Two Heads Are Enough: Dual U-Net – Design & Analysis of a Fast and Efficient Cell Classification & Segmentation Algorithm

Muhammad Abdul Rafey Farooqi
369797

Problem & Motivation

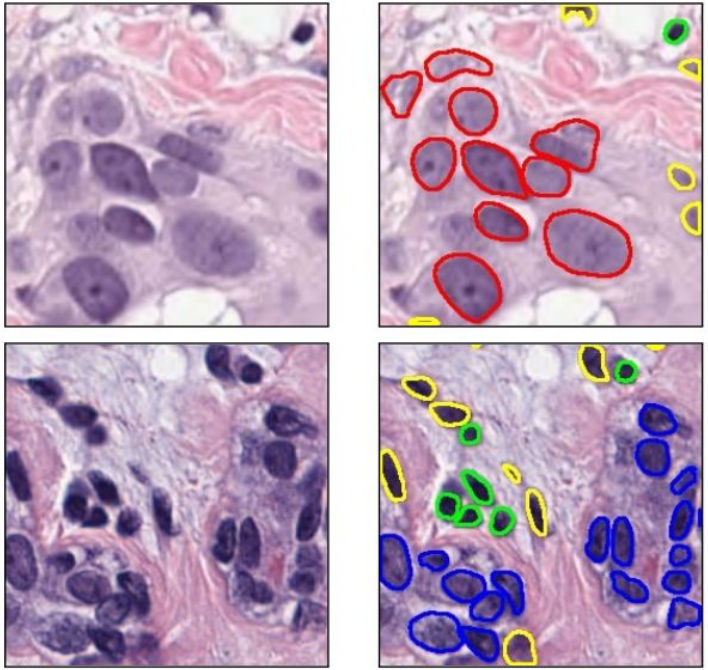
Goal: Efficient nuclei segmentation in high-res histopathology

Challenges:

1. Large image sizes → high computational cost.
2. Existing models (e.g., HoVer-Net) use 3 decoders → heavy compute & memory.

Algorithmic framing:

- *Input*: Large pixel grids (often 1000×1000+ patches).
- *Output*: Two maps – segmentation & centroid detection.



Architecture Overview

Key Change

Dual heads instead of triple decoder (HoVer-Net).

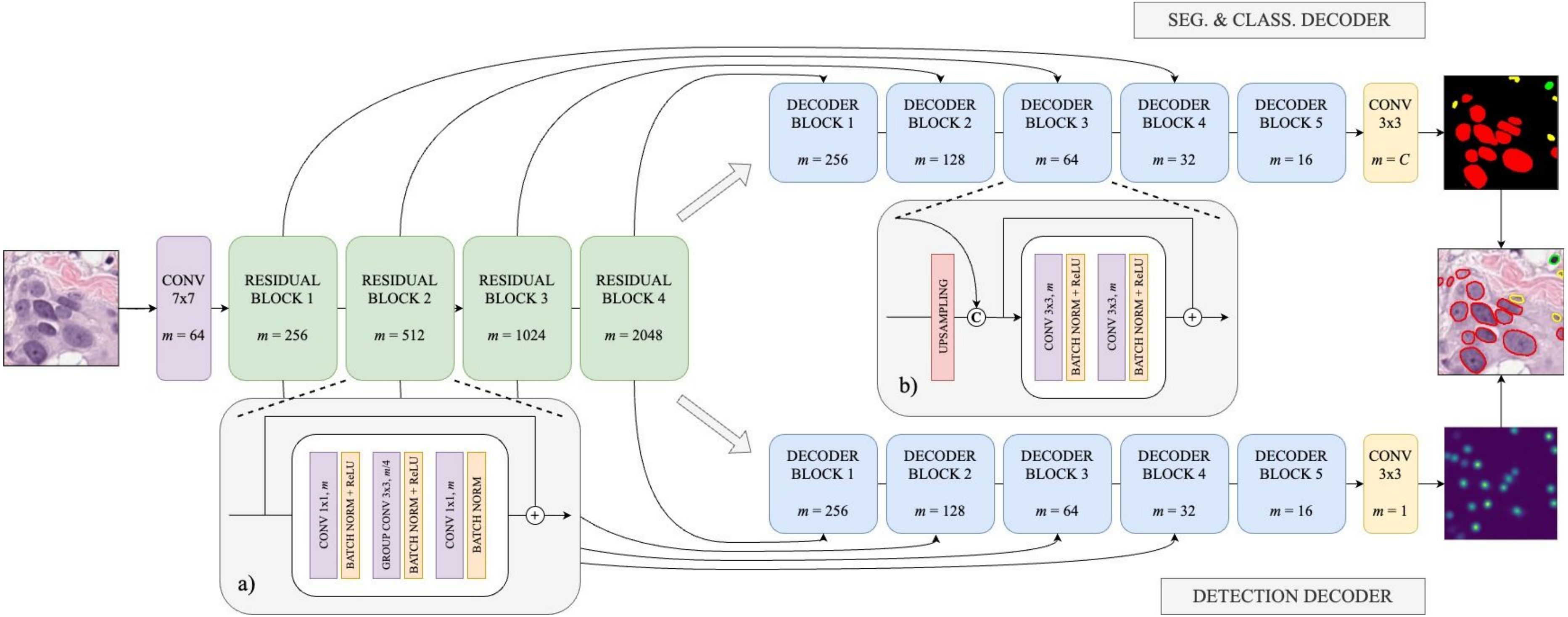
Segmentation Head

Outputs per-pixel class labels.

Detection Head

Outputs centroid heatmap.

Architecture Overview



Algorithmic Effect

Reduced Cost

Reduces $O(f \times p)$ style cost from extra decoder passes (f = feature maps, p = pixels).

Simpler Fusion

Less feature fusion complexity.

Data Flow & Algorithm Steps

Algorithm 1 DualU-Net Data Flow and Complexity Model

Input: Image patch of N pixels

Output: Segmentation map, centroid heatmap

Step 1: Encoder

Extract multi-scale features using U-Net backbone.

Complexity per layer:

$$O\left(N \times k^2 \times c_{\text{in}} \times c_{\text{out}}\right)$$

Step 2: Decoder Stage

Feed features into d parallel decoders.

Complexity:

$$O\left(L \times N \times k^2 \times C \times d\right)$$

where L = layers per decoder, C = channels term.

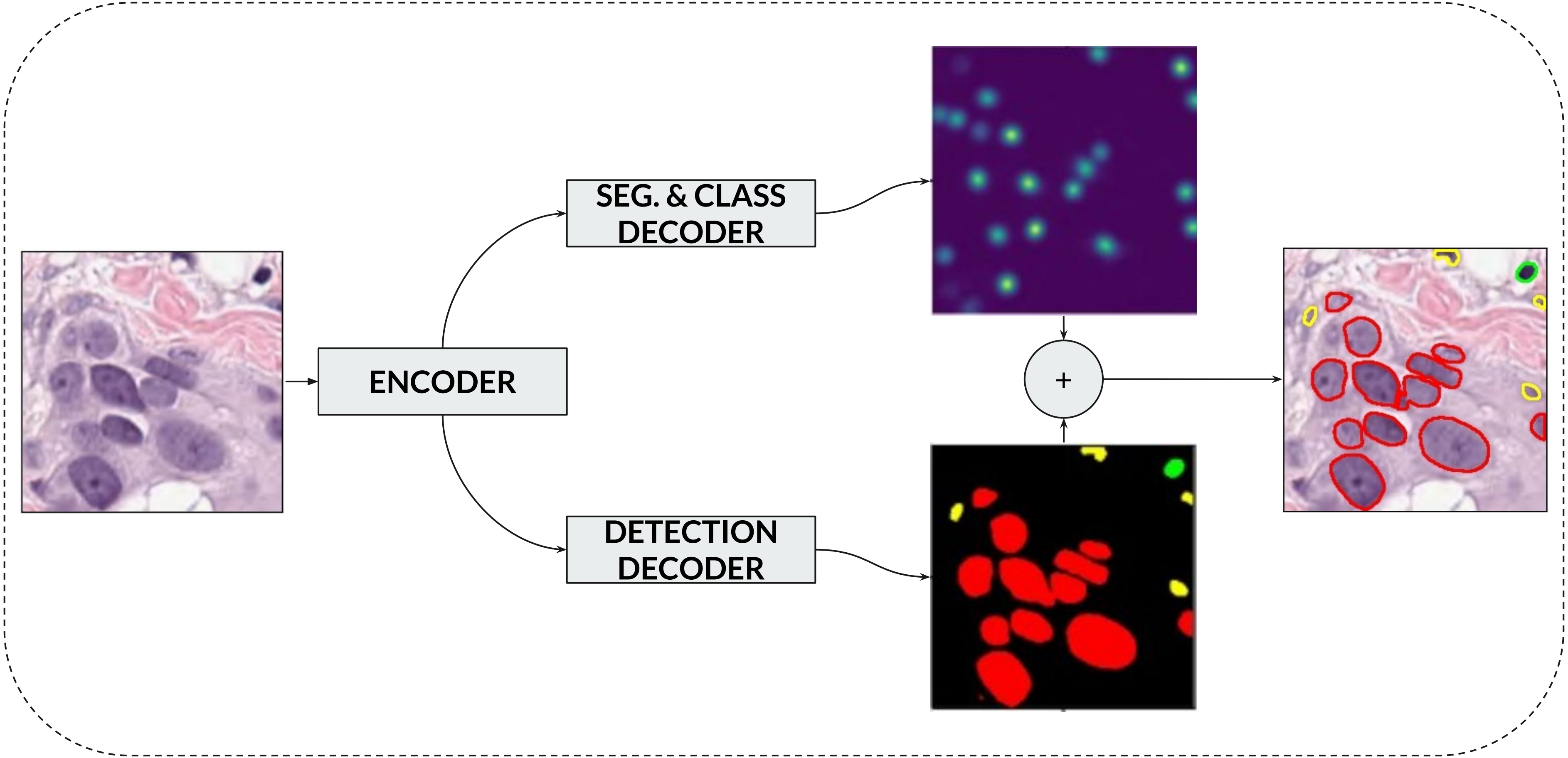
Step 3: Output Combination

Merge segmentation and centroid heatmap to obtain instance-level segmentation.

Note: For DualU-Net, $d = 2$; for HoVer-Net, $d = 3$.

Thus, DualU-Net has $\frac{2}{3}$ of the decoder-side computation.

Data Flow & Algorithm Steps



Time Complexity Analysis

Let:

$$N = \text{number of pixels in input patch (e.g., } W \times H) \tag{1}$$
$$d = \text{number of decoder branches} \tag{2}$$
$$L = \text{number of convolutional layers per decoder} \tag{3}$$

Derivation:

$$\text{Per-conv cost: } O(N \times k^2 \times C) \tag{4}$$
$$\text{Total decoder cost: } O(L \times N \times k^2 \times C \times d) \tag{5}$$

Comparison:

Model	d	Complexity	Relative Cost
HoVer-Net	3	$O(L \times N \times k^2 \times C \times 3)$	100%
DualU-Net	2	$O(L \times N \times k^2 \times C \times 2)$	67%

Space Complexity Analysis

Activation memory: $O(N \times C \times d)$ at decoder output.

Weight storage: Linear in $d \times L \times k^2 \times C$.

Fewer decoders = fewer parameters & less intermediate feature storage.

This leads to **lower GPU VRAM** usage → allows larger batch sizes.

Empirical Runtime

Paper reports approx. 5× faster inference than HoVer-Net in practice.

Reason:

- Fewer convolution ops in decoder phase.
- Less memory transfer overhead.

Algorithmic takeaway: Real-world performance improvement matches predicted asymptotic reduction.

Model	Nº Parameters (M)	GLOPs ↓		Latency (ms) ↓	
		256	1024	256	1024
$CellViT_{256}$	46.75	132.89	2125.94	35.71 ± 0.37	1169.7 ± 148.92
NuLite-S	34.10	23.15	370.25	29.99 ± 1.79	310.44 ± 24.64
NuLite-M	47.93	32.54	520.45	33.37 ± 1.34	446.3 ± 35.25
Ours	41.01	16.26	260.23	12.05 ± 0.41	141.88 ± 0.69
Ours ConvNeXt	97.81	26.78	428.49	20.82 ± 0.17	264.19 ± 1.48

Conclusion

- Algorithm design choices (like number of heads) directly affect complexity.
- DualU-Net $\rightarrow$ $O(2 \times \text{decoder cost})$ vs HoVer-Net's $O(3 \times \text{decoder cost})$.
- Efficiency gains validated in real deployment (8 hospitals).
- **Takeaway:** *Smart architectural simplification beats brute force.*