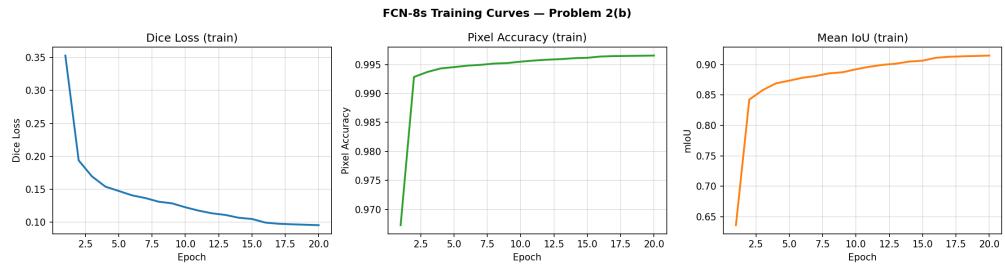
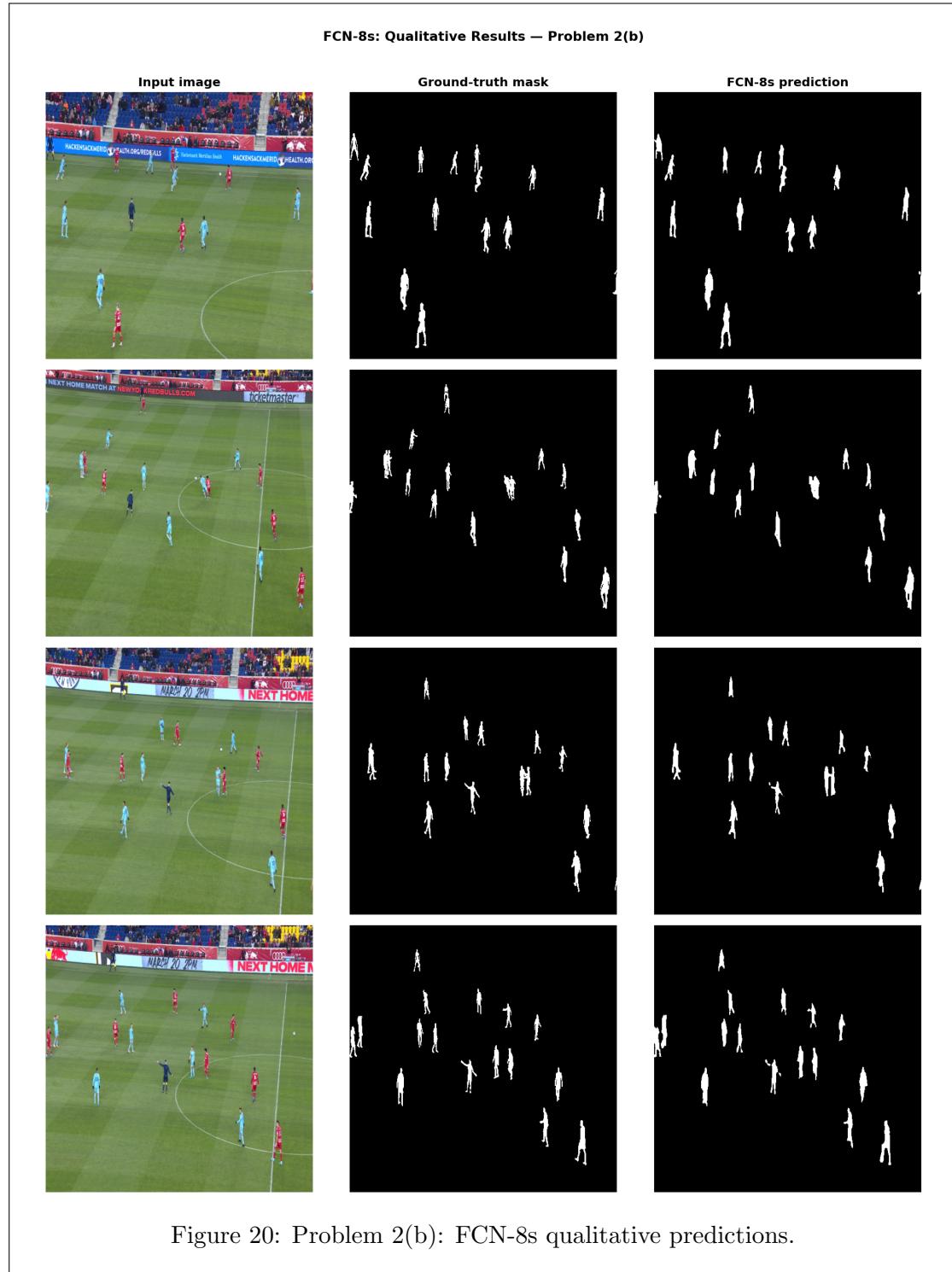


Figure 19: Problem 2(b): FCN-8s training curves.



(c)

Ans: Quantitatively, FCN-8s outperforms FCN-32s on both metrics:

$$\Delta \text{Pixel Accuracy} = 0.9960 - 0.9778 = 0.0182,$$

$$\Delta \text{mIoU} = 0.9056 - 0.6650 = 0.2406.$$

For the foreground class, the improvement is

$$\Delta \text{IoU}_{\text{person}} = 0.8152 - 0.3524 = 0.4628.$$

These gaps are consistent with the decoder design: progressive upsampling decreases aliasing from coarse logits, and skip connections inject high-frequency encoder information. As a result, boundary transitions are sharper and thin person structures are retained more reliably. In contrast, direct 32 \times upsampling from deep low-resolution features causes boundary smoothing and foreground erosion.

Table 1: Test-set comparison of decoder designs.

Model	Pixel Accuracy	mIoU	Person IoU
FCN-32s	0.9778	0.6650	0.3524
FCN-8s	0.9960	0.9056	0.8152