Preprocessing training network :
The objective is to train a neural network to distinguish between normal, non-normal and unused areas (i.e. blanks). The DeepPath paper doesn't distinguish between the first two in generating the training data for cancer identification from a slide. Such an approach *may* introduce undesired noise (i.e. the normal areas) in identifying the patterns that lead to cancer. We would like to find out if removing this noise helps in making the training more efficient (i.e. smaller dataset can provide same or better performance) and prediction more accurate (i.e. prediction is correct with higher percentage) and precise (i.e. probabilities of correct prediction have less variance).

1) **Dataset :** TCGA
2) **Training phase :** In this phase, the cancer slides are marked down regions of normal type and the pure tiles of an image are fed into the network.
   A pathologist (Dr. Mishra in our case) will mark areas on the slides which are normal and non-normal. Not all areas need to be marked. Tiles will be created from these areas with tags normal and non-normal. Rest of the areas will be discarded. The non-normal area may inherit the tag of the slide (e.g. cancer, LUSC, LUAD, grade). These later tags will not be used in training in this phase (at least for now). With this set of tiles we can set up a NN for categorisation. Note that we can use any other network, not only inception V3. How do we decide which one will be best suited?
3) **Testing phase :** In this phase, the bare cancer slides are given as input, it needs to predict the probability of each tile of the whole slide being normal (just like the heat map depicted in the paper).
   The bare slide will be chopped off in tiles. The network have to predict which part is normal, which is non-normal and rest will be considered unused. Dr. Mishra will take a look at these predictions and give them scores.

Proposed Workflow of the project: (Small scale)

1) **Make the network ready :**

- Proposal :
  Build a own image classification small scale network or modify and adopt the existing two way classifier in the Github.

2) **Get the slides, markings on the slides and train the network :**

- Retrieval :
  Download around 50 of the images from the dataset with different stages of the slides.
- Marking :
  Get the slides marked using aperio and break down the slides as tiles for training step **1**
  Any alternative way to get this done will be good. I will be thinking about it.

3) **Test the trained network on the output data :**

- Testing :
  Get the statistics of each tile in the slide.
- Performance :
  Performance of the slide is to be noted carefully.