Deep learning architecture-based method to detect spalling in concrete surfaces

This code is based on the works from the Advanced Robotics and Automation Laboratory;

1. H. Ahmed, C. P. Le, and H. M. La. Pixel-level classification for bridge deck rebar detection and localization using multi-stage deep encoder-decoder network. Developments in the Built Environment. Vol. 14. 2023. 100132.
2. H. Ahmed and H. M. La* and K. Tran. Rebar Detection and Localization for Bridge Deck Inspection and Evaluation using Deep Residual Network. Journal of Automation in Construction, Elsevier publisher, Vol. 120, December 2020. Impact Factor: 5.669.

***However, we have highly modified it for our work.

Training and Validation of Deep Encoder-Decoder Networks:

1. Download the code ZIP file.

2. Unzip the folder

3. Install gpu for Tensorflow (recommended version: 2.4.1)
https://www.tensorflow.org/install/gpu

4. Resolve the following dependencies:

Keras (recommended version: 2.4.3)
- Use the command “**pip3 install keras**“

OpenCV for Python
imgaug
tqdm

5. Run the code for training the SegNet or other frameworks using the code line given below:

```
python3 -m keras_segmentation train --checkpoints_path="checkpoints"
--train_images="example_dataset/train"
--train_annotations="example_dataset/train_annot"
--val_images="example_dataset/valid" --val_annotations="example_dataset/valid_annot"
--n_classes=3 --input_height=764 --input_width=764 --model_name="segnet" --epoch=100
--batch_size=8
```

where **segnet** is, switch out different type of encoder + base model combinations like ‘xception_segnet’, ‘vgg_segnet’, etc. Refer to the paper for all different types of combinations. **checkpoints** are where the model will go. If you just have checkpoints, then all model will be in **Segmentation + segnet/** folder. If you want your model to go to a different folder like mine where I have it in **checkpoints/,** then first create a folder name **<folder name>**, then in place of **checkpoints**, put in **<folder name>/checkpoints.**

For input height and width, please check the base model requirement by opening **keras_segmentation/models/<base model name>.py** and input the right width and height, you will get errors. Below is the base model and encoder names:

6. Run the code line given below in the command prompt to validate the trained Deep Encoder-Decoder system:

```
python3 -m keras_segmentation predict --checkpoints_path="checkpoints"
--input_path="example_dataset/test" --output_path="prediction"
```

For **checkpoints**, if in **step 5** you choose to put your model in **<folder name>/checkpoints**, then you will have to put **<folder name>/checkpoints** in place of **checkpoints**

7. Run the code line given below in the command prompt to get metrics like precision, iou score, etc

```
python3 -m keras_segmentation evaluate_model --checkpoints_path="checkpoints"
--images_path="example_dataset/test" --segs_path="example_dataset/test_annot"
```

note: there might be some warnings so you might have to scroll around to find the print statement in the terminal.

For **checkpoints**, if in **step 5** you choose to put your model in **<folder name>/checkpoints**, then you will have to put **<folder name>/checkpoints** in place of **checkpoints**

Data:

- The filenames of the annotation images should be the same as the filenames of the RGB images.
- The size of the annotation image for the corresponding RGB image should be the same.
- The annotation should follow the RGB value range based on the class number. If the segmentation is on n-class the annotation RGB values should be within (0-n-1) range.