**Project 3 – Computer Vision**

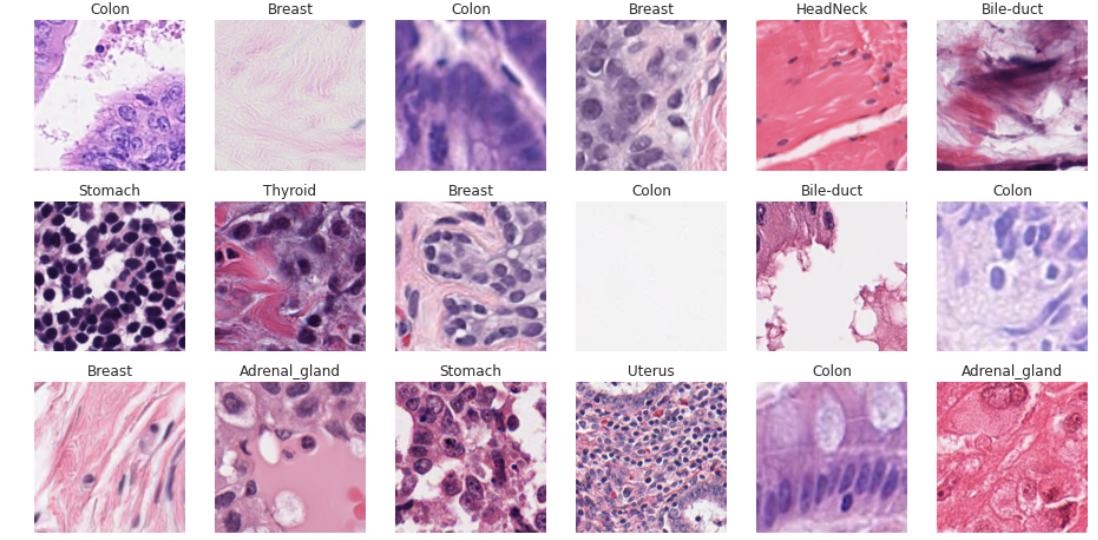
1. Description of the data set

We selected the dataset : <https://www.kaggle.com/datasets/andrewmvd/cancer-inst-segmentation-and-classification>.

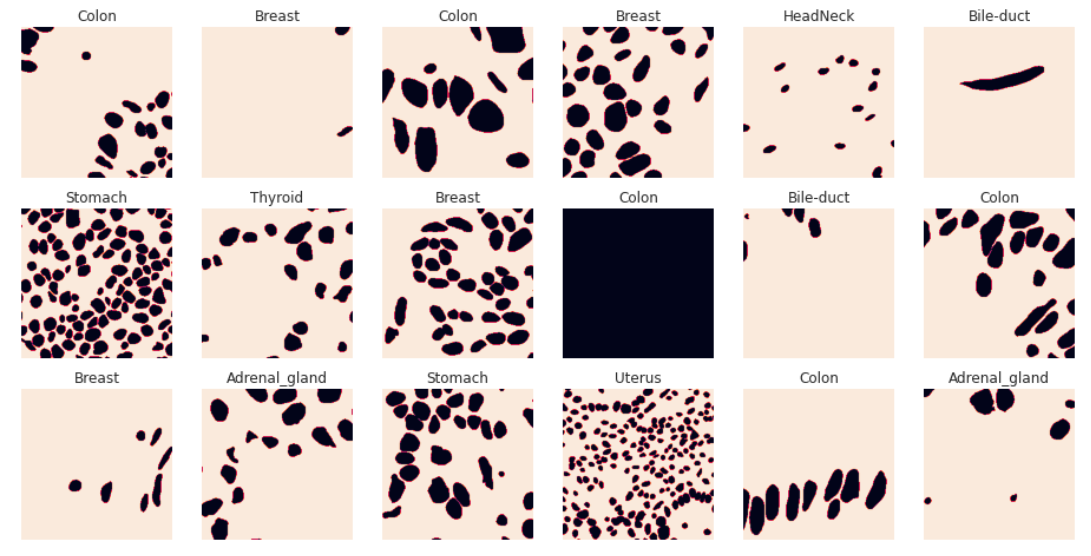
This dataset is known as PanNuke and contains semi automatically generated nuclei instance segmentation and classification images with exhaustive nuclei labels across 19 different tissue types. (Breast, Colon, Bile-duct, Esophagus, Uterus, Lung, Cervix, Head&Neck, Skin, Adrenal Gland, Kidney, Stomach, Prostate, Testis, Liver, Thyroid, Pancreas, Ovary, Bladder).

The datset contains 2656 images of tissue, the label of the tissue type it belongs to and the masks (array of 6 channel instance-wise masks, 0:Nocleoplastic cells, 1:Inflammatory, 2 : Connective/Soft tissue cells, 3: Dead Cells, 4: Epithelial, 6: Background/non nuclei part of the tissue).

This is a set of 18 images from our dataset with their corresponding labels :



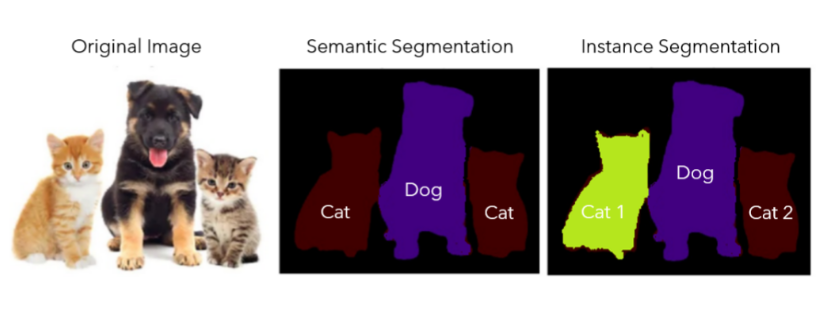
The masks give us this result for those images (colored zone are part of background, in black there is part of images that belong to a nuclei) :



1. Description of the problem

The problem that we choose is the problem of instance segmentation.

Instance segmentation is a form of image segmentation (deals with detecting instances of objects and demarcating their boundaries). So, Instance Segmentation is the technique of detecting, segmenting, and classifying every individual object in an image. The principal difference with semantic segmantation is that we don’t just want to detect the class of the object that the object belongs to but we also want to determine the number of instance of each class that we can find on our image.

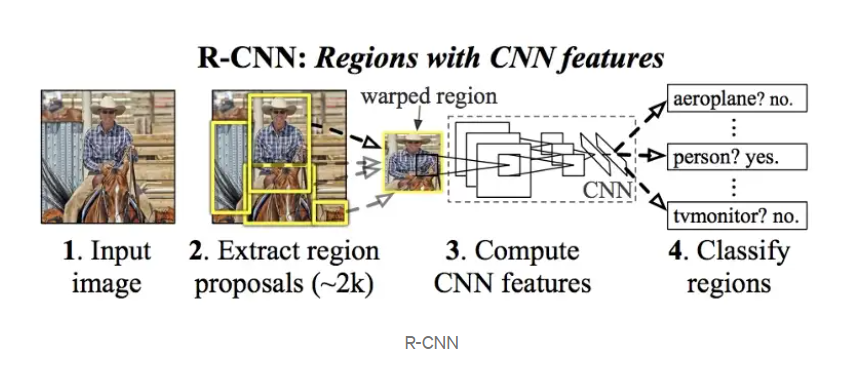
Image from https://www.v7labs.com/blog/instance-segmentation-guide

1. Description of the Model
2. R-CNN

R-CNN (Regions with CNN features) is a type of convolutional neural network (CNN) architecture for object detection. The basic idea behind R-CNN is to generate a set of region proposals, which are locations of objects in an image. These regions are then passed through a CNN to extract features, which are then used to classify the objects in the regions and refine the bounding box coordinates.

Three main steps:

1. Region Proposal: generates a set of candidate regions (proposals) that contain objects. [Using methods like Selective Search or Edge Boxes.]
2. Feature Extraction: each region proposal is resized to a fixed size and passed through a CNN to extract a feature vector.
3. Classification and Localization: a classifier is trained on the extracted features to predict the class label of the object and adjust the coordinates of the bounding box.



R-CNN demand to first convert data in COCO format. We thus need to create a COCO JSON file with annotations looking like this:

« annotations » :[{

"image\_id": 42, image file name

"category\_id": 18, label of the object in the image

"bbox": [258.15,41.29,348.26,243.78], bounding box coordinates of the object in the format [x,y,width,height], x and y are the coordinates of the top-left corner of the bounding box.

"area": 34104.66, area of the bounding box

"iscrowd": 0 binary value that indicates whether the object is part of a group or crowd of objects.

},...]

We also need to provide a list of all the objects labels we want to detect, and id for each label.

« categories » :[ { "id": 1, "name": "name\_of\_type1\_nuclei" }, { "id": 2, "name": "name\_of\_type2\_nuclei" },

...]

1. YOLO
2. …
3. Bibliography

Dataset : https://www.kaggle.com/datasets/andrewmvd/cancer-inst-segmentation-and-classification

https://www.v7labs.com/blog/instance-segmentation-guide

https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Code inspiration for ResNet50 : https://pytorch.org/tutorials/intermediate/torchvision\_tutorial.html

The Pan-Cancer-Nuclei dataset is a challenging dataset and it might require a powerful machine with a GPU to train the model. Need to carefully preprocess the dataset and adjust the parameters of the model to achieve good performance.