Digital Pathology Cancer Recognition System (DPATH-CRS)

An artificial neural network (ANN) is used to classify the regions of the digital pathology slides. An ANN tries to assimilate the recognition ability of a pathologist. ANNs are actually a very non-linear mathematical function that maps the input data (such as an image) to their classes. This function has too many free parameters that need to be tuned to the specific applications; this tuning process is called training. Then this trained ANN is evaluated by asking to make a decision on some unseen data. If it can guess the correct classes, then it can be employed as a diagnostic system, and if not, the training process should be repeated.

We designed an ANN training and evaluation pipeline to make a cancer diagnosis system to assimilate the pathologists' skills. In the following, we talked about the five steps of this pipeline.

Patch Extraction

A digital pathology slide file contains several images with different resolutions but from a single specimen. Annotators, that are trained to diagnose the regions, attach the labels to some selected regions, but not all parts of a slide. So, the slides are not totally annotated, but partially annotated.

Pathologists usually look at the very subtle details of the slide to diagnose the region. Therefore, the processing is done on the highest resolution which is called level 0. Level 0 contains a very large image, such as 50K in 50K pixels. The OpenSlide library is used to read the content of a slide. Every pixel has four channels, RGBA. But the alpha channel is not usually being used. So, three channels, RGB, are enough to keep the data for every pixel.

Patches are square-shaped images that are cropped from some parts of level 0 and processed as an atom, it means that a patch is assumed to contain a single class, not different classes, and the diagnosis system assigns a single label to it. Therefore, the size (width and height) of these elemental patches are highly dependent on the application. On the other hand, the training needs annotated data and the slides are partially annotated, hence, the patches are extracted from the regions that are annotated and the un-annotated parts are not used for further processing.

For every slide in the dataset, a different directory is made with the same name as the slide file. For every possible class, a subdirectory is made in the slide directory. Patches with similar labels from a slide are stored in the corresponding label subdirectory in the corresponding slide directory. If some labels do not exist in a slide, then that subdirectory remains empty.

Annotated areas are made with freehand drawing tools, so they can have any shapes. Also, all patches in an annotated area have a single label. A surrounding rectangle of the area cropped out of level 0. A rolling window is swept from the upper left point of this surrounding rectangle to the upper right and then one step down and repeat until all the surrounding rectangle is swept. Based on the observations, the window size is selected to be 256 in 256 and the step size is 256 in both directions. So, the patches will not have overlap. Every window that has more than 50% overlap with the annotated area is assumed to be an acceptable patch and is stored as a 256 in 256 TIFF image in the corresponding patches directory.

Finding if a window is in the annotated area or not is very time-consuming, so the slide is rescaled to (1/256, 1/256) and a binary mask is made with a logical value of 1 as annotated and 0 as unannotated. Then a window with size 256 in 256 can be assumed as a single logical value, 1 means annotated and 0 means unannotated. Scaling a large image might be too time-consuming, but the surrounding rectangles are usually very smaller than the whole slide. Moreover, the scaling is done with the NearestNeighbor method which is computationally very lightweight.

At this point, all annotated areas from all slides are broken down into patches, and patches are stored in the corresponding directories.

Patch Distribution

Patch distribution is necessary to allocate different slides to the training, development, and evaluation steps of an ANN. Two considerations make the intelligent distribution of the patches crucial. First, the ratio of patches in every class in comparison to the total patches must be kept approximately similar. It means that if the ratio of patches of class 1 in the training dataset is 10%, then it is expected to have the same percentage in the development and evaluation dataset. Second, to have an open-loop (blind) development and evaluation, there must be different slides in different datasets and no similar slides should be in two different datasets.

To satisfy these two constraints, it needs an NP-hard search space. To make the whole process fast, we have to relax some constraints. The second constraint is crucial, so we relaxed the first one. Instead of maintaining all classes' ratios fixed, just the classes with the smallest percentages are tried to be kept fixed. Hence, a greedy search algorithm is developed. It tries to maintain the percentage of classes in order. It distributes the slides that contain the rarest class with predefined ratios. Then, it distributes the slides with the second rare classes, again with predefined ratios, but with priority given to training, development, and evaluation, respectively.

Therefore, a greedy algorithm with two given priorities, e.g. maintaining the rarer class percentage and the order of the training, development, and evaluation datasets distributes the slides into three datasets in a fast semi-optimal way.

Training

At this point, the slides that are used for training are known. But the number of patches in the classes is usually very different, sometimes on the scale of 1 to 10000. This situation is usually called the training in an unbalanced dataset.

Training operation is optimizing a loss function with the steepest gradient descent algorithm or one of its successors, such as Adam. Optimization of the loss function in a highly unbalanced dataset reaches severe bias toward the classes with an exceeded number of samples and it never tries to learn the patterns of classes with a few samples.

To solve the issue with the unbalanced dataset two approaches are examined; weighted loss and random sampling. The weighted loss function assumes that the prior probabilities of classes in the datasets are proportional to the number of patches in that class. This proportional weighting can be used in either original format or after a softmax operation. Either way, the ANN is forced to consider more loss for classes with less number of patches. Then some impacts of the unbalanced number of patches alleviate, but not completely because the datasets are highly unbalanced.

The second approach tries to solve this highly unbalanced impact. The idea comes from the observation that most of the patches in the outnumbered classes are very similar, especially in the patterns. So, it is rational that instead of using all the available patches in these outnumbered classes, a limited number of them are selected randomly and then train the ANN on this random subset of that class, while keeping all the rare classes as they were. This approach is called random sampling.

Combining the random subset approach with the weighted loss function solves the highly unbalanced dataset issue. But how effective a random sampling is. To find out, the ANN is trained on some random subset of the training dataset, then the one that has the best loss on the development set is selected as the best trained ANN. Since the random subset of the training dataset is usually much smaller, the training finishes fast, but the development process takes more time because the ANN is examined on all the patches on the development dataset.

The model of an ANN needs to be predetermined before the training. The convolutional layers in the image analysis with the deep learning methods are proven to be very effective for making a pyramidal representation of an image. Therefore, most image analysis systems use some convolutional filtering in a hierarchical order or in a parallel format for extracting the patterns in different resolutions and making a new set of features. Then with the help of one or two fully connected layers, they classify these features. We follow the same approach, but instead of designing and training an ANN from scratch, a pre-designed and pre-trained ANN is being used. This approach is usually known as transfer learning. It means that the ANN which is being trained on a different but large set of images retrained again on a specific set of images to be tuned on a new domain.

ImageNet is a large competition and the dataset is fully populated with natural images and too many classes. Several large ANNs are trained on this large dataset and are now openly available to be fine-tuned on different specialized domains. We use a very well-known, simple, and very effective kind of these network which is called Residual Network or ResNet. The residual name comes from the fact that the output of every layer of the neural network consists of its processed output and its raw input which is called residual. These residuals make the pyramidal multiresolution representation of the input image and cause the neural network to converge faster because they prevent vanishing gradients.

A pretrained ResNet18 on the ImageNet is trained on our training dataset. For every epoch of the training, blind evaluation is employed on a random subset of the development dataset. The best model is selected based on the least weighted loss on the random development sub-dataset. Several of these ResNet18s are trained on different random sub-datasets of the training dataset, and the one with the least weighted loss on the development dataset is kept as the best model.

Decoding

Given an unannotated whole slide, it is cropped into too many patches. These patches are shown to the neural network and the ANN classifies every patch, which is called decoding. The output of decoding is a mask of size (1/256, 1/256) of the original level0 of the slide with categorical (or indexed) pixels. This mask is usually small and can be seen as a whole in every simple image viewer.

In the decoding phase, some minor errors can be pruned. We can define a minimum size for every region. For example, a cancerous region in the size of a single patch should not be surrounded by all the normal neighbors and it is expected that some similar patches make a large area. So, a 2D moving average filter or a 2D Gaussian filter with a predefined radius is used to estimate the center patch and this estimation is being compared with the decoded label. If they are close to each other, nothing is changed, but if they are too different, the estimated output of the filter is used instead of the decoded decision.

So, it depends on how much contrast is accepted; the pure output of decoding or the smoothed decision. If the decoding output seems to be too noisy, then smoothing is necessary.

Evaluation

Evaluation needs annotated data. So, only the decoded parts that contain annotations are used. The decoded mask and annotated area are compared with each other. If they match, the detection is done perfectly, but if they differ, it is important to know the actual class is confused with which other class. This makes the confusion matrix. In a confusion matrix, actual labels are located in the rows, and detected labels are located in the columns. Every element (i, j) in the confusion matrix shows how much class i is confused by class j.

The confusion matrix is very essential to find out the effectiveness of many clinical computer-aided diagnosis (CAD) systems. In most clinical applications, the pure accurate diagnosis is not the only parameter that defines the cost. There is not a perfect diagnosis, so in every diagnosis, the cost of correct and incorrect diagnosis must be considered. For example, incorrect diagnosing of a normal tissue as malignant may cause financial loss, but diagnosing a cancerous tissue as normal may cost a human life. The confusion matrix can be combined with some clinical considerations to further process the detections. This post-processing makes a scientific model to be applicable in real-world applications and can be our future investigations.