Progetto 3
Segmentazione della scena stradale

La segmentazione semantica di immagini risulta essere molto utile soprattutto per la comprensione delle scene stradali, di cui è riportato un esempio in figura 1. Obiettivo di questo progetto è la segmentazione usando il dataset IDD (India Driving Datase) [1] costituito da circa 20k immagini di scene stradali. Per ogni immagine è fornita la mappa di segmentazione a 34 classi. Per questo progetto, userete come architettura una variante di U-Net in cui la parte di encoder, denominata anche *backbone*, viene realizzata attraverso una rete più profonda. In particolare, sostituirete l'encoder classico con ResNet50 inizializzata con pesi pre-addestrati su ImageNet. Per valutare la segmentazione utilizzatore l'F-score (F1) come indice prestazionale. Data una classe target, l'F-score (F1) è una misura compresa tra 0 e 1 (localizzazione perfetta) definita come:

$$\text{F1} = \frac{2\,TP}{2\,TP + FN + FP}$$

dove TP, TN, FP e FN sono:

- TP (true positive): # pixel della classe target dichiarati della classe target;
- TN (true negative): # pixel no della classe target dichiarati no della classe target;
- FP (false positive): # pixel no della classe target dichiarati della classe target;
- FN (false negative): # pixel della classe target dichiarati no della classe target;

Per il progetto utilizzate la libreria `segmentation_models`, che fornisce molte funzioni utili per la segmentazione. La documentazione di `segmentation_models` si trova al seguente link: https://segmentation-models.readthedocs.io

In questo progetto i passi da seguire sono:

1. **Installazione di** `segmentation_models`. Per installare la libraria `segmentation_models` eseguite le seguenti istruzioni all'inizio del codice:
   ```
   !pip install git+https://github.com/qubvel/segmentation_models
   import os
   os.environ['SM_FRAMEWORK'] = 'tf.keras'
   import segmentation_models as sm
   ```
2. **Download dei dati**. Scaricare dal sito ufficiale [1] la versione Lite del dataset IDD contenenti 1403 immagini di training a 204 immagini di test. Tutte le immagini hanno la risoluzione di 227×320 pixel.

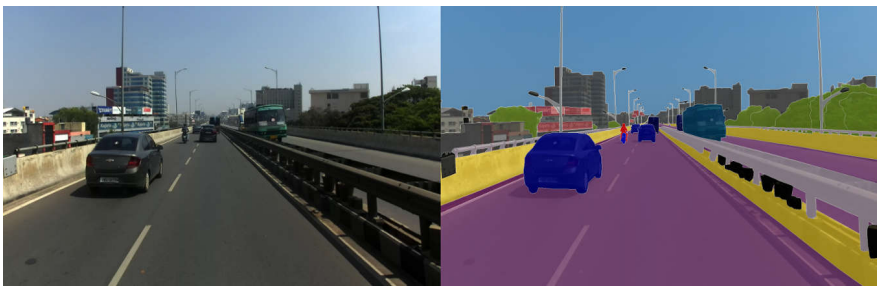

Figure 1: Esempio di scena stradale con relativa mappa di segmentazione semantica.

3. **Preparazione dei dati**. Delle 1403 immagini di training utilizzate, 1200 per il training, e il restante per la validazione. Tutte le immagini e le relative mappe di segmentazione vanno inoltre ritagliate a 224×224 pixel.

4. **Architettura**. Per definire l'architettura U-Net utilizzate la funzione `segmentation_models.Unet()`. In particolare, utilizzare il parametro `backbone_name` della funzione per adottare l'architettura ResNet50 come encoder. Inoltre, utilizzate il parametro `encoder_weights` per utilizare i pesi pre-addestrati su ImageNet.

5. **Addestramento**. Per l'addestramento utilizzate l'ottimizzatore Adam tramite la funzione di Keras `keras.optimizers.Adam`. Inoltre utilizzate come metrica l'F-score, mentre per la funsione di loss provate sia la categorical cross-entropy loss che la dice loss. Sia per la metrica che per le loss utilizzate le funzioni fornite della libreria `segmentation_models`. Infine, utilizzate le prestazioni sul set di validazione per selezionare i migliori valori per il learning-rate, il batch-size e il numero di epoche.

6. **Valutazione delle prestazioni**. Per ogni immagini di test valutate le prestazioni a livello pixel calcolando l'F-score (F1) per ogni classe.

References

[1] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. V. Jawahar "IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments" IEEE Winter Conf. on Applications of Computer Vision (WACV 2019) http://idd.insaan.iiit.ac.in/