



SMART INDIA HACKATHON-2019: SOFTWARE EDITION

IMPACT 3/3

Technology Bucket: Healthcare & Biomedical devices

Category: Software

Ministry name: Department of Atomic Energy

Problem code: RC3

Problem Statement: The goal of this project is to develop segmentation methods to fragment features of the brain like white matter, tumor etc in 3D, based on ML techniques, which require no human intervention and are robust to the low quality of medical images, with user friendly GUI for showing annotation, developed in Python using open-source ML libraries. Standard data-sets can be used for training validation and testing purposes.

Team Leader Name: B Kiran Kumar Ashish

College Code: 1-3513887662

Idea/Approach Details

IDEA/SOLUTION:

Expert's monitoring in detecting tumour/abnormalities by looking at MRI images is a time-consuming task, that too, expert cannot be present everywhere as the machine does. It is scientifically proven that if well enough of data is given to train a model, the machine actually gives a higher performance with better accuracy than the expert. Our approach is using Deep Learning model architecture for accurate detection of abnormalities and precise training of the model. We would not let the model be much computationally strain by giving the model input as 3D images, instead we would convert them into 2D format and supply as input to the model. As training the model with 3D images needs a lot of computation power and memory, we are opting for 2D conversion techniques. The machine learning/deep learning model is trained in such a way that this model can be used for any bio-medical applications for detecting abnormalities provided to do some minor changes with accordance with the application. We are using pixel-level grading a.k.a pixel-to-pixel comparison for accurate results without further expert cross-check. To avoid over-fitting, we are splitting the data into: training data, validation data and testing data. This helps to avoid over-fitting problem. Broadly, defined, dataset is first pre-processed and fed as proper structured format to ease the training phase in the model. First 3D to 2D conversion, then separate each class images into their respective folders, i.e., white matter folder contains all white matter images and healthy folder contains all healthy images. This is to ease the effort and for easy-training. A proper and complete data pre-processing is done before feeding it to the model. The model contains several hidden layers which consists of several filters and various operations are performed. The convolution layers extract features from each image that too pixel wise. We define multi-nodes as output where we get the accuracy figure of each class for that particular test image.

TECHNOLOGY STACK:

i. Data Pre-Processing:

First step in Machine Learning is to randomly read the data provided and check its properties to analyse what kind of data are we provided with. For a proper Machine Learning model to be build, with not taking much computation power and memory, data/raw data provided must be properly processed into a structured and clean format. Simply put, raw data provided must be cleaned before feeding it to the ML model. If image dataset is given along with



SMART INDIA HACKATHON-2019: SOFTWARE EDITION

groundtruths containing the “class” of each image, then we separate images into different folders based on their class names in python using standard “os.path” commands. The second case is the image labels are itself describing the class name, we can simply move them to their respective class folders by reading their label names.

Framework:



This is to ease the training phase for the model. Primarily import all necessary packages required to perform the data pre-processing phase. The images are converted to 2D from 3D since 3D are intensive memory consuming hence more computational power is required. The images are all resized to a fixed $m \times n$ size. For better network performance, image augmentation is performed. Image is randomly left and right flip, rotation, shifting, shearing, zooming and the most important one- elastic transformation.

Augmentation can produce more training data from original one. This can be done by keras on TensorFlow backend. We split our dataset into: training set (60%), validation set (20%) and testing set (20%). Validation set is used to compare the performances of the prediction algorithms that were created based on the training set. We choose the algorithm that has the best performance.

ii. Model Building:

We typically are using U-Net architecture for training our model. U-Net is specially designed for bio-medical images, proven better performance with better accuracy. Since, we need to perform pixel-level grading, U-Net does that with lot more features extracted during computation. U-Net basically “Increase the ‘what’, reduce the ‘where’!”. Architecture consists of 3 parts: Contraction path (auto-encoder), bottle neck and expansion path(decoder). Our model drives in 2-phases:

1. Classifying the image as a whole- global view (malign or benign)
2. Dividing the images into patches and classifying them- local view (pixel-level classification/grading).

- **Contraction Path/Encoder:**

Consecutive of two-times 3x3 Conv and 2x2 Max-Pooling is done. This operation helps to extract more advanced features but reduces size of feature maps.

- ✓ The first layer will have 32 filters of size 3 x 3, followed by a down-sampling (max-pooling) layer



SMART INDIA HACKATHON-2019: SOFTWARE EDITION

- ✓ The second layer will have 64 filters of size 3 x 3, followed by another down-sampling layer,
- ✓ The final layer of encoder will have 128 filters of size 3 x 3.
- **Expansion Path/Decoder:**
Consecutive of 2x2 Up-Conv and two times of 3x3 Conv is done to recover the size of segmentation map.
 - ✓ The first layer will have 128 filters of size 3 x 3 followed by a up-sampling layer
 - ✓ The second layer will have 64 filters of size 3 x 3 followed by another up-sampling layer,
 - ✓ The final layer of encoder will have 1 filter of size 3 x 3.

The max-pooling layer will down-sample the input by two times each time you use it, while the up-sampling layer will up-sample the input by two times each time it is used.

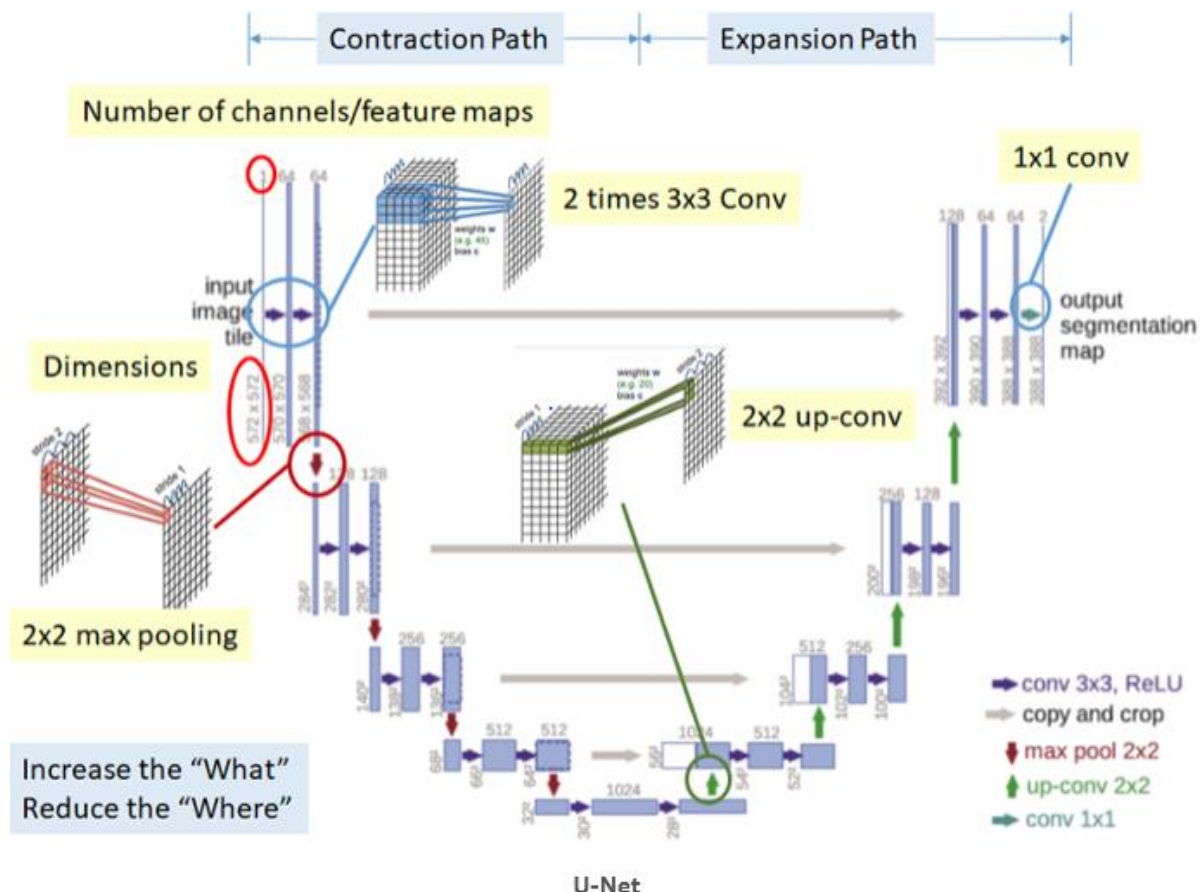


Fig. U-Net architecture

Segmentation image is retrieved as output for tumour presence images, indicating the location. Probably even heatmaps of tumour particles can be impeded for better



SMART INDIA HACKATHON-2019: SOFTWARE EDITION

understanding. Suitable activation functions such as ReLu, Softmax and padding = 'same' function, Batch Normalization, 'adam' optimizer and 'categorical_crossentropy' loss functions are used in hidden layers.

iii. Accuracy and Validation:

There is no pre-defined values to be kept in hidden layers, based on the size of dataset, we change it accordingly and validate our accuracy number. We can check our testing loss, weights, precision on testing set using sklearn library_classification report and accuracy_score. Since, we are using validation set, we can check the loss figure for cross-validation and its precision too which will be helpful for better performance of the model.

iv. Final Analysis and Comparison- GUI:

We are using tkinter library for GUI where user will understand in a better-visual way. Using Matplotlib library, we can plot accuracy score graphically which is ease for comparison. User-friendly GUI is designed where accuracy score, classification result is displayed.

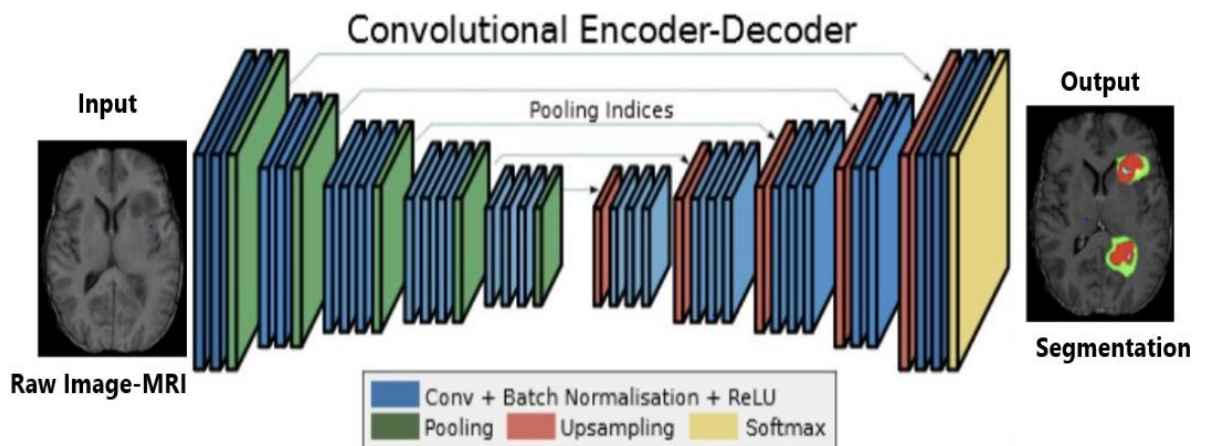


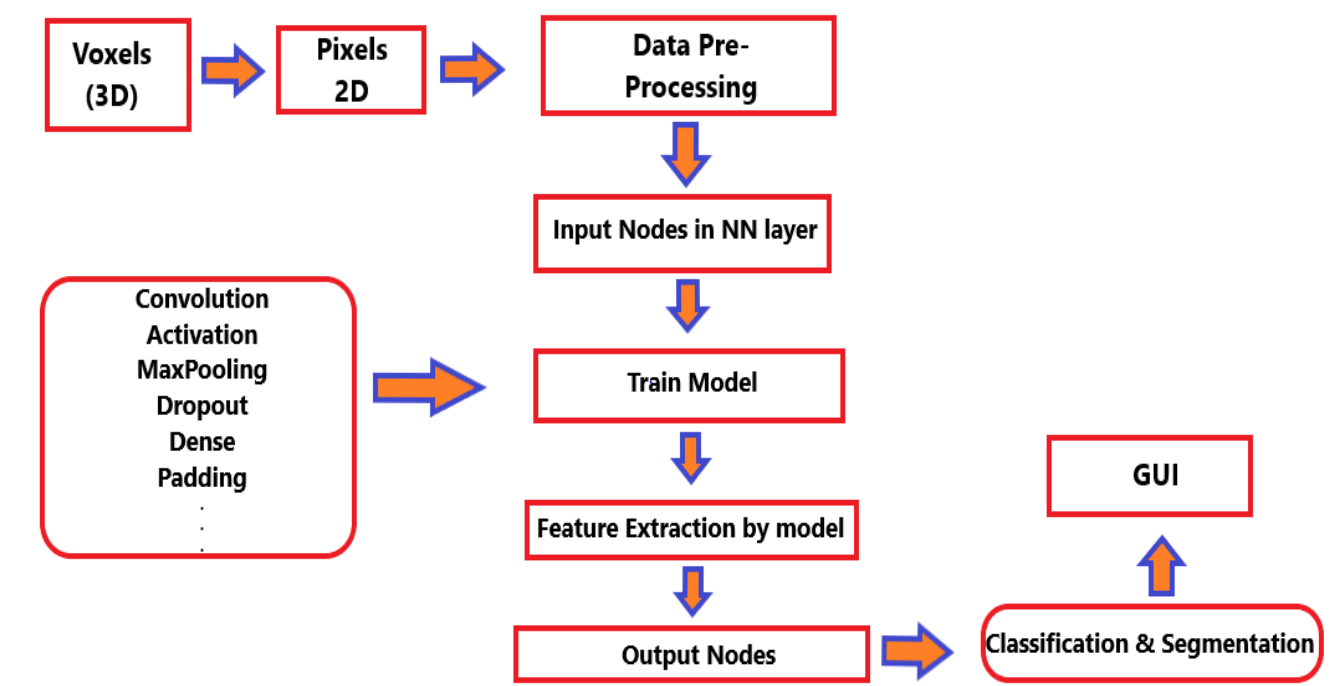
Fig. Outline theme of architecture

USE CASES:

As our framework model defines, input is dataset and program to train the model and output is a program, we are designing and training the model in such a way that, our model is applicable to any bio-medical imaging techniques for any sort of classification and segmentation/masking tasks. With slight modifications like defining label names (application specific), loading data, defining output nodes (application specific). Hence, our model can classify as global view and even does pixel level grading for local view which is helpful for accurate detection. As expert doctors cannot be present all the time everywhere, Machine Learning algorithms reduce the burden of unavailability of doctors. Places such as Nuclear research centres, Mines, even extended to hospitals and hard-core powerful research labs, our algorithm works well where diagnosis images must be given to the model and our model accurately gives the results. As AI/ML is the new hot cake in technology, several advancements have been made in bio-medical field where machine learning and deep learning models are used for precise efficiency. Therefore, our model can be also used in all diagnostic centres and hospitals for bio-medical screening applications.



Algorithm Flowchart:



DEPENDENCIES:

Our model is built on purely open-source libraries. We are using Python programming language on Jupyter notebook either from Anaconda environment or computational efficient Google Colab. Several libraries such as Keras, TensorFlow, Matplotlib, numpy, OpenCV, random, tkinter and so forth open source libraries are used.

For classification, a simple CNN classifier architecture model can be used and pixel-level grading and segmentation/masking, U-Net or Fully Convolutional Network is used. For getting better accuracy well enough, we need to adjust the number of epochs, batch size and so forth parameters depending on the size of dataset.

Pipeline:

