# Preliminary Study on Deep Iterative Semantic Segmentation

Diana Teixeira e Silva, Ricardo Cruz, Tiago Gonçalves

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO
INESCTEC

## Motivation

A common practice for segmentation of high-resolution images is splitting the image into patches and processing each patch separately. Such approach has two problems:

- ✕ It spends as much time in hard-to-segment regions as it does in easy-to-segment regions, where there is nothing of relevance;
- ✕ The boundary between patches is a problem and must be dealt with in a special way.

## Related Work

- There are works that focus on improving training and/or inference time, typically by working with multiple scales, but not on segmentation tasks [1].
- For example, due to computational constraints, Google AI performs alpha matting on mobile devices by a two-stage process whereby a neural network performs an initial step and a secondary network is used only on any areas that might require further work [2].

## Proposed Method

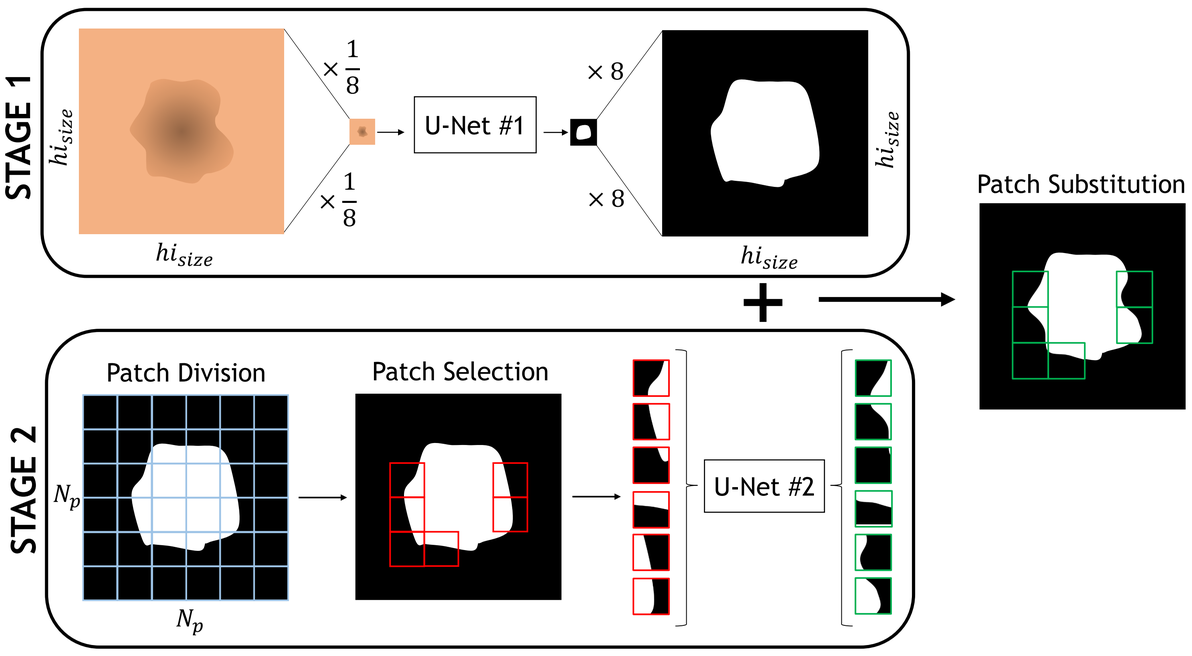


Figure 1. Two-stage segmentation.

**STAGE 1 (S1):** Segmentation of a low-resolution version of the image by a first neural network.

**STAGE 2 (S2):** Based on the probabilities produced by the neural network from S1, identify poorly segmented image patches and use a second neural network to refine those patches.

**Patch Selection Criteria:** $N_p$ patches with lowest $|mean - 0{,}5|$.

## Implementation

Table 1. Datasets for semantic segmentation.

| Dataset | N | Avg Res | $hi_{size}$ | %Fg | Example |
|---|---|---|---|---|---|
| PH2 | 200 | 575×766 | 768 | 31.4 |  |
| RETINA | 66 | 745×782 | 768 | 7.3 |  |
| KITTI | 200 | 375×1271 | 512 | 6.6 |  |

Training:

- **U-Net #1 (S1):** Images were resized to $(1/8 + 5\%) \cdot hi_{size}$, followed by a random cropping of dimensions corresponding to $1/8 \cdot hi_{size}$. Trained for 200 epochs.
- **U-Net #2 (S2):** Images were resized to $hi_{size}$ in both dimensions, with a random crop to a square of length $(1/N_p) \cdot hi_{size}$. Trained for 1000 epochs.
- **Baseline (BL):** Traditional approach consisting of the same U-Net architecture using the original images as input.

Loss Function: $\mathcal{L}(y, \hat{y}) = \mathcal{L}_f(y, \hat{y}) + (1 - D(y, \hat{y}))$, where $\mathcal{L}_f$ is the focal loss.

## Results

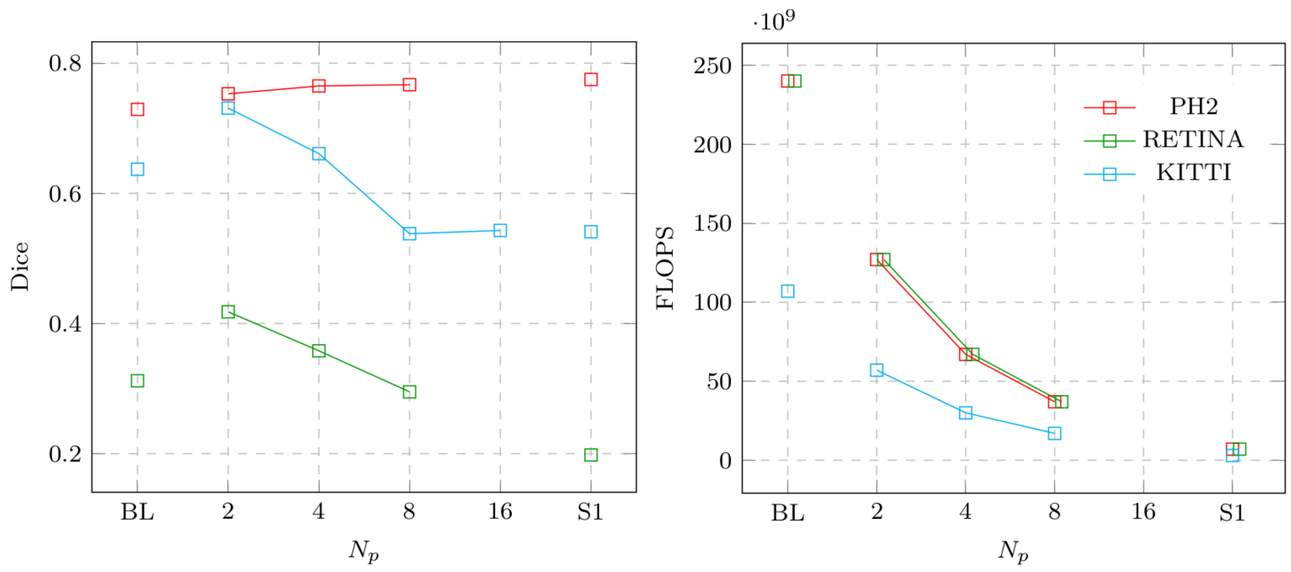


Figure 2. Overall results for the method.

- Dice Coefficient: Typical noticeable gain between S1 and S2. Increasing the number of patches $N_p$ either results in a plateau or reduced performance.
- FLOPS: Increasing the number of patches, the number of operations reduces because, while S1 is the same, S2 operates with smaller patches. In all cases, our method requires fewer operations than the baseline.

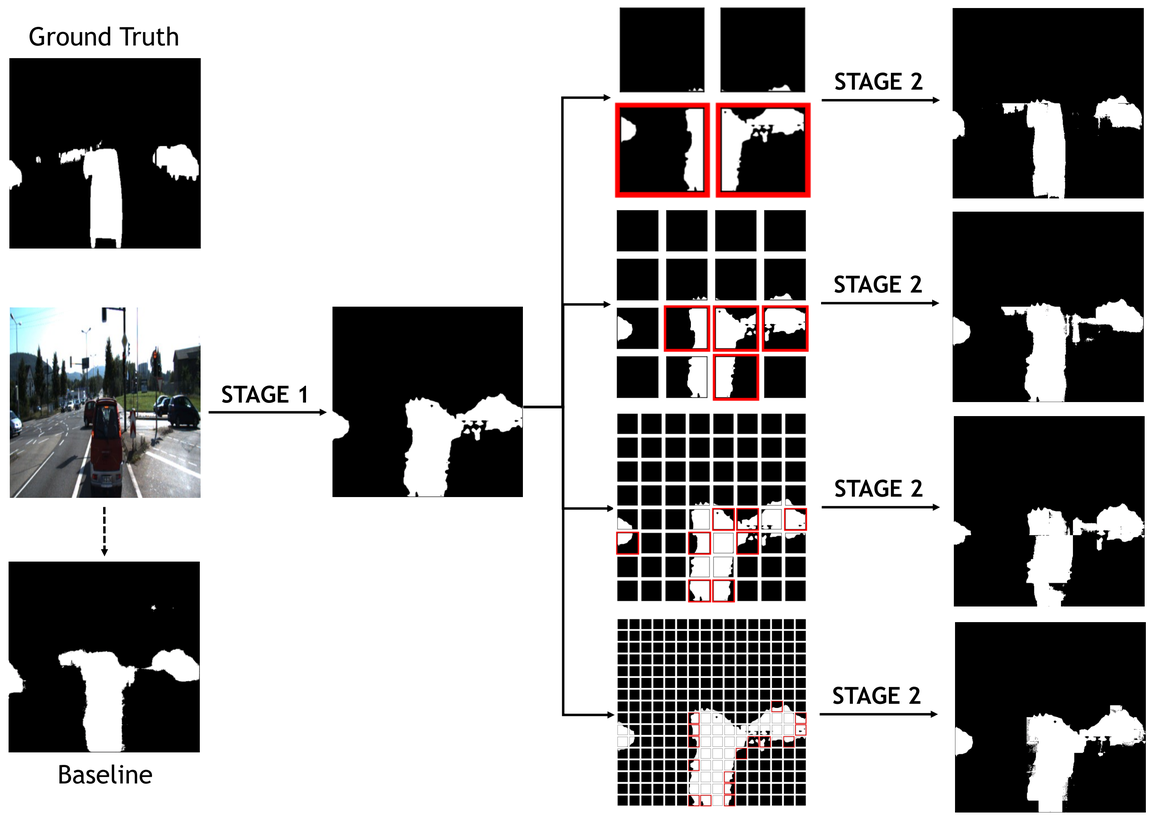


Figure 3. Results for an image from the KITTI dataset.

## Conclusion and Future Work

- Our method showed overall similar Dice coefficients to the baseline while saving up between 50% and 80% of the total number of operations.
- For future work, the choice of patches could be improved using a more fine-grained sliding window instead of a contiguous sliding window.

**References:**

[1] K. Thandiackal et al., "Differentiable Zooming for Multiple Instance Learning on Whole-Slide Images," arXiv preprint arXiv:2204.12454, 2022.
[2] Google AI Blog, "Accurate Alpha Matting for Portrait Mode Selfies on Pixel 6." 2022.

This work is supported by European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 047264; Funding Reference: POCI-01-0247-FEDER-047264], and by National Funds through the Portuguese funding agency FCT - Fundação para a Ciência e a Tecnologia - within the PhD grant "2020.06434.BD".