

Deep Learning for Lesion Segmentation

Krishna Kant Dubey

Research Intern at Philips

Training and Deploying AI Models for Lesion Segmentation

Internship Project Report

Krishna Kant Dubey

Junior Undergraduate

Department of Electrical Engineering

Indian Institute of Technology (BHU)

Mentor: Srinivasa Rao

Senior Data Scientist at Philips

Table of Content

Chapter 1: Medical Image Processing	Page 3-8
Chapter 2: TGS Salt Identification Challenge	Page 9-13
..	
Chapter 3: Literature Survey on Lesion Segmentation	Page 14-35
Chapter 4: Problem-Solving Methods	Page 36-45
Chapter 5: Model Exploration of FasterSeg Network	Page 46-51
Chapter 6: U-Net Implementation on Medical Dataset	Page 52-..
References	

CHAPTER - 1

MEDICAL IMAGE PROCESSING

A medical image is the representation of the internal structure or function of an anatomic region in the form of an array of picture elements called pixels or voxels. It is a discrete representation resulting from a sampling/reconstruction process that maps numerical values to positions of the space. The number of pixels used to describe the field-of-view of a certain acquisition modality is an expression of the detail with which the anatomy or function can be depicted. What the numerical value of the pixel expresses depends on the imaging modality, the acquisition protocol, the reconstruction, and eventually, the post-processing.

Medical Images have 4 key constituents—Pixel Depth, Photometric Interpretation, Metadata and Pixel data. These constituents are responsible for the size and resolution of the image.

Image Size = Header size (includes metadata) + Rows*Columns*Pixel Depth*(Number of Frames)

DICOM Images:

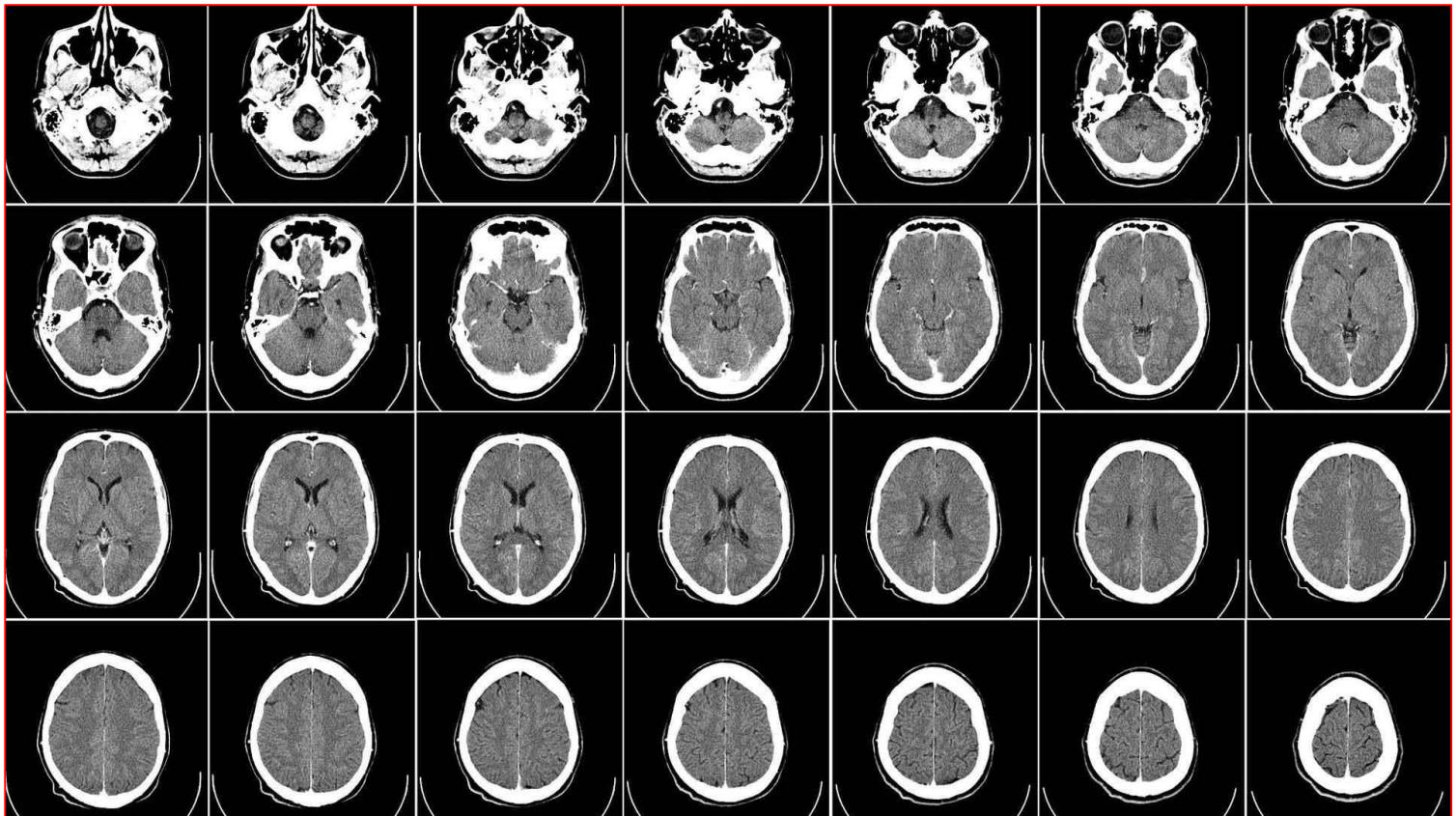
Digital Imaging and Communications in Medicine (DICOM) - an international standard related to the exchange, storage and communication of digital medical images.

While DICOM 16-bit images (with values ranging from -32768 to 32767), other 8-bit greyscale images store values 0 - 255. These value ranges in DICOM are useful, as they correlate with the **Hounsfield Scale**. Each voxel can store a large amount of information.

HOUNSFIELD SCALE - The Hounsfield unit (HU) scale is a linear transformation of the original linear attenuation coefficient measurement into one in which the radiodensity of distilled water at standard pressure and temperature (STP) is defined as zero Hounsfield units (HU), while the radiodensity of air at STP is defined as -1000 HU.

Thus, a change of one Hounsfield unit (HU) represents a change of 0.1% of the attenuation coefficient of water since the attenuation coefficient of air is nearly zero.

It is the definition for CT scanners that are calibrated with reference to water.



NOTE: Rescale intercept and rescale slope are DICOM tags that specify the linear transformation from pixels in their stored on disk representation to their in memory representation.

$$U = m \cdot SV + b$$

Where, U is in output units, m is the rescale slope, SV is the stored value, and b is the rescale intercept. Other tags describe the stored value format further, such as Bits Allocated, Bits Stored, Pixel Representation, and Sample per Pixel.

Some other commonly used medical image formats are:

Analyze

Analyze 7.5 was created at the end of 1980s as format employed by the commercial software Analyze developed at the Mayo Clinic in Rochester, MN, USA. For more than a decade, the format was the standard “de facto” for the medical imaging post-processing. The big insight of the Analyze format was that it has been designed for multidimensional data (volume). Indeed, it is possible to store in one file 3D or 4D data (the ftheth dimension being typically the temporal information). An Analyze 7.5 volume consists of two binary files: an image file with extension “.*img*” that contains the voxel raw data and a header file with extension “.*hdr*” that contains the metadata, such as number of pixels in the *x*, *y*, and *z* directions, voxel size, and data type. The header has a fixed size of 348 bytes and is described as a structure in the *C* programming language. The reading and the editing of the header require a software utility. The format is today considered “old” but it is still widely used and supported by many processing software packages, viewers, and conversion utilities.

Nifti

Nifti is a file format created at the beginning of 2000s by a committee based at the National Institutes of Health with the intent to create a format for neuroimaging maintaining the advantages of the Analyze format, but solving the weaknesses. The Nifti can in fact be thought as a *revised Analyze format*. The format fills some of the unused/little used fields present in the Analyze 7.5 header to store new information like image orientation with the intent to avoid the left-right ambiguity in brain study. Moreover, Nifti include support for data type not contemplated in the Analyze format like the unsigned 16-bit. Although the format also allows the storage of the header and pixel data in separate files, images are typically saved as a single “.*nii*” file in which the header and the pixel data are merged. The header has a size of 348 bytes in the

case of “.hdr” and “.img” data storage, and a size of 352 bytes in the case of a single “.nii” file for the presence of the additional bytes at the end, essentially to make the size a multiple of 16, and also to provide a way to store additional-metadata, in which case these 4 bytes are nonzero.

Converting DICOM to NIFTI

A popular tool for converting DICOM to NIfTI is [dicom2nii](#). A python library to read and write nifti files is [nibabel](#). If one would like to convert DICOM to Nifti, there are tools for automatic conversion (e.g. [dcm2nii](#)). Python 2 library “[dcmstack](#)” allows series of DICOM images to be stacked into multi-dimensional arrays. These arrays can be written out as Nifti files with an optional header extension (the *DcmMeta* extension) containing a summary of all the meta data from the three DICOM files. A newer library [dicom2nifti](#) is available for Python 3. I would also recommend the reader to check out the nipy project.

COORDINATE SYSTEM: DICOM defines a term: "*Reference Coordinates System*" or RCS. The RCS is a very intuitive coordinate system of the patient body: X direction is from Right to Left. So if the patient is standing in front of you with the arm raised to the sides, then X direction is from the right hand to the left hand. Y direction is from front to back or medialwise from Anterior to Posterior so if the patient is standing in front of us so we see him/her from his/her left side, then Y goes from the left to the right. Z direction goes from Feet to Head.

Format	Header	Extension	Data types
Analyze	Fixed-length: 348 byte binary format	.img and .hdr	Unsigned integer (8-bit), signed integer (16-, 32-bit), float (32-, 64-bit), complex (64-bit)
Nifti	Fixed-length: 352 byte binary format ^a (348 byte in the case of data stored as .img and .hdr)	.nii	Signed and unsigned integer (from 8- to 64-bit), float (from 32- to 128-bit), complex (from 64- to 256-bit)
Minc	Extensible binary format	.mnc	Signed and unsigned integer (from 8- to 32-bit), float (32-, 64-bit), complex (32-, 64-bit)
Dicom	Variable length binary format	.dcm	Signed and unsigned integer, (8-, 16-bit; 32-bit only allowed for radiotherapy dose), float not supported

NEED FOR PREPROCESSING OF MEDICAL IMAGES

MRI and CT scan images of the brain offer higher contrast between brain tissues. They result in High spatial resolutions and even small abnormalities can be detected easily in these images. In spite of these advantages, a major pitfall lies in small impurities glued to the image which need to be removed using preprocessing techniques. Preprocessing of medical images acquired through MRI or CT scans become the preliminary and necessary step in image processing due to several reasons. The images may have marks or labels (Film artifacts) like patient name, age and marks or they may come with several types of noises. In a wide variety of image processing applications, it becomes necessary to smooth such images as image gray level overlaps can hamper segmentation and feature extractions. These impurities can affect postprocessing techniques, making it imperative to enhance image quality and making the image suitable for further processing. Thus the primary aim of preprocessing is to remove underlying artifacts.

- **Windowing** Using the parameters window center and window width, we window out the region of interest from the CT scan
- **Resampling** Different images have different pixel spacing. In order to standardize it, we do resampling
- **Visualization** We plot the image in 3D in order to clearly find what actually we want to do with it.
- **Segmentation** If necessary, in order to reduce the problem space, we can segment the region of interest using thresholding based on HU values.
- **Normalization and zero centering** Using normalization, we set a global min/max and normalize the image as per the bounds. As a final preprocessing step, we zero center the data so that the mean value is 0.

CHAPTER - 2

UNET Segmentation for TGS Salt Identification Challenge

Several areas of Earth with large accumulations of oil and gas *also* have huge deposits of salt below the surface. But unfortunately, knowing where large salt deposits are precisely is very difficult. Professional seismic imaging still requires expert human interpretation of salt bodies. This leads to very subjective, highly variable renderings. More alarmingly, it leads to potentially dangerous situations for oil and gas company drillers.

The challenge is to build an algorithm that automatically and accurately identifies if a subsurface target is salt or not. I've done UNET segmentation for the same.

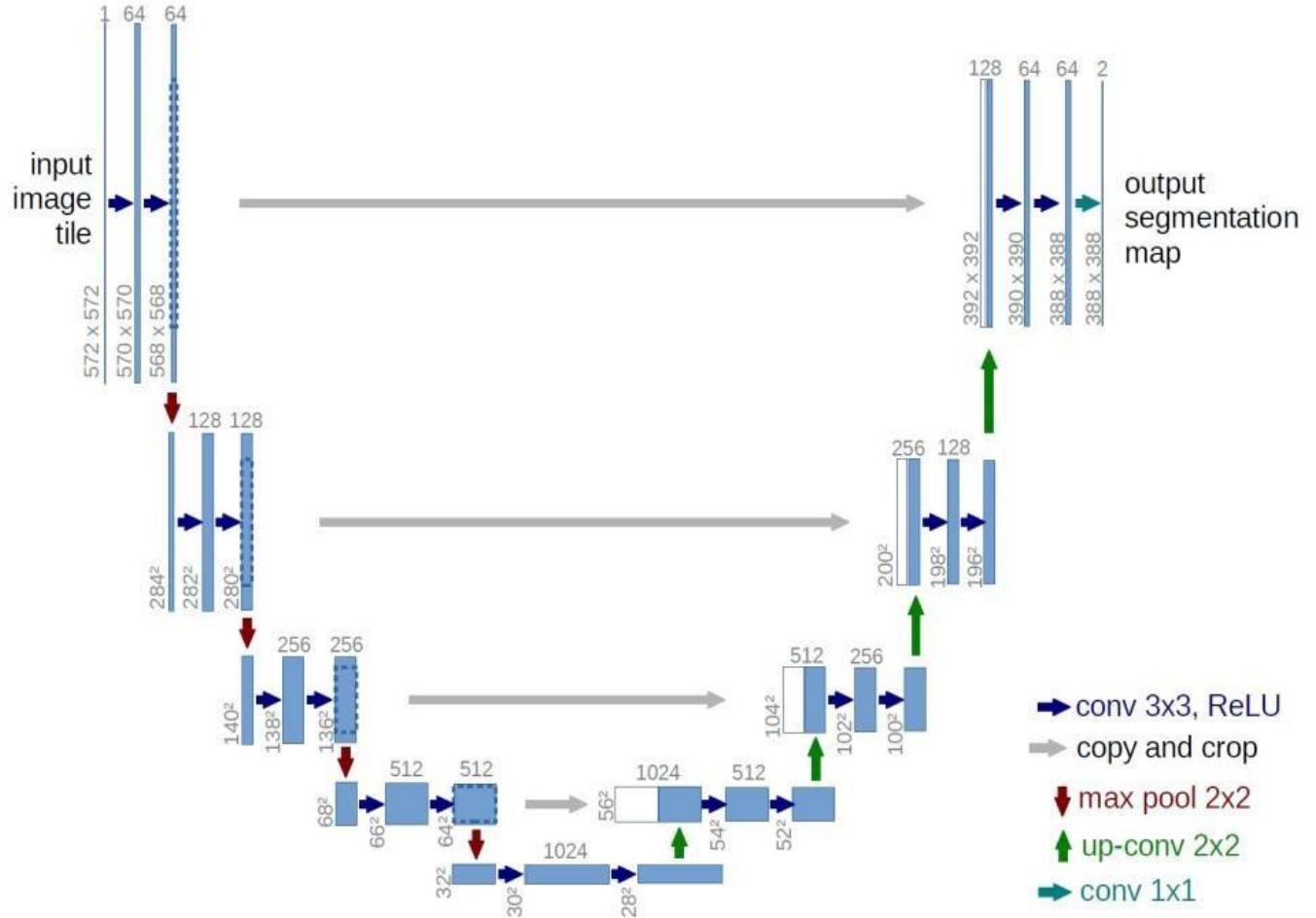
Data

The data is a set of images chosen at various locations chosen at random in the subsurface. The images are 101 x 101 pixels and each pixel is classified as either salt or sediment. In addition to the seismic images, the depth of the imaged location is provided for each image. The goal of the competition is to segment regions that contain salt.

UNET Architecture:

The U-Net architecture is built upon the Fully Convolutional Network and modified in a way that it yields better segmentation in medical imaging. Compared to FCN-8, the two main differences are: U-net is symmetric and the skip connections between the downsampling path and the upsampling path apply a concatenation operator instead of a sum. These skip connections intend to provide local information to the global information while upsampling. Because of its symmetry, the network has a large number of feature maps in the upsampling path, which allows to transfer information. By comparison, the basic FCN architecture only had number of classes feature maps in its upsampling path.

- The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3×3 convolutions, each followed by a batch normalization layer and a rectified linear unit (ReLU) activation and dropout and a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. The purpose of this contracting path is to capture the context of the input image in order to be able to do segmentation.
- Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution (“upconvolution”) that halves the number of feature channels, a concatenation with the correspondingly feature map from the contracting path, and two 3×3 convolutions, each followed by batchnorm, dropout and a ReLU. The purpose of this expanding path is to enable precise localization combined with contextual information from the contracting path.
- At the final layer a 1×1 conv is used to map each 1616-component feature vector to the desired number of classes.



```

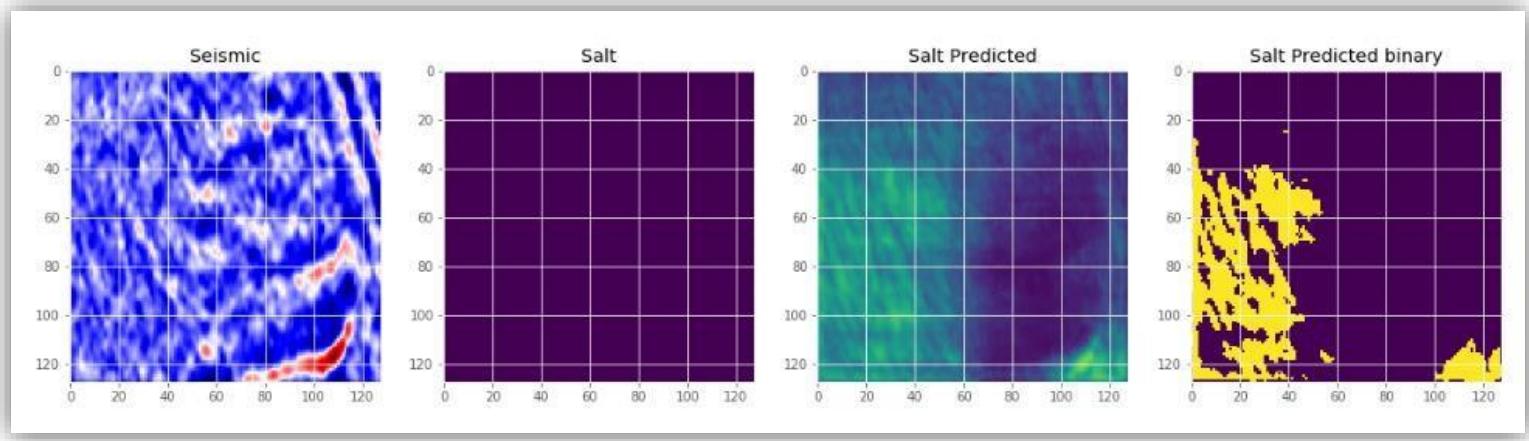
def conv2d_block(input_tensor, n_filters, kernel_size = 3, batchnorm = True):
    """Function to add 2 convolutional layers with the parameters passed to it"""
    # first layer
    x = Conv2D(filters = n_filters, kernel_size = (kernel_size, kernel_size), \
    kernel_initializer = 'he_normal', padding = 'same')(input_tensor)
    if batchnorm:
        x = BatchNormalization()(x)
        x = Activation('relu')(x)
    # second layer
    x = Conv2D(filters = n_filters, kernel_size = (kernel_size, kernel_size), \
    kernel_initializer = 'he_normal', padding = 'same')(input_tensor)
    if batchnorm:
        x = BatchNormalization()(x)
        x = Activation('relu')(x)
    return x
def get_unet(input_img, n_filters = 16, dropout = 0.1, batchnorm = True):
    # Contracting Path
    c1 = conv2d_block(input_img, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)
    p1 = MaxPooling2D((2, 2))(c1)
    p1 = Dropout(dropout)(p1)
    c2 = conv2d_block(p1, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)
    p2 = MaxPooling2D((2, 2))(c2)
    p2 = Dropout(dropout)(p2)
    c3 = conv2d_block(p2, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)
    p3 = MaxPooling2D((2, 2))(c3)
    p3 = Dropout(dropout)(p3)
    c4 = conv2d_block(p3, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)
    p4 = MaxPooling2D((2, 2))(c4)
    p4 = Dropout(dropout)(p4)
    c5 = conv2d_block(p4, n_filters = n_filters * 16, kernel_size = 3, batchnorm = \
    batchnorm)
    # Expansive Path
    u6 = Conv2DTranspose(n_filters * 8, (3, 3), strides = (2, 2), padding = 'same')(c5)
    u6 = concatenate([u6, c4])
    u6 = Dropout(dropout)(u6)
    c6 = conv2d_block(u6, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)
    u7 = Conv2DTranspose(n_filters * 4, (3, 3), strides = (2, 2), padding = 'same')(c6)
    u7 = concatenate([u7, c3])
    u7 = Dropout(dropout)(u7)
    c7 = conv2d_block(u7, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)
    u8 = Conv2DTranspose(n_filters * 2, (3, 3), strides = (2, 2), padding = 'same')(c7)
    u8 = concatenate([u8, c2])
    u8 = Dropout(dropout)(u8)
    c8 = conv2d_block(u8, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)
    u9 = Conv2DTranspose(n_filters * 1, (3, 3), strides = (2, 2), padding = 'same')(c8)
    u9 = concatenate([u9, c1])
    u9 = Dropout(dropout)(u9)
    c9 = conv2d_block(u9, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)
    outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)
    model = Model(inputs=[input_img], outputs=[outputs])
    return model

```

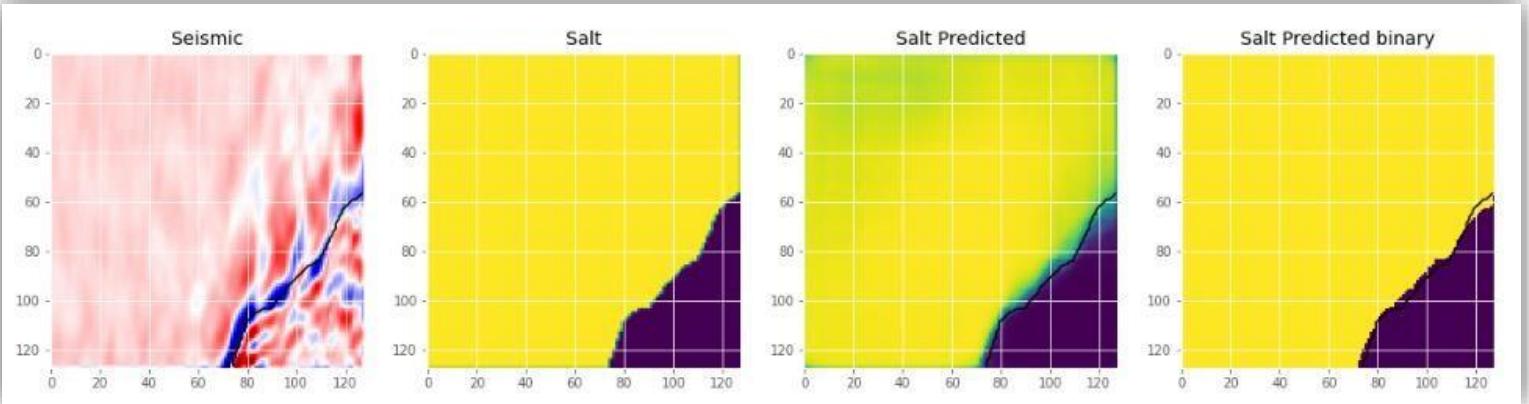
Model Features

- Model is compiled with Adam optimizer and we use binary cross entropy loss function since there are only two classes (salt and no salt).
- Used Keras callbacks to implement:
- Learning rate decay if the validation loss does not improve for 5 continues epochs.
- Early stopping if the validation loss does not improve for 10 continues epochs.
- Save the weights only if there is improvement in validation loss.
- Used a batch size of 32.

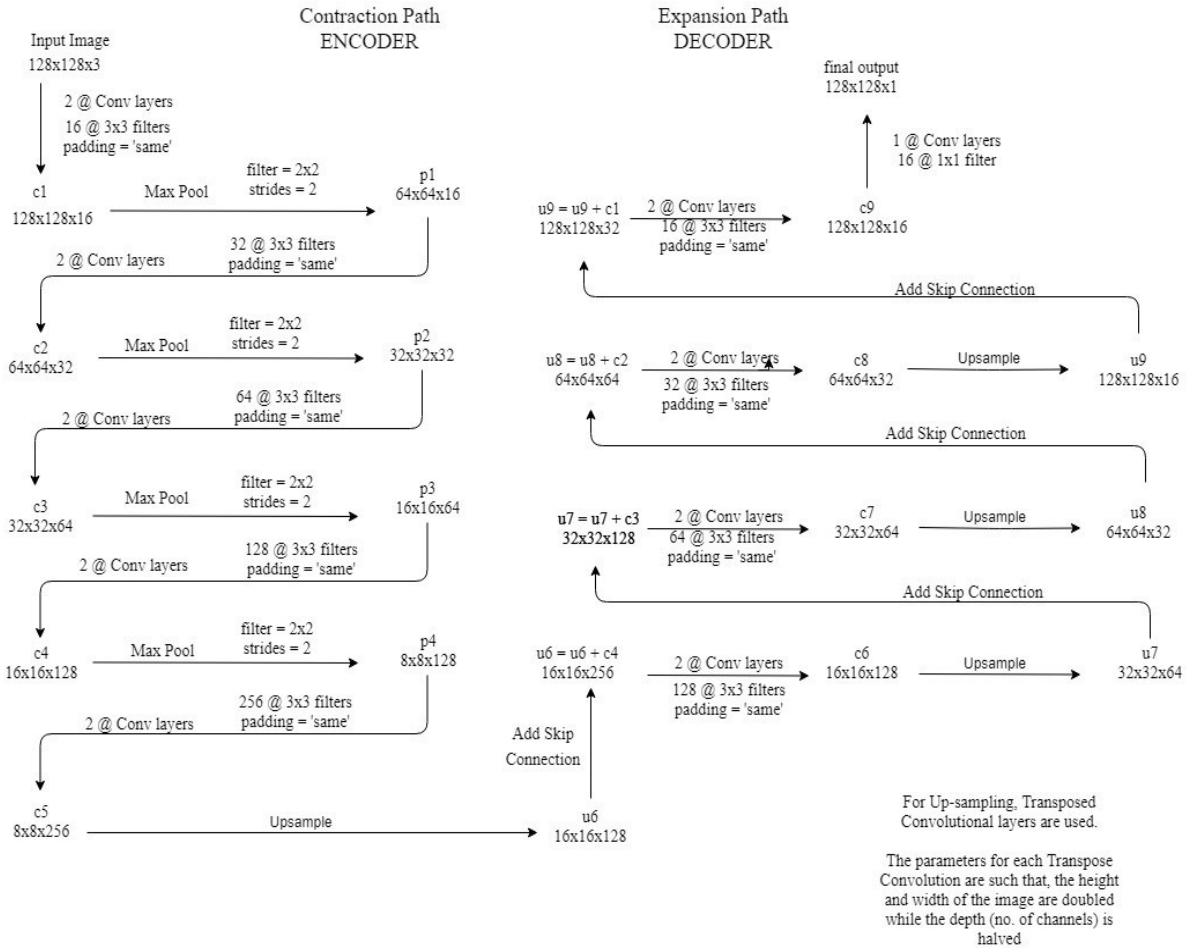
Here are some results on both training set and validation set:



Training Data Output



Validation Data Output

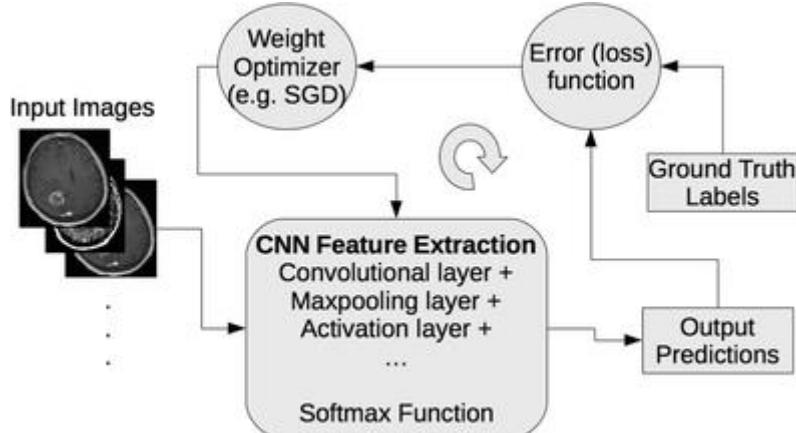


CHAPTER – 3

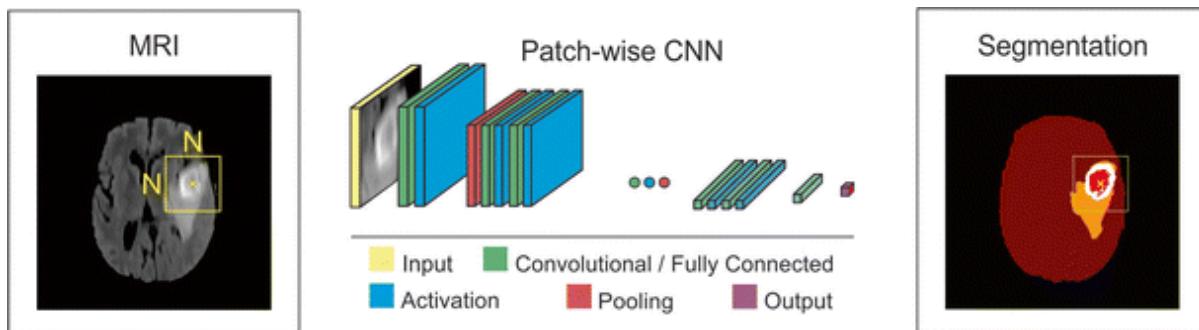
Literature Survey on Lesion Segmentation

A lesion is any damage or abnormal change in the tissue of an organism, usually caused by disease or trauma.

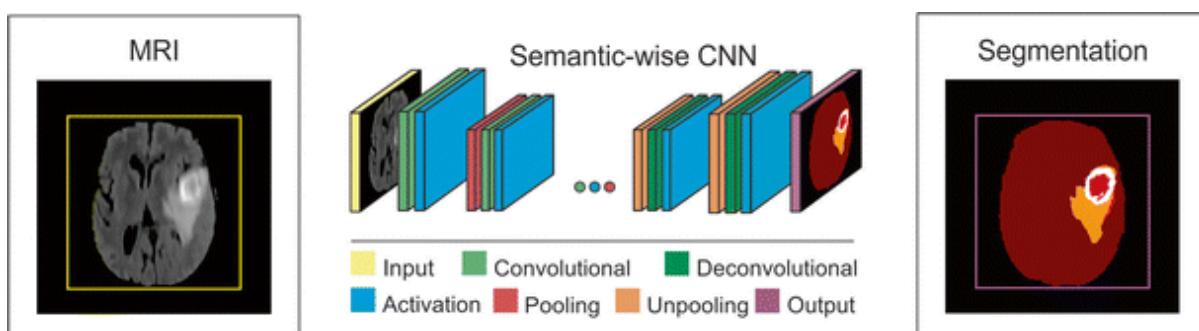
The process of delineating the boundary of a lesion from an image or image series either by use of interactive computer tools (manual) or by automated image segmentation algorithms.



A schematic representation of a convolutional neural network (CNN) training process



Schematic illustration of a patch-wise CNN architecture for brain tumor segmentation task



Schematic illustration of a semantic-wise CNN architecture for brain tumor segmentation task

Patch-Wise CNN Architecture

This is a simple approach to train a CNN algorithm for segmentation. An NxN patch around each pixel is extracted from a given image, and the model is trained on these patches and given class labels to correctly identify classes such as normal brain and tumor. The designed networks contain multiple convolutional, activation, pooling, and fully connected layers sequentially. Most of the current popular architectures use this approach. To improve the performance of patch-wise architectures, multiscale CNNs use multiple pathways, where each uses a patch of different size around the same pixel. The output of these pathways are combined by a neural network and the model trained to correctly identify the given class labels.

Semantic-Wise CNN Architecture

This type of architecture makes predictions for each pixel of the whole input image like semantic segmentation. Similar to autoencoders, they include encoder part that extracts features and decoder part that upsamples or deconvolves the higher level features from the encoder part and combines lower level features from the encoder part to classify pixels. The input image is mapped to the segmentation labels in a way that minimizes a loss function.

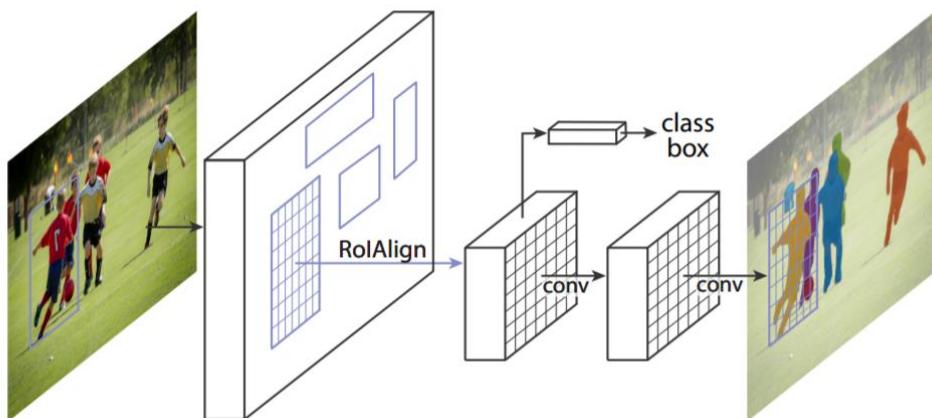
Paper: MASK R-CNN

Instance segmentation is challenging because it requires the correct detection of all objects in an image while also precisely segmenting each instance. Mask R-CNN therefore combines elements from the classical computer vision tasks of object detection, where the goal is to classify individual objects and localize each using a bounding box, and semantic segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.

Training on COCO trainval35k, Testing on COCO minival.

ADVANTAGE - Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; to this we add a third branch that outputs the object mask. The additional mask output is distinct from the class and box outputs, requiring extraction of much finer spatial layout of an object.

A unified model which can simultaneously predict boxes, segments, and keypoints at 5 fps.



Paper: EDGE DETECTION TECHNIQUES FOR IMAGE SEGMENTATION

Edges are local changes in the image intensity. Edges typically occur on the boundary between two regions. The main features can be extracted from the edges of an image. Edge detection has major feature for image analysis.

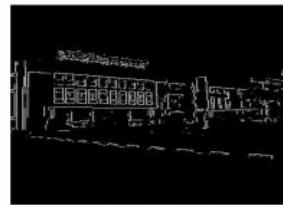
USES - object detection in various applications like medical image processing, biometrics etc.

There are many edge detection techniques in the literature for image segmentation. The most commonly used discontinuity based edge detection techniques are reviewed in this section. Those techniques are: Roberts edge detection, Sobel Edge Detection, Prewitt edge detection, Kirsh edge detection, Robinson edge detection, Marr-Hildreth edge detection, LoG edge detection and Canny Edge Detection.

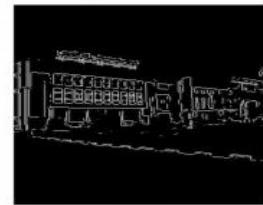
RESULT - Roberts, Sobel and Prewitt results actually deviated from the others. Marr-Hildreth, LoG and Canny produce almost same edge map. Kirsch and Robinson edge maps are almost same. It is observed from the figure, Canny result is superior by far to other results.



Original



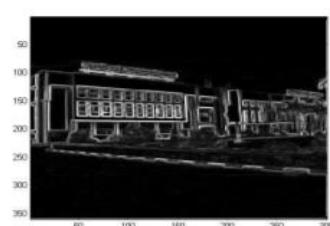
Roberts



Sobel



Prewitt



Kirsch



Robinson



Marr-Hildreth



LoG



Canny

Paper: Invariant Information Clustering for Unsupervised Image Classification and Segmentation

METHOD USED - Invariant Information Clustering, which can be used to cluster any kind of unlabelled paired data by training a network to predict cluster identities is applied to image clustering and segmentation, by generating the required paired data using random transformations and spatial proximity.

Tested on STL10, which is ImageNet adapted for unsupervised classification, as well as CIFAR10, CIFAR100-20 and MNIST. The main setting is pure unsupervised clustering (IIC) but also tested two semisupervised settings: finetuning and overclustering.

For semi-supervised learning, established a new state-of-the-art on STL10 out of all reported methods by finetuning a network trained in an entirely unsupervised fashion with the IIC objective. COCO stuff 3 and Potsdam3 datasets are used.

Algorithm	Description	Advantages	Limitations	Papers
Region-Based Segmentation	Separates the objects into different regions based on some threshold value(s).	a. Simple calculations b. Fast operation speed c. When the object and background have high contrast, this method performs really well	When there is no significant grayscale difference or an overlap of the grayscale pixel values, it becomes very difficult to get accurate segments.	Region-based semantic segmentation with end-to-end training
Edge Detection Segmentation	Makes use of discontinuous local features of an image to detect edges and hence define a boundary of the object.	It is good for images having better contrast between objects.	Not suitable when there are too many edges in the image and if there is less contrast between objects.	Classification With an Edge: Improving Semantic Image Segmentation with Boundary Detection
Segmentation based on Clustering	Divides the pixels of the image into homogeneous clusters.	Works really well on small datasets and generates excellent clusters.	a. Computation time is too large and expensive. b. k-means is a distance-based algorithm. It is not suitable for clustering non-convex clusters.	Invariant Information Clustering for Unsupervised Image Classification and Segmentation
Mask R-CNN	Gives three outputs for each object in the image: its class, bounding box coordinates, and object mask	a. Simple, flexible and general approach b. It is also the current state-of-the-art for image segmentation	High training time	Mask R-CNN

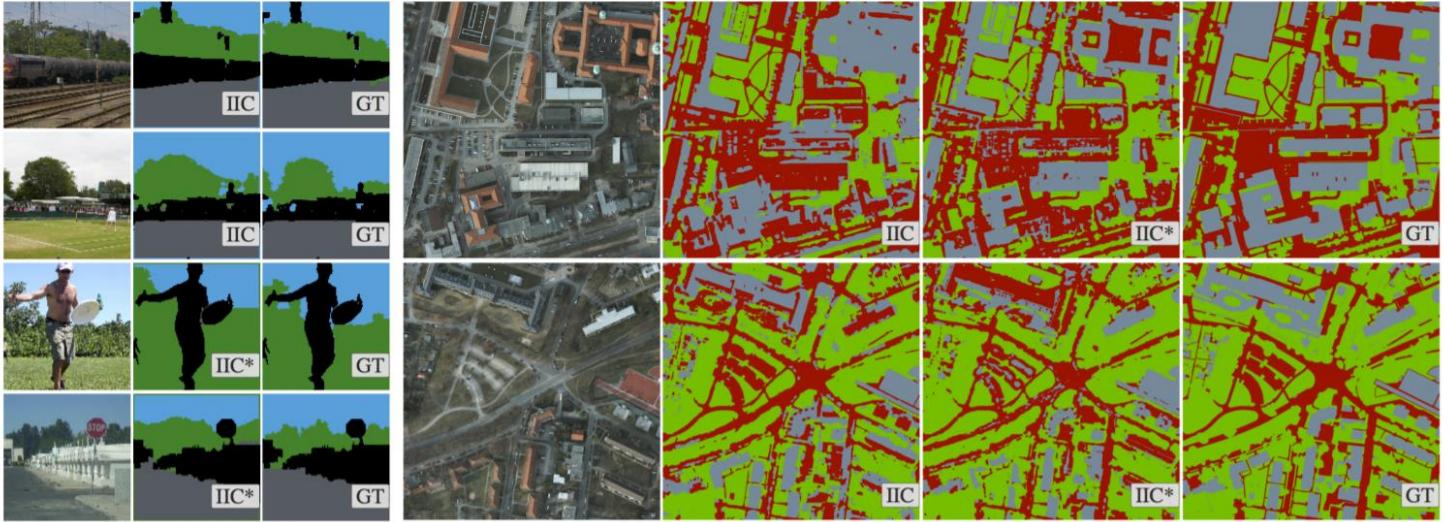


Figure 7: **Example segmentation results (un- and semi-supervised).** Left: COCO-Stuff-3 (non-stuff pixels in black), right: Potsdam-3. Input images, IIC (fully unsupervised segmentation) and IIC* (semi-supervised overclustering) results are shown, together with the ground truth segmentation (GT).

Conclusion: While some modifications to the conventional edge and clustering techniques can optimize them to give better results, the Mask R-CNN remains the most powerful state-of-the-art approach for image segmentation. The flexibility of CNN architecture allows it to integrate with other techniques for desired results.

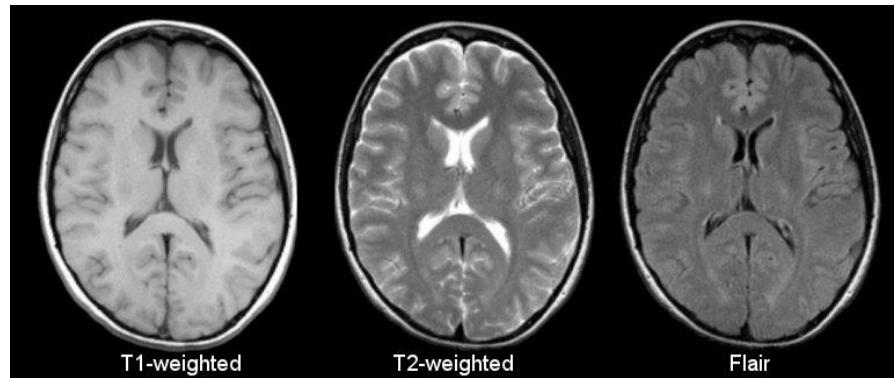
Segmentation of Brain Lesions

Quantitative analysis of brain lesions include measurement of established imaging biomarkers such as the largest diameter, volume, count, and progression, to quantify treatment response of the associated diseases, such as brain cancer, MS, and stroke. Reliable extraction of these biomarkers depends on prior accurate segmentation. Despite the significant effort in brain lesion segmentation and advanced imaging techniques, accurate segmentation of brain lesions remains a challenge. Many automated methods have been proposed for lesion segmentation problem, including unsupervised modeling methods that aim to automatically adapt to new image data supervised machine learning methods that, given a representative dataset, learn the textural and appearance properties of lesions, and atlas-based methods that combine both supervised and unsupervised learning into a unified pipeline by registering labeled data or a known cohort data into a common anatomical space. Several review papers provide overview of classical methods for brain tumor segmentation, and MS lesion segmentation.

In the field of lesion segmentation, numerous methods have been designed to automatically delineate lesions. These can be classified as supervised or unsupervised methods. The performance of supervised methods was reported to be high but the main caveat of these methods is the choice of the training set which may not include the full extent of pathology which exists in the population, thus producing a less-than-optimal representation of pathological variability. Caution is further warranted: with problems such as lesion segmentation, for which inter- and intra-rater variability is high, clinical definition can be disputed. Alternative strategies, whose strengths and weaknesses have been discussed in, include the use of atlases, the application of tissue segmentation to drive the lesion detection or the direct application of empirical rules. Most of the methods however make use of complementary information

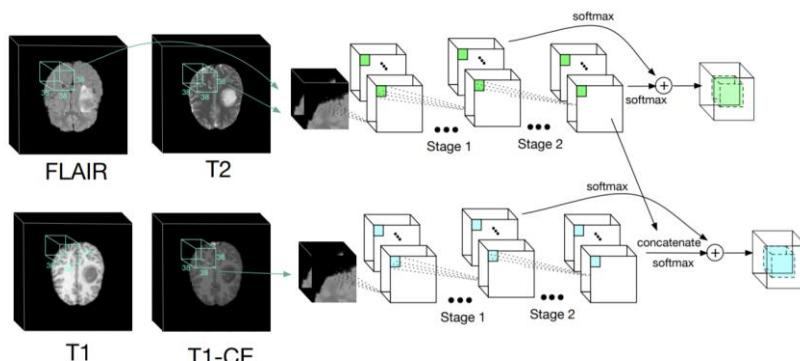
obtained using different structural MR acquisition sequences. For example, the T1-weighted (T1) images are known to provide a good contrast between the healthy tissues while FLuid Attenuated Inversion Recovery (FLAIR) sequences are widely used to distinguish pathologies present in the white matter. In addition, FLAIR images offer a good contrast between the CSF and the lesions but is known to suffer from acquisition artefacts such as pulsatile flow in the CSF and overestimation of the demyelination. Multimodal information may be used directly in a joint manner or following a multi-steps scheme, making use of specific modalities to obtain either the tissue or the final lesion segmentation. It has been noted that, since the different acquisition sequences represent different physical behaviors, their combined use makes the definition of the lesions even more complex. A spatially weighted model characterising the amount of information provided by each modality has also been proposed. Finally, some methods avoid problems related to the registration of multiple acquisition sequences by using one unique acquisition sequence.

	TR (msec)	TE (msec)
T1-Weighted (short TR and TE)	500	14
T2-Weighted (long TR and TE)	4000	90
Flair (very long TR and TE)	9000	114



Paper: MRI Tumor Segmentation with Densely Connected 3D CNN

A new approach for brain tumor segmentation in MRI scans. DenseNet was initially introduced for the image classification problem. In this work, the potential of densely connected blocks in 3D segmentation tasks is explored. Compared with traditional networks with no skip connections, the improved information flow extracts better features and significantly help the optimization. Multi-scale receptive fields are taken into account to accurately classify voxels. The model is made to predict 123 voxels in one iteration to increase the efficiency. Experimental results indicate that the proposed model performs on par with the state-of-the-art models without advanced tricks. The ablation study substantiates the effectiveness of each component of the whole architecture. It is concluded that the proposed model has great potential for MRI segmentation or other medical image segmentation tasks.



: The overview of our segmentation approach with densely connected 3D CNN hierarchical structure.

Paper: X-Net: Brain Stroke Lesion Segmentation Based on Depthwise Separable Convolution and Long-range Dependencies

Performance of conventional deep learning methods are limited due to the insufficient training of a large number of parameters, which sometimes fail in capturing long-range dependencies. To address these issues, a depthwise separable convolution based X-Net is proposed that designs a nonlocal operation namely Feature Similarity Module (FSM) to capture long-range dependencies. The adopted depth-wise convolution allows to reduce the network size, while the developed FSM provides a more effective, dense contextual information extraction and thus facilitates better segmentation. The effectiveness of X-Net was evaluated on an open dataset Anatomical Tracings of Lesions After Stroke (ATLAS) with encouraging performance achieved compared to other six state-of-the-art approaches.

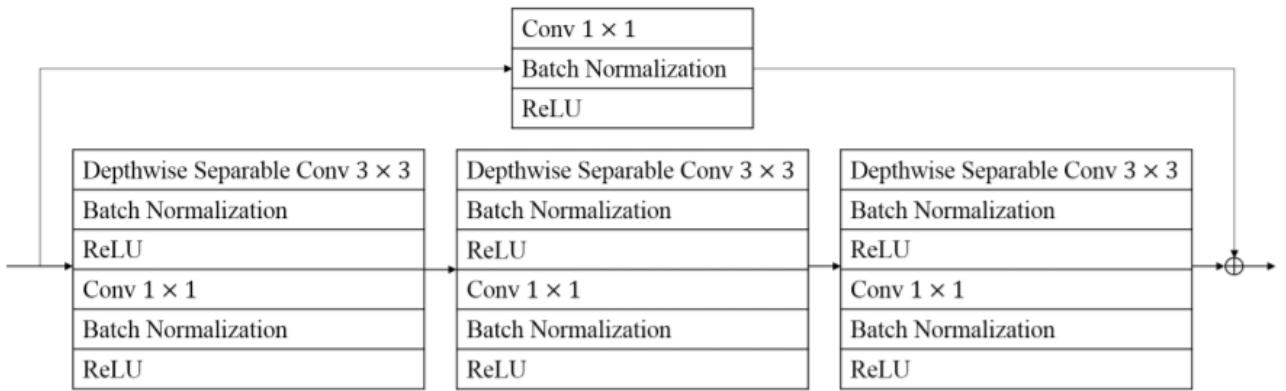


Fig. 3. The details of X-block.

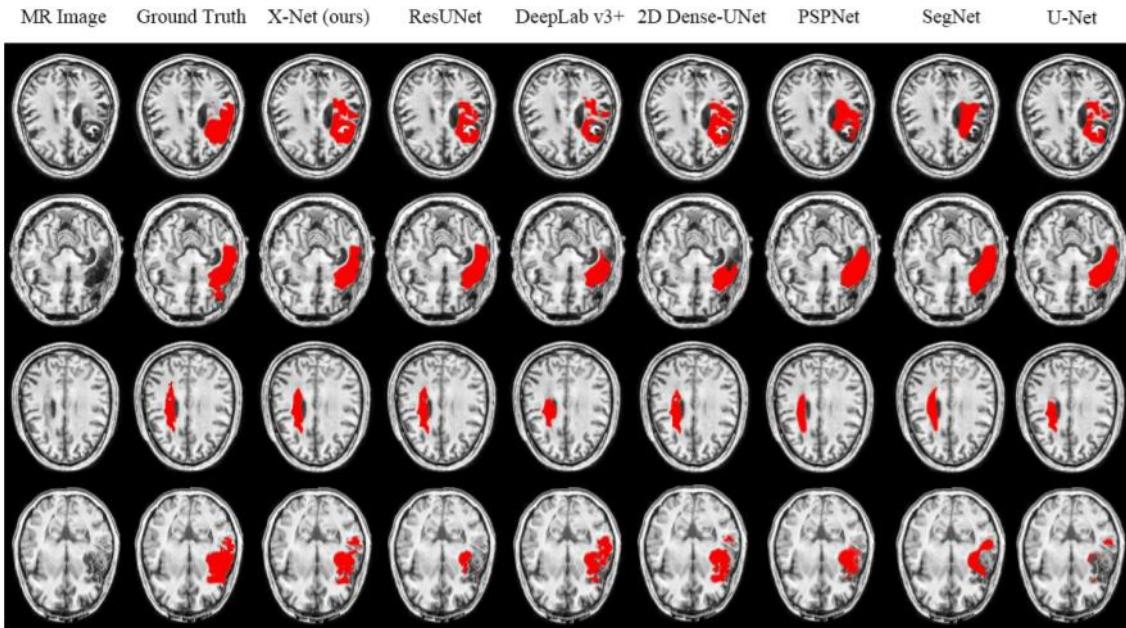


Fig. 4. Examples of segmentation results on ATLAS dataset.

Paper: DALS: Deep Active Lesion Segmentation

The developed framework combines the capabilities of the CNN and the level-set ACM (Active Contour Models) to yield a robust, fully automatic medical image segmentation method that produces more accurate and detailed boundaries compared to competing state-of-the-art methods in lesion segmentation of multiple organs. The DALS framework includes an encoder-decoder that feeds a level-set ACM with per-pixel parameter functions. We evaluated the framework in the challenging task of lesion segmentation with a new dataset, MLS (Multiorgan Lesion Segmentation), which includes a variety of images of lesions of various sizes and textures in different organs acquired through multiple imaging modalities. The results demonstrate favorable performance compared to competing methods, especially for small training datasets.

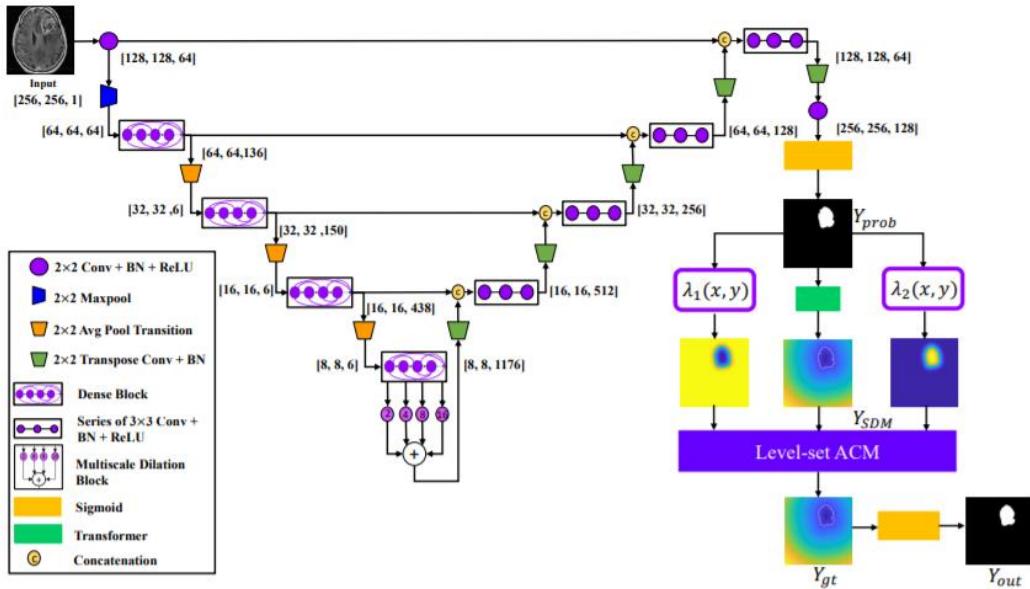


Fig. 2: The proposed DALS architecture. DALS is a fully automatic framework without the need for human supervision. The CNN initializes and guides the ACM by its learning local weighted parameters.

Paper: Spatio-temporal Learning from Longitudinal Data for Multiple Sclerosis Lesion Segmentation

Two proposed methods for detecting MS lesions: (a) Longitudinal Network: longitudinal scans are concatenated and given to the segmentation model to implicitly use the structural differences (b) Multitask Longitudinal Network: The network is trained with an auxiliary task of deformable registration between two longitudinal scans, to explicitly guide the network towards using spatio-temporal changes.

A common pitfall in multi task learning is the imbalance of different tasks which leads to under-performance of multitask learning compared to single tasks. So loss functions should be normalized.

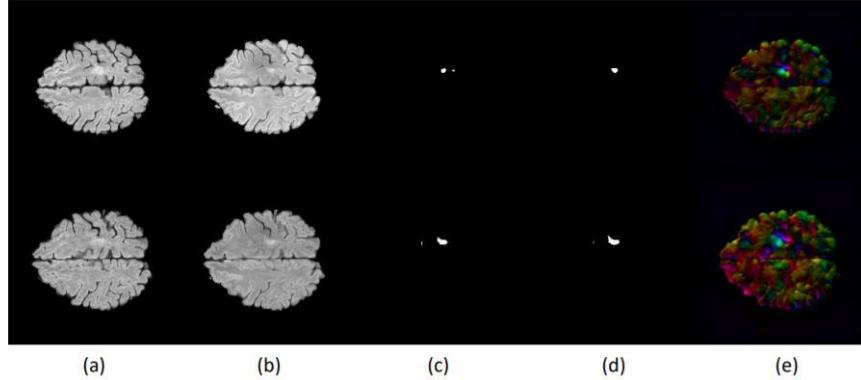


Fig. 2. Visualisation of MS lesion's structural change in two longitudinal MR FLAIR scans. Each row presents data from one patient. (a) is the scan from the first time-point, (b) is the scan from the follow up study, (c) visualizes ground truth of MS lesions on the follow up image, (d) shows the predicted segmentation mask of Multitask Longitudinal Network on the follow up image, (e) represents the predicted displacement field between the two scans using the registration module of our multitask method.

Table 2. Comparison of different approaches on the clinical dataset. Our methods are shown in bold letters. For LFPR and VD, lower is better.

Method	DSC	PPV	LTPR	LFPR	VD	Overall Score
Multitask Longitudinal Network	0.695	0.771	0.680	0.212	0.221	0.745
Longitudinal Network with Pretraining	0.692	0.777	0.660	0.205	0.232	0.739
Longitudinal Network	0.694	0.752	0.654	0.227	0.227	0.731
Static Network	0.684	0.762	0.647	0.250	0.247	0.718
Longitudinal Siamese Network (Tiramisu) [5]	0.684	0.777	0.614	0.194	0.245	0.726

Paper: Efficient Multi-Scale 3D CNN with fully connected CRF for Accurate Brain Lesion Segmentation

To overcome the computational burden of processing 3D medical scans, a dual pathway, 11 layers deep 3D CNN is proposed as an efficient and effective dense training scheme which joins the processing of adjacent image patches into one pass through the network while automatically adapting to the inherent class imbalance present in the data. The method consists of two main components, a 3D CNN that produces highly accurate, soft segmentation maps, and a fully connected 3D CRF that imposes regularization constraints on the CNN output and produces the final hard segmentation labels.

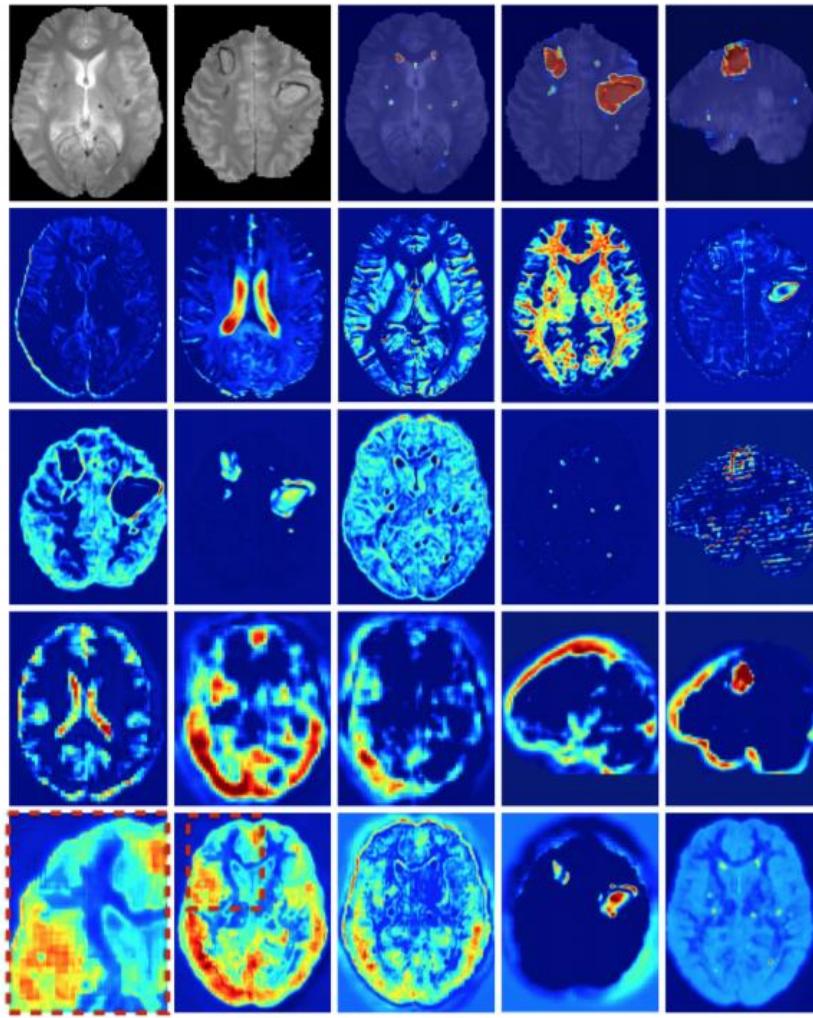


Figure 14: (First row) GE scan and DeepMedic's segmentation. (Second row) FMs of earlier and (third row) deeper layers of the first convolutional pathway. (Fourth row) Features learnt in the low-resolution pathway. (Last row) FMs of the two last hidden layers, which combine multi-resolution features towards the final segmentation.

Paper: An effective method for computerized prediction and segmentation of multiple sclerosis lesions in brain MRI

MS lesions do not appear within skull portion of human head. So, a skull removal methodology is used to improve the MS lesion segmentation on input image $I_N(x, y)$ and store it into $I(x, y)$. Then decompose the method in two key steps as background generation, and binarization. Final step for MS lesions detection and segmentation is used after background and binarization. The details methodology of binarization to correctly identify the MS lesion is described in second stage. Binarization using global threshold gives us the MS lesions with few other normal tissues as output. Finally, MS lesions are identified by selecting only those regions which are present in binarization but not present in background image for accurate detection and segmentation.

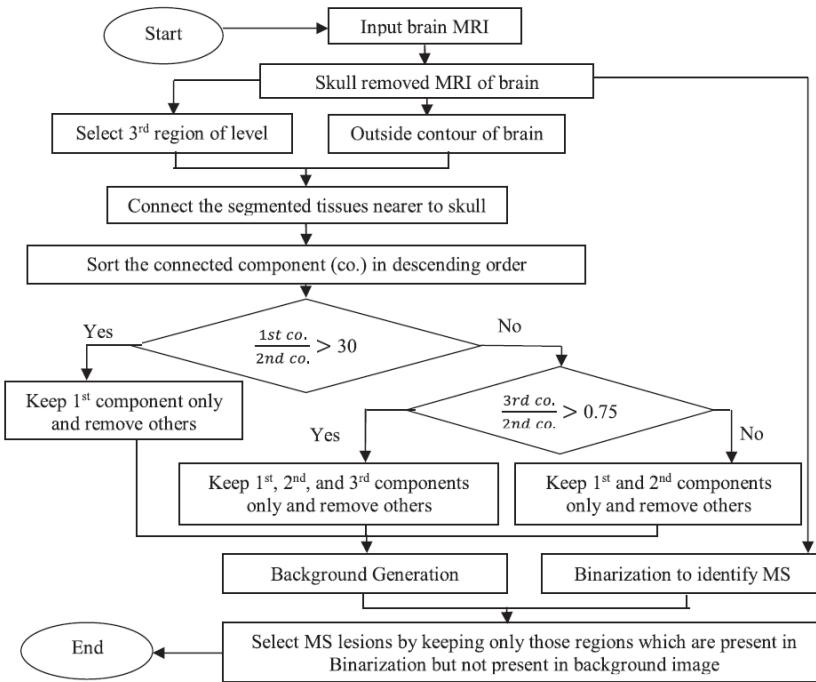


Fig. 1. Flowchart of proposed methodology.

Paper: Integration of morphological preprocessing and fractal based feature extraction with recursive feature elimination for skin lesion type classification

The method is proposed for the identification of each of these diseases using recursive feature elimination (RFE) based layered structured multiclass image classification technique. Prior to the classification, different quantitative features have been extracted by analysing the shape, the border irregularity, the texture and the color of the skin lesions, using different image processing tools. Primarily, a combination of gray level co-occurrence matrix (GLCM) and a proposed fractal-based regional texture analysis (FRTA) algorithm has been used for the quantification of textural information. The performance of the framework has been evaluated using a layered structure classification model using support vector machine (SVM) classifier with radial basis function (RBF).

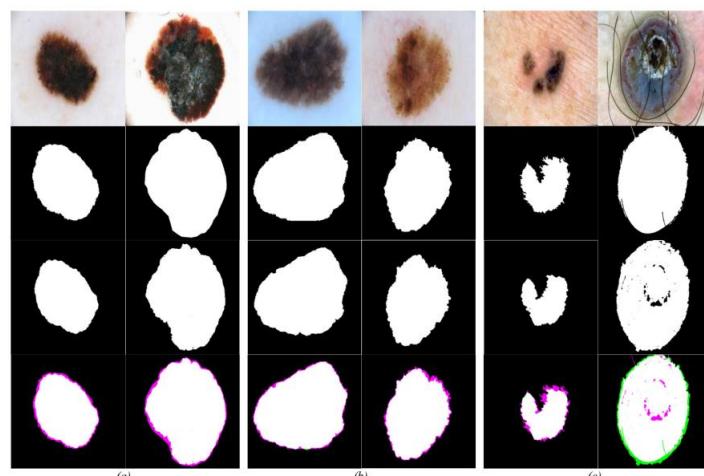


Fig. 9. Segmentation performance analysis of the proposed methodology for (a) melanoma, (b) nevus and (c) BCC images. First Row: Original RGB images; Second Row: Ground Truth (GT) images; Third Row: Segmented images using proposed algorithm; Fourth Row: Similarity measure between the GT images and the resultant Segmented images of the proposed algorithm.

LITERATURE SURVEY

Paper Name	Dataset Used	Metrics	Comments	Methods
MRI Tumor Segmentation with Densely Connected 3D CNN	BraTS 2017 Challenge (MRI Scans of 210 patients) Dataset for training and testing	Dice Dice = $2TP / (FN + FP + 2TP)$ Achieved 0.72, 0.83 and 0.81 for the complete tumor, tumor core and enhancing tumor segmentation	The sub-regions of tumor are also segmented. N4ITK bias correction is applied to remove bias field distortions. Experimental results indicate that the proposed model performs on par with the state-of-the-art models without advanced tricks. The ablation study substantiates the effectiveness of each component of the whole architecture. CODE AVAILBLE HERE	Concatenated patches from FLAIR and T2, and concatenate patches from native (T1) and post contrast T1-weighted (T1-CE) to generate separate inputs for two independent feature extractors, which share the same structure. Trained the model with a patch-wise training schema to mitigate the class imbalance problem.
X-Net: Brain Stroke Lesion Segmentation Based on Depthwise Separable Convolution and Long-range Dependencies	ATLAS Dataset having 229 T1-weighted normalized 3D MR images with diverse lesions manually segmented by specialists and collected from 11 cohorts worldwide. Each 3D image has 189 slices, and the size of each slice is 233×197.	Dice=0.4867, IoU=0.3723, Precision=0.6000, Recall=0.4752	Long Short Term Memory (LSTM) based network – Feature Similarity Module is used to capture complex spatial contextual information. Proposed X-Net model can segment the brain stroke lesions in T1-weighted MR images very well. Furthermore, the X-Net has a significant smaller number of trainable parameters (15.1M), which could better fit the clinical requirements on fast image analysis. CODE AVAILABLE HERE	Used the strategy of reduce learning rate on plateau to reduce learning rate automatically, in which the learning rate is reduced by a constant factor when the performance metric plateaus on the validation set. The initial learning rate is set to 0.001. Selected a sum of Dice loss and Cross Entropy loss as the loss function. The batch size for training is set to 8, and the maximum number of epochs is set to 100.
DALS: Deep Active Lesion Segmentation	85% of MLS Dataset (732 total images from brain, lung, liver CT and MR) images for training, 10% for testing, and 5% for validation	Brain MR: Dice= 0.888 ± 0.0755 Hausdorff= 2.322 ± 0.824 Lung CT: Dice= 0.869 ± 0.113 Hausdorff= 2.095 ± 0.623 Liver MR: Dice= 0.894 ± 0.0654	DALS performs well for different image characteristics, including low contrast lesions, heterogeneous lesions, and noise. DALS outperformed the manually-initialized ACM and its backbone CNN in all metrics across all evaluations on every organ. It well assists in capturing local and global context for highly accurate lesion localization.	The framework combines the capabilities of the CNN and the level-set ACM to yield a robust, fully automatic medical image segmentation method that produces more accurate and detailed boundaries compared to competing state-of-the-art methods. The DALS framework includes an encoder-decoder that feeds a level-set ACM

		Hausdorff= 1.298 ± 0.434 Liver CT: Dice= 0.846 ± 0.090 Hausdorff= 3.113 ± 0.747		with per-pixel parameter functions.
Spatio-temporal Learning from Longitudinal Data for Multiple Sclerosis Lesion Segmentation	1)ISBI Longitudinal MS Lesion Segmentation Challenge dataset containing 3T MR images over time. 5 subjects in training set and 14 subjects in test set with 3 to 5 follow-up images per subject. 2)Data of 40 patients as training set (30 train, 10 validation). Data of 30 patients as testing set.	0.125DSC + 0.125PPV + 0.25VC + 0.25LTPR + 0.25(1 – LFPR) Dice Similarity Coefficient (DSC), Positive Predictive Value (PPV), Pearson's correlation coefficient (VC), Lesion-wise True Positive Rate (LTPR), Lesion-wise False Positive Rate (LFPR) Scores: Multitask Longitudinal Network=91.97 Longitudinal Network with Pretraining=91.96 Longitudinal Network=92	Uses the data from multiple time-points. The proposed Longitudinal Network and Multitask Longitudinal Network achieve higher ISBI score compared with both Static Network and Longitudinal Siamese Network. The proposed approaches on two longitudinal MS lesion datasets and showed that incorporating spatio-temporal information into segmentation models improves the segmentation performance. CODE AVAILABLE HERE	Concatenated the multimodal inputs from two time-points and the result is given to the network as input. The early fusion of inputs allows for proper capturing of the differences between inputs from two time-points in the encoder. The second approach is complementary to the baseline longitudinal architecture and further enforces the use of structural changes through time.
Efficient Multi-Scale 3D CNN with fully connected CRF for Accurate Brain Lesion Segmentation	Training on BRATS and ISLES Datasets. Testing on images acquired from multiple clinical centres (As a result, performance is decreased as testing data differs significantly	DSC= 63.0(16.3) Precision=67.7(18.2) Sensitivity=63.2(16.7) ASSD= 4.02(2.54) Haussdorf= 55.68(15.93)	DeepMedic, a 3D CNN architecture for automatic lesion segmentation that surpasses state-of-the-art on challenging data is proposed. Application of a 3D fully connected CRF on medical data, employed as a post-processing step to refine the network's output, a method that has also been shown promising for processing 2D natural images.	The segmentation of each voxel is performed by taking into account the contextual information that is captured by the receptive field of the CNN when it is centred on the voxel. The spatial context is providing important information for being able to discriminate voxels that otherwise appear very similar when considering only local appearance.

	from training data)		For the multi-class problem it is challenging to find a global set of parameters for the CRF which can consistently improve the segmentation of all classes.	
An effective method for computerized prediction and segmentation of multiple sclerosis lesions in brain MRI	Training on 'Whole Brain ATLAS' Image Database.	The average Kappa index is 94.88%, Jaccard index is 90.43%, correct detection ration is 92.60284%, false detection ratio is 2.55% and relative area error is 5.97% for proposed method.	Two different strategies are employed to improve the lesion detection and segmentation. The use of background subtraction concept deals with higher level of MS lesion segmentation which does not generates any spurious lesions. The use of entropy and standard deviation based binarization increased detection efficiency. Thus, the combination of two concepts gives higher accuracy with lower error rate compared to other recent AMMR, AC, and MGC method. The method accurately identifies the size, number of lesions and location of lesion detections.	The use of background subtraction concept deals with higher level of MS lesion segmentation which does not generates any spurious lesions. The use of entropy and standard deviation based binarization increased detection efficiency. Thus, the combination of two concepts gives higher accuracy with lower error rate compared to other recent AMMR, AC, and MGC method. The method accurately identifies the size, number of lesions and location of lesion detections as a radiologist perform. The adaptability of the proposed method creates a number of potential opportunities in clinical practice for the detection of MS lesions in MRI.
Integration of morphological preprocessing and fractal based feature extraction with recursive feature elimination for skin lesion types classification	16655 Dermoscopic images for training, 9211 images for testing	Pixel Level Sensitivity, Specificity, Accuracy (ACU) Jaccard Similarity Index (JSI), Dice Similarity Coefficient (DSC) are respectively 0.9134, 0.9510, 0.9396, 0.8060, and 0.8889	Among the wide variety of skin abnormalities, only images for three classes belonging to melanocytic and epidermal skin lesion family have been considered for analysis and classification. Testing the proposed methodology on dermoscopic images of cases with wider varieties of abnormalities, would have given a better measure of its prowess. The proposed integrated scheme for multiclass skin disease	Hair artifacts removal from the dermoscopic images. The proposed algorithm has efficiently removed both the thick and thin hair like structures. Introduced circular kernel as a structuring element for morphological operations. Fractal based regional texture analysis (FRTA) algorithm subdivided the lesion area into smaller sub regions depending on the textural complexity, which

			<p>identification has introduced a simple and effective systematic approach for dermoscopic image preprocessing, lesion segmentation, feature extraction and classification. A fractal based border irregularity measurement and regional texture analysis technique has been introduced for the efficient feature extraction from Dermoscopic images along with some well-established feature extraction methods</p>	<p>has been quantified by estimating the fractal dimension. Layered structured multiclass classification technique used for the identification of skin diseases along with the segregation of misclassified images for further consideration and decision-making.</p>
--	--	--	---	---

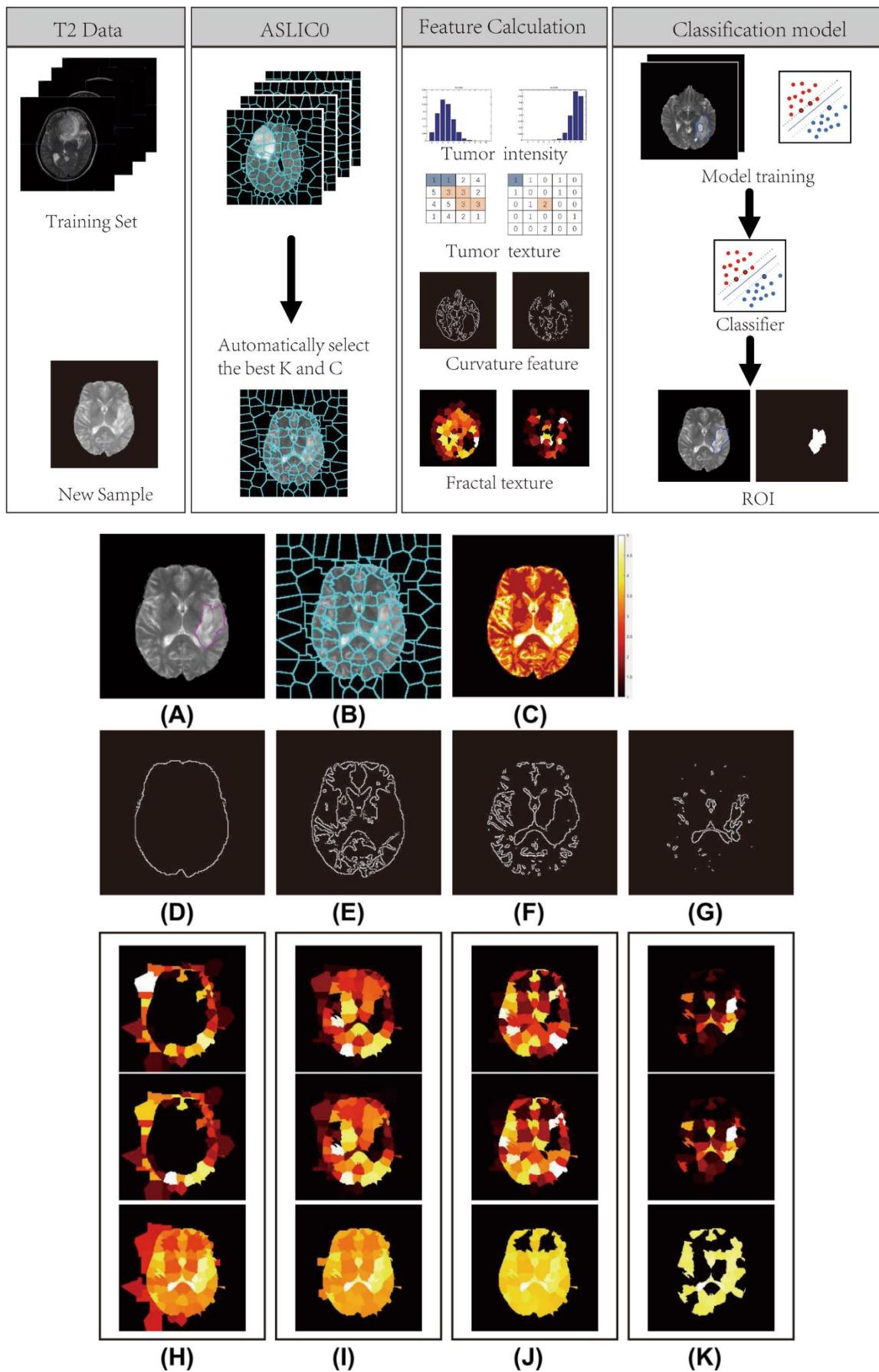
FRACTALS: A fractal is a never-ending pattern. Fractals are infinitely complex patterns that are self-similar across different scales. They are created by repeating a simple process over and over in an ongoing feedback loop. Driven by recursion, fractals are images of dynamic systems – the pictures of Chaos. Geometrically, they exist in between the familiar dimensions. Fractal patterns are extremely familiar, since nature is full of fractals. For instance: trees, rivers, coastlines, mountains, clouds, seashells, hurricanes, etc. Abstract fractals – such as the Mandelbrot Set – can be generated by a computer calculating a simple equation over and over.

USE OF FRACTALS IN DEEP LEARNING:

Biological central nervous systems with their massive parallel structures and recurrent projections show fractal characteristics in structural and functional parameters (Babloyantz and Ltheen, 1994). Julia sets and the Mandelbrot set are the well-known classical fractals with all their harmony, deterministic chaos and beauty, generated by iterated non-linear functions. The according algorithms may be transposed, based on their geometrical interpretation, directly into the massive parallel structure of neural networks working on recurrent projections.

Paper: Automatic glioma segmentation based on adaptive superpixel

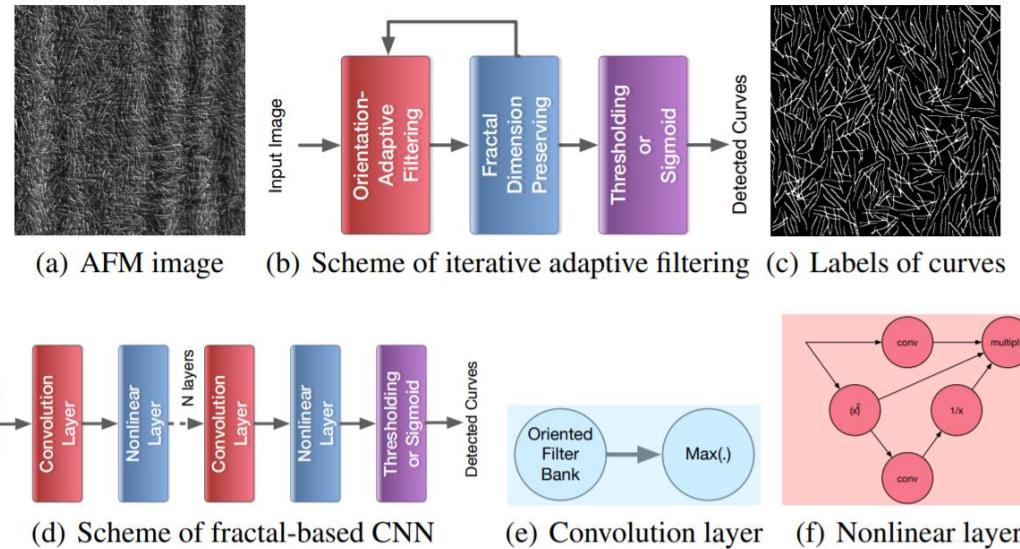
Proposed an adaptive superpixel generation algorithm based on simple linear iterative clustering version with 0 parameter (ASLICO). The algorithm can acquire a superpixel image with fewer superpixels and better fit the boundary of region of interest (ROI) by automatically selecting the optimal number of superpixels. Composed a training set by calculating the statistical, texture, curvature and fractal features for each superpixel. Support Vector Machine (SVM) is used to train classification model based on the features of the second step.



Paper: A Fractal-based CNN for Detecting Complicated Curves in AFM Images

A geometrical interpretation of CNN based on local fractal analysis of image is explored and propose a fractal-based CNN (FCNN) for complicated curve detection is proposed. Fractal dimension contains important structural information of image. For detecting structures, e.g., curves in images, robustly, one should preserve fractal dimension from changing after feature extraction. A significant property of fractal dimension is its invariance to bi-Lipschitz transform.

A Fractal-based CNN for Detecting Complicated Curves in AFM Images



Paper: 3D FractalNet: Dense Volumetric Segmentation for Cardiovascular MRI Volumes

Proposed a novel deeply-supervised 3D FractalNet for automated whole heart and great vessel segmentation from cardiovascular magnetic resonance (CMR) images. By adopting 3D fully convolutional neural networks, the network can perform accurate, efficient, volume-to-volume prediction. Under a recursive fractal scheme, the network can fuse interacting subpaths of different convolution lengths and thus utilize multi-scale features to enhance its discrimination capacity.

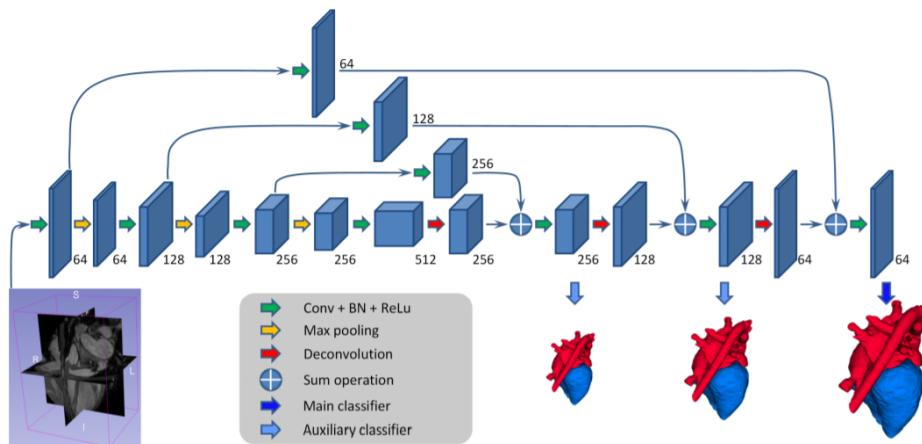
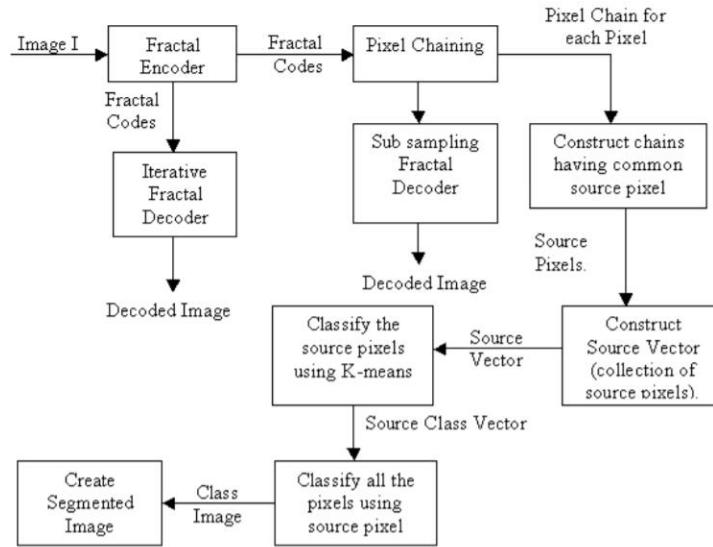


Fig. 1: Illustration of our proposed deeply-supervised 3D FractalNet architecture. Numbers represent the number of feature volumes in each layer.

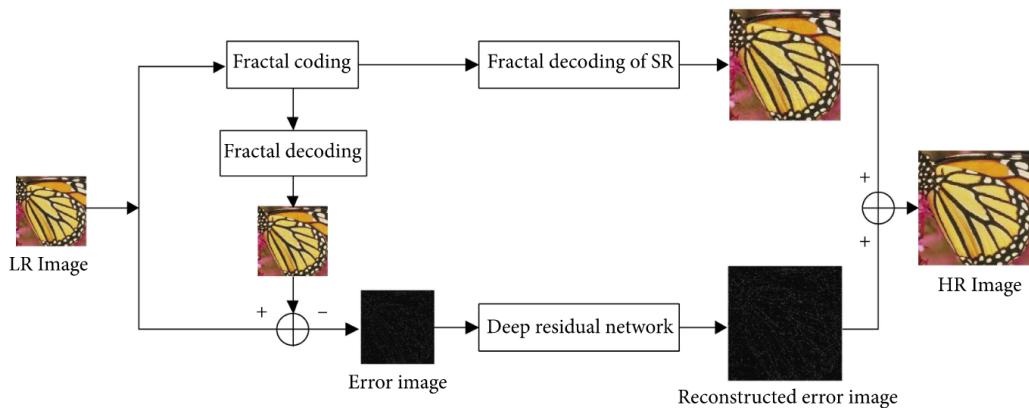
Paper: Fractal Based Image Segmentation

Proposed an algorithm for image segmentation, using the fractal codes. The basic idea is to use fractal codes for the image segmentation. This method uses compressed codes instead of the gray levels of the image. It is cost effective since it operates directly on the fractal codes of the image. Hence, the proposed scheme can be directly used on the images accessed from the image database where images are kept in fractal-compressed format.



Paper: Image Super Resolution Using Fractal Coding and Residual Network

Proposed a super-resolution method based on fractal image interpolation and depth residual network, which can effectively recover texture structure details, improve edge blur and distortion problems, and have high SR reconstruction performance. First, the image is fractal coded to establish the similarity between the range block and the domain block. Then, used the similarity relationship to SR reconstruction of the image with SR fractal decoding. In the fractal decoding process, since the fractal image encoding is lossy compression, there is an error between the decoded image and the original image. To estimate the error compensation term more accurately, the residual network is used to train the mapping relationship between high- and low-resolution error images.



PAPER	DATASET	METRIC	METHOD	COMMENTS
Automatic glioma segmentation based on adaptive superpixel	BraTS2017 Dataset	Dice= 0.8492 Hausdorff distance= 3.4697 pixels, sensitivity=81.47, and specificity= 99.64% against ground truth	Clustering method used to segment superpixels and classifiers trained on the basis of features calculated from each superpixel to classify tumor regions. Otsu algorithm is used to obtain binary images with different thresholds, while the fractal edge information of each binary image is obtained. The fractal features include the area, average brightness, and fractal distance, which is obtained using the box algorithm.	The SVM prediction model is trained by calculating the statistical, texture, curvature, and fractal features of each superpixel. The superpixels are then classified into tumor or non-tumor types.
A Fractal-based CNN for Detecting Complicated Curves in AFM Images	AFM Dataset	ODS=0.740 OIS=0.777 AP=0.709 Time(sec)=1.054	Taking an image as a union of local fractals with intensity-based measurement, combined orientation-adaptive filters with fractal dimension preserving processes, and proposed an iterative framework to extract local features for curves.	The first attempt to combining fractalbased image model with neural networks. Enctheaging results of curve detection are achieved, especially for texture-like noisy images. The approach can be utilized as a preprocessing step or feature extraction method for images.
3D FractalNet: Dense Volumetric Segmentation for Cardiovascular MRI Volumes	HVS MR 2016 Challenge dataset	Dice coefficient (Dice)=0.852 Hausdorff Distance of Boundaries (Hdb[mm])=4.279 Average Distance of Boundaries (Adb[mm])=0.675	By recursively applying a single expansion rule, a network in a novel self-similar fractal scheme is constructed and is promoted in combining hierarchical clues for accurate segmentation. Deep supervision mechanism is employed to alleviate the vanishing gradient problem and improve the training efficiency of the network on small medical image dataset.	The proposed 3D fractal network takes advantage of fully convolutional architecture to perform efficient, precise, volume-to-volume prediction.
Fractal Based Image Segmentation	ORL Database	K in K-means Clustering	Extracted the pixel chains using the convergence of the Partition Iterated Function System (PIFS). The pixel chains so extracted were verified by decoding the images using Sub-sampling Fractal Decoder and the results are compared with those obtained using the Iterative Fractal Decoder. Extracted the	The advantage of this approach is that it operates directly in the fractal domain. The quality of segmentation achieved is enctheaging. Since the performance of algorithm depends on the PIFS or fractal code, it is natural that the quality of the segmented image is

			dynamics of the convergence of PIFS and used it for the segmentation of the image.	subject to correctness of these fractal codes.
Image Super Resolution Using Fractal Coding and Residual Network	800 training images, 100 validation images, and 100 test images of DIV2K Dataset	Peak signal-to-noise ratio (PSNR)= 28.84 Structural similarity (SSIM)=0.7878	Image SR reconstruction is used to reconstruct a poor-quality low-resolution (LR) image into a high-resolution (HR) image close to the real image. An edge-directed interpolation method is used to estimate unknown pixels in HR images from pixels in their known domain. Input the encoded and decoded error image into the depth residual network for estimation and used this as a compensation term to correct the interpolation image to further improve the reconstruction accuracy.	The fractal-based SR method utilizes the locality of the similarity and transitivity of a single image to search for similar image blocks. Using the analogy method, the information of the LR image block is merged to reconstruct the HR image blocks. The method has some improvement in the recovery of the edges, but the fractal dimension does not accurately represent the texture details.

DATASETS:

ATLAS Dataset –

ATLAS (Anatomical Tracings of Lesions After Stroke), an open-source dataset of 229 T1-weighted MRI scans (n=220) with manually segmented lesions and metadata. This large, diverse dataset can be used to train and test lesion segmentation algorithms and provides a standardized dataset for comparing the performance of different segmentation methods.

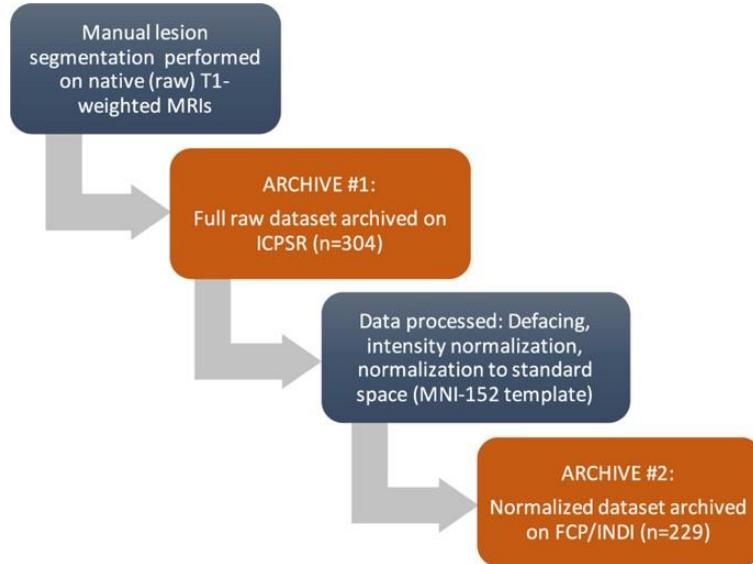
The dataset includes:

- 229 T1-weighted MRI scans (n=220) with lesion segmentation
- MNI152 standard-space T1-weighted average structural template image
- A .csv file containing lesion metadata

304 MRI images from 11 cohorts worldwide were collected from research groups in the ENIGMA Stroke Recovery Working Group consortium. Images consisted of T1-weighted anatomical MRIs of individuals after stroke. These images were collected primarily for research purposes and are not representative of the overall general stroke population (e.g., only including individuals who opt in to participate in a research study, and excluding individuals with stroke who cannot undergo MRI safely).

For each MRI, brain lesions were identified and masks were manually drawn on each individual brain in native space using MRIcron24, an open-source tool for brain imaging visualization and defining volumes of interest. At least one lesion mask was identified for each individual MRI. If additional, separate (non-contiguous) lesions were identified, they were traced as separate masks.

Finally, a separate tracer performed quality control on each lesion mask. This included assessing the accuracy of the lesion segmentations, revising the lesion mask if needed, and categorizing the lesions to generate additional data such as the number of lesions in left and right hemispheres, and in cortical and subcortical regions. This dataset is provided in native subject space and archived (n=304). A subset of this dataset was also defaced, intensity normalized, and provided in standard space (normalized to the MNI-152 template, n=229; for an overview of the dataset and archives).



- **Data Characteristics –**

All T1-weighted MRI data were collected on 3T MRI scanners at a resolution of 1 mm³ (isotropic), with the exception of data from cohorts 1 and 2 which were collected on a 1.5T scanner with a resolution of 0.9 mm×0.9 mm×3.0 mm (excluded from the normalized dataset). Scanner information (scanner strength, brand) and image resolution are included in the ATLAS meta-data, and sample image header information for a subject from each of the cohorts can be found in Supplementary Information.

Characteristics of the ATLAS dataset include an average lesion volume across all cohorts of $2.128 \pm 3.898 \times 10^4$ mm³, with a minimum lesion size of 10 mm³ and a maximum lesion size of 2.838×105 mm³. Information regarding the distribution of lesions in the ATLAS dataset (e.g., single versus multiple lesions per individual, percent of lesions that are left versus right hemisphere, or subcortical versus cortical) can be found in Tables 1 and 2. Overall, slightly more than half of the subjects had only one lesion (58%) while the rest had multiple lesions (42.1%). Lesions were roughly equally distributed between left and right hemispheres (48.4% left hemisphere, 43.8% right hemisphere, 7.7% other location such as brainstem or cerebellum). In this dataset, there were more subcortical lesions than cortical lesions (70.7% subcortical, 21.5% cortical, 7.7% other).

- **Metadata –**

For each lesion, we also provided metadata on the lesion properties to give the user additional qualitative information, beyond the binary lesion mask. This information can be used to quickly sort the dataset based on specific lesion characteristics (e.g., only left hemisphere lesions, or only subcortical lesions). It can also provide additional insight into the types of lesions that succeed or fail for a given lesion segmentation algorithm. The lesion properties were manually reported for each individual lesion mask. These include the number of lesions identified and traced, and the location of each lesion (i.e., right/left, subcortical, cortical, or other).

ISLES Dataset –

Uncompressed Neuroimaging Informatics Technology Initiative (NIfTI) format: *.nii.

SISS

number of cases:	28 training 36 testing
number of centers:	1 (train), 2 (test)
number of expert segmentation for each case:	1 (train), 2 (test)
MRI sequences:	<u>FLAIR</u> , <u>T2w TSE</u> , <u>T1w TFE/TSE</u> , <u>DWI</u>

SPES

number of cases:	30 training 20 testing
number of centers:	1
number of expert segmentation for each case:	1
MRI sequences:	T1c, T2, DWI, CBF, CBV, TTP, Tmax

Training data set consists of 63 patients. Some patient cases have two slabs to cover the stroke lesion. These are non-, or partially-overlapping brain regions. Slabs per patient are indicated with letters "A" and "B" for first and second slab, respectively. The mapping between case number and training name is also provided at SMIR (e.g. Train_40_A = case 64; Train_40_B = case 65). Developed techniques will be evaluated by means of a testing set including 40 stroke cases.

Imaging data from acute stroke patients in two centers who presented within 8 hrs of stroke onset and underwent an MRI DWI within 3 hrs after CTP were included.

CHAPTER – 4

Problem Solving Methods

Problem Statement:

The **segmentation** of **brain lesions** from a **brain** magnetic resonance (MR) image is of great significance for the clinical diagnosis and follow-up treatment. One of the most important tasks in clinical practice is to analyse multisequence MR images and segment the brain lesions to calculate the shape and volume of the lesion regions. However, having radiologists segment multiple three-dimensional (3D) images manually is time-consuming, and the segmentation results are generally not repeatable. Therefore, automatic or semiautomatic segmentation methods for brain lesions are important. Exploring the scope of various methods in Brain Stroke Lesion Segmentation containing greyscale images of Brain MRI (ATLAS Dataset and ISLES Dataset).

Objective:

Implement a deep learning method for effective lesion segmentation of Brain MRI to improve state-of-the-art models.

State of the Art Model:

- DALS: Deep Active Lesion Segmentation

DALS performs well for different image characteristics, including low contrast lesions, heterogeneous lesions, and noise. DALS outperformed the manually-initialized ACM and its backbone CNN in all metrics across all evaluations on every organ. It well assists in capturing local and global context for highly accurate lesion localization.

The framework combines the capabilities of the CNN and the level-set ACM to yield a robust, fully automatic medical image segmentation method that produces more accurate and

detailed boundaries compared to competing state-of-the-art methods. The DALS framework includes an encoder-decoder that feeds a level-set ACM with per-pixel parameter functions.

Method 1: CNN Based Implementation of Fractal Dimension Invariant Filtering

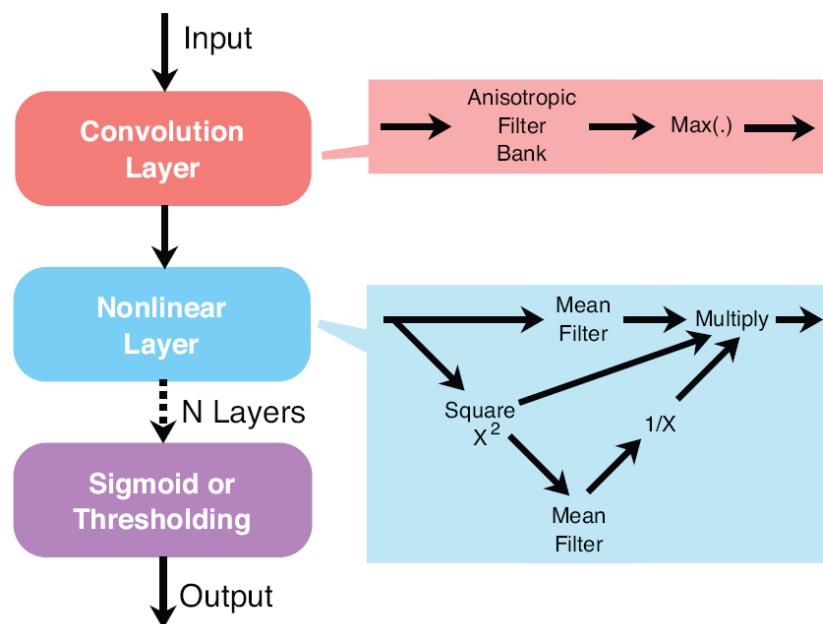
The fractal dimension is invariant to bi-Lipschitz transformation. This property means that the fractal dimension is robust to geometrical deformation (e.g., ridge and non-ridge transformation) of image. Hence, the fractal dimension reflects intrinsic structural information of image, which can be treated as a representative feature of image.

Bi-Lipschitz Invariance

For a fractal F with fractal dimension D , its bi-Lipschitz transformation is still a fractal, whose fractal dimension $D_g = D$.

Anisotropic Filtering

To suppress fractal dimension change, the expectation of the filter shall be as close as to impulse function. Anisotropic filters have been one natural choice for this purpose.



Fractal Based Image Model

A typical fractal is generated via transforming a geometry G to N analogues with scaling factor s and then applying the transformation infinitely on each analogue. The union of the analogues is a fractal, denoted as F. The fractal F is a “Mathematical monster” that is unmeasurable in the measure space of G. The analysis of fractal is mainly based on the Hausdorff measure, which gives rise to the concept of fractal dimension. The fractal dimension is involved by a power law of measurements across multiple scales, i.e., the quantities $N \propto 1/s^D$. Here D is called fractal dimension, which is larger than the topological dimension of F.

-
- 1: **Input:** $f(\mathbf{X})$, the number of scales R .
 - 2: **Output:** Fractal dimension $D(\mathbf{X})$.
 - 3: For $r \in \{1, \dots, R\}$, perform a convolution of $f(\mathbf{X})$ with G_r to get $\{\mu(B_r(\mathbf{x}))\}_{\mathbf{x} \in \mathbf{X}}$.
 - 4: $\min_{D,L} \sum_r |\log \mu(B_r(\mathbf{x})) - D \log 2r - L|^2, \mathbf{x} \in \mathbf{X}$.
 - 5: $D(\mathbf{X}) = \{D(\mathbf{x})\}_{\mathbf{x} \in \mathbf{X}}$.
-

To pursue the fractal dimension preservation philosophy in face of the reality that filtering will inevitably change fractal dimension, we aim to suppress the expected change between original fractal dimension and filtered one. Denote the proposed filter as F, the measurement and the fractal dimension of filtering result as μ_F and D_F , respectively. We assume that the filter F is a random variable yielding to a probabilistic distribution.

CNN Architecture

In the convolution layer, the anisotropic filtering can be approximately implemented via a filter bank. At each pixel x , the process can be written. The normalization term is implemented via a convolution, where M is a mean filter, which sums the intensities in the neighborhood $B(x)$ for each x . Different from neuroscience, we explain the rectified linear unit (ReLU, $\max\{\cdot, 0\}$) based on fractal analysis. The ReLU ensures the filtering result to be a valid measurement (as the measurement used in the box-counting method.)

Add a sigmoid layer to the end of the CNN and train the model via traditional backpropagation algorithm, or apply a thresholding layer for the final output. In contrast to many CNN models with a disadvantage of their ravenous appetite for labelled training, we believe the

adaptability for unlabelled data of the method is perhaps due to the fact that we instantiate the tailored CNN from the fractal-based geometry perspective.

-
- 1: **Input:** Image $f(\mathbf{X})$, filter bank F_Θ , layer number N .
 - 2: **Output:** Binary map $b(\mathbf{X})$ corresponding to curves.
 - 3: For $n = 1, \dots, N$, obtain $f_{T \circ F}(\mathbf{X})$ from $f(\mathbf{X})$ via (5,6), and set $f(\mathbf{X}) = f_{T \circ F}(\mathbf{X})$.
 - 4: **Unsupervised:** $b(\mathbf{X}) = \text{binary}(f(\mathbf{X}))$.
 - 5: **Supervised:** $b(\mathbf{X}) = \text{sigmoid}(\boldsymbol{\beta}^\top \mathbf{P})$. $\boldsymbol{\beta}$ is learned parameters, \mathbf{P} are patch matrix of $f(\mathbf{X})$.
-

Expected Results

Taking an image as a union of local fractals, presented a model involving anisotropic filtering with fractal dimension preservation. Compared with the state-of-art learning-based detector, an important advantage of the proposed method is that it is able to detect unlabeled curves. The ground truth of curves is manually labeled.

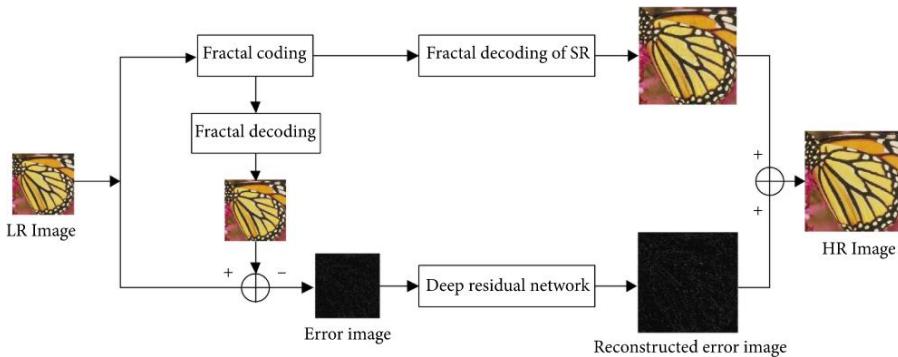
Method 2: Image Super Resolution Using Fractal Coding and Residual Network

Image SR reconstruction is used to reconstruct a poor-quality low-resolution (LR) image into a high-resolution (HR) image close to the real image. SR is widely used in various fields, such as video surveillance, remote sensing, and medical imaging. Due to the uncertainty of the image degradation model and the nonuniqueness of the reconstruction constraints, SR is essentially an ill-posed problem. There are three main approaches to solving the SR problem: interpolation-based, reconstruction-based, and learning-based approaches.

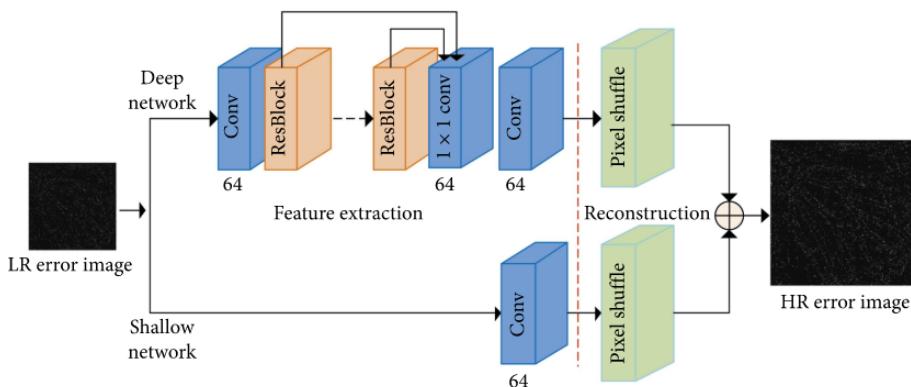
- The basic idea of the learning-based method is to obtain a mapping of LR images to HR images by training samples, thereby predicting HR images.
- The fractal-based SR method utilizes the locality of the similarity and transitivity of a single image to search for similar image blocks. Using the analogy method, the information of the LR image block is merged to reconstruct the HR image blocks. The similarity can be considered as a contractive fractal transformation operator that performs shrinking and gray-level modifying operations on the image.
- 1) Fractal Image Coding - The process of fractal image compression coding is based on the collage theorem. A set of compression maps is obtained by a given image so that the attractor of the

iterated function system is approximated to a given image, and then the corresponding parameters are recorded.

- 2) Fractal Image Decoding - Fractal decoding is a process of generating the original image's attractor by using the fractal information of the original image and the initial image. In the fractal decoding stage, we can select any initial image as the initial image, which is generally a blank image. Then, we use the iterative function for decoding until it converges to approximate the original image.
- To reduce the error, a quadtree fractal coding strategy with variable block size can be adopted. The basic idea is to first use a larger size range block to search for each optimal domain block. If the corresponding tile error is greater than a given threshold, the range block is decomposed into the smaller blocks which have a transformed relationship with the corresponding domain blocks. Each block is then researched recursively for optimal domain blocks and encoded until all image blocks have been encoded.



- The proposed method is based on fractal technology to achieve SR image reconstruction. To better estimate the estimation error compensation term more accurately, the network architecture with depth residual is used to estimate. The deep network includes a feature extraction phase and a reconstruction phase. The traditional feature extraction methods mostly use first-order and two-step methods to filter the input image, while the deep learning method does not need to manually design the filter but automatically learns from the training data. In the feature extraction stage, take the LR error image as input and forward it through a neural network as a series of feature maps. The feature extraction network consists of a convolutional layer and a plurality of residual blocks. First, we use a convolutional layer to learn shallow features and join the rectified linear unit (ReLU) for nonlinear mapping.



- **Conclusion** - First, the image is fractal coded to establish the similarity between the range block and the domain block. Then, we use the similarity relationship to SR reconstruction of the image with SR fractal decoding. In the fractal decoding process, since the fractal image encoding is lossy compression, there is an error between the decoded image and the original image. To estimate the error compensation term more accurately, the residual network is used to train the mapping relationship between high- and low-resolution error images.

Method – 3: Fractal image compression hybrid algorithm based on convolutional neural network and gene expression programming

Image segmentation is a key step in fractal image compression; however, all existing image segmentation methods have the problems of insufficient automation and intelligence and low image recovery accuracy. To this end, we classify the fractal image through a five-layer CNN in CPU/GPU parallel system environment, and use the adaptive quadtree to automatically segment the classification results. The image classification is processed in parallel by CNN, and it creates multiple threads through the multi-core CPU, and each thread is responsible for transmitting the image convolution processing data to the GPU, and performing the scheduling and data receiving of GPU array processor, respectively. The original image is classified in parallel by CNN on the GPU and its processing includes five convolution layers, five maximum pooling layers, one fully connected layer and one upsampling layer. With the automatic segmentation of image range block and domain block, the classified images are divided into the non-overlapping range block and the overlapping domain block by CPU multi-threading through the adaptive quadtree, thus achieving the parallel segmentation of fractal images.

Architecture:

According to the design idea of ‘Image classification principle of CNNs’ section, through the fractal image parallel segmentation of deep CNN and the fractal image compression encoding of gene expression programming, the fractal image compression hybrid algorithm based on deep CNN and gene expression programming is obtained, and its steps are as follows:

Input: Read the original image data; initial parameters of convolution training; initial parameters of gene expression programming such as population size, gene head and tail length, number of genes, maximum iteration number, termination iteration fitness value, mutation rate, interpolation rate, and recombination rate.

Output: Output the optimal coded IFS.

Begin:

Step 1: Create multiple (set $k = 2, 4, 8, 16\dots$) threads from the multi-core CPU, one of which is the main thread and the other is the slave thread.

Step 2: Read the original image data, and send the image data to the corresponding array computing core of the GPU by multiple threads.

Step 3: Call the CNN fractal image classification and segmentation algorithm. The image classification is processed in parallel by CNN, which creates multiple threads through the multi-core CPU, and the multiple threads are responsible for transmitting the image convolution processing data to the GPU, and respectively performing the scheduling and data receiving on GPU array processor. The CNN is used to classify the original image in parallel in GPU, and its processing includes five convolution layers, five maximum pooling layers, one fully connected layer and one upsampling layer, and finally the Soft-max layer obtains the classification result, wherein there is one ReLU activation function layer behind each convolution layer, and the pooling layer adopts the maximum pooling.

Step 4: The main thread divides the classified image B of the size $2N \times 2N$ into fthe equal non-overlapping sub-blocks $2N-2 \times 2N-2$, and one sub-block is reserved by itself, and the other three sub-blocks are distributed to other slave threads for processing; the main thread again divides the reserved sub-block $2N-2 \times 2N-2$ into fthe grand-subblocks $2N-4 \times 2N-4$ of equal size, one part is reserved by itself, and the other three grand-subblocks are distributed again to other slave threads. Similarly, the other three sub-blocks are also divided into fthe equal parts by their respective slave threads, and one grand-subblock is reserved by itself, and the other three parts are distributed to other slave threads until each thread can divide

the sub-blocks $R_i (i = 1, 2, \dots, 2N-R \times 2N-R)$ of size $2R \times 2R$, then these sub-blocks are all sent back to the main thread for unified storage, and a range block pool is established.

Step 5: The main thread divides the classified image B of the size $2N \times 2N$ into the equal overlapping sub-blocks $2N-2 \times 2N-2$, and one sub-block is reserved by itself, and the other three sub-blocks are distributed to other slave threads for processing; the main thread again divides the reserved sub-block $2N-2 \times 2N-2$ into the grand-subblocks $2N-4 \times 2N-4$ of equal size, one part is reserved by itself, and the other three grand-subblocks are distributed again to other slave threads. Similarly, the other three sub-blocks are also divided into the equal parts by their respective slave threads, and one grand-subblock is reserved by itself, and the other three parts are distributed to other slave threads until each thread can divide the parent-blocks $D_i (i=1,2,\dots,2N-D \times 2N-D)$ of size $2D \times 2D$, then, these sub-blocks are all sent back to the main thread for unified storage, and a domain block pool is established.

Step 6: Initialize the population. In the main thread, input the initial parameters such as population size, gene head and tail length, number of genes, maximum iteration number (maxg), termination iteration fitness value (minf), mutation rate, interpolation rate, recombination rate, and distribute them to each slave thread.

Step 7: Adopt the dynamic task allocation–work pool parallel search and encoding method. The main thread is responsible for the task management and allocation of range block classification pool. First, the complete i th ($i = 1, 2, \dots, m$) class range sub-block set is sent to the k th ($k = 1, 2, \dots, p$) slave thread according to the classification order. If the distribution cannot be fully completed in one time, the remaining range sub-block sets ($p + 1, \dots, m$) are allocated as required by slave thread.

Step 8: After the k th ($k = 1, 2, \dots, p$) slave thread receives the first range sub-block set, the sub-blocks are grouped together and the sub-block diagrams $R_{ij} (i = 1, 2, \dots, m; j = 1, 2, \dots, g)$ are taken out one by one for encoding with the corresponding domain sub-block set $D_{ij} (i = 1, 2, \dots, n; j = 1, 2, \dots, q)$ in domain block classification pool.

Step 9: In the encoding process, each thread calculates the fitness value according to formulas (11) to (14).

Step 10: The k th ($k = 1, 2, \dots, p$) slave thread sends nine basic operators of gene expression programming used in the programming process (including selection, mutation, reversal, interpolation, root interpolation, gene transformation, single-point recombination, two-point recombination, and gene recombination) and the calculation of individual fitness values to the $k*10$ kernels of GPU for parallel computations:

do

{

U_{k1} nuclear calculation: selection operation;

U_{k2} nuclear calculation: mutation operation;

U_{k3} nuclear calculation: reversal operation;

U_{k4} nuclear calculation: interpolation operation;

U_{k5} nuclear calculation: root interpolation operation;

U_{k6} nuclear calculation: gene transformation;

U_{k7} nuclear calculation: single-point recombination;

U_{k8} nuclear calculation: two-point recombination;

U_{k9} nuclear calculation: gene recombination;

U_{k10} nuclear calculation: calculation of individual fitness values;

} while (fitness \leq minf or gen \leq maxg)

Step 11: k ($k = 1, 2, \dots, p$) slave threads will complete the encoding of a range block R_{ij} , and obtain the parameters of classification sub-block sets such as the transform parameters, the domain block D_i and the compression transform ω_{ij} corresponding to the range block R_{ij} , which are then sent back to the main thread.

The main threads form them into the complete and optimal coded IFS according to the original image classification order.

Step 12: The main thread outputs the optimal coded IFS.

PUGH Matrix

Baseline Solution: Lesion Segmentation using Deep Learning on ATLAS

Criteria	Weight	Baseline	Alternatives		
		Canny Edge Detection	Method 1	Method 2	Method 3
Intensity Variation Sensitivity		0	+	+	+
Edge Distinction		0	-	-	+
Pattern Sensitivity		0	+	+	+
Texture Detection		0	+	+	-
Homogeneity in Regions		0	+	+	0
Contrast		0	+	0	0
Architecture Complexity		0	0	+	-

Inferences:

- Fractal Dimension invariant filtering based CNN method is best suitable for the given objective using fractals.
- Image compression using hybrid filtering can be modified further for higher training accuracy.

CHAPTER – 5

Model Exploration: FasterSeg Network

Network Implementation – FasterSeg: Searching for Faster Real-time Semantic Segmentation

FasterSeg is an automatically designed semantic segmentation network with not only state-of-the-art performance but also faster speed than current methods. Utilizing neural architecture search (NAS), FasterSeg is discovered from a novel and broader search space integrating multi-resolution branches that has been recently found to be vital in manually designed segmentation models. To better calibrate the balance between the goals of high accuracy and low latency, we propose a decoupled and fine-grained latency regularization that effectively overcomes the observed phenomena that the searched networks are prone to "collapsing" to low-latency yet poor-accuracy models. Moreover, we seamlessly extend FasterSeg to a new collaborative search (co-searching) framework, simultaneously searching for a teacher and a student network in the same single run. The teacher-student distillation further boosts the student model's accuracy. Experiments on popular segmentation benchmarks demonstrate the competency of FasterSeg. For example, FasterSeg can run over 30% faster than the closest manually designed competitor on Cityscapes, while maintaining comparable accuracy.

The key features of the proposed network are:

- A novel NAS search space tailored for real-time segmentation, where multi-resolution branches can be flexibly searched and aggregated.
- A novel decoupled and fine-grained latency regularization that successfully alleviates the "architecture collapse" problem in the latency-constrained search.

- A novel extension to teacher-student co-searching for the first time, where the teacher network is distilled to the student for further accuracy boost of the latter.
- Extensive experiments demonstrating that FasterSeg achieves extremely fast speed (over 30% faster than the closest manually designed competitor on CityScapes) and maintains competitive accuracy.

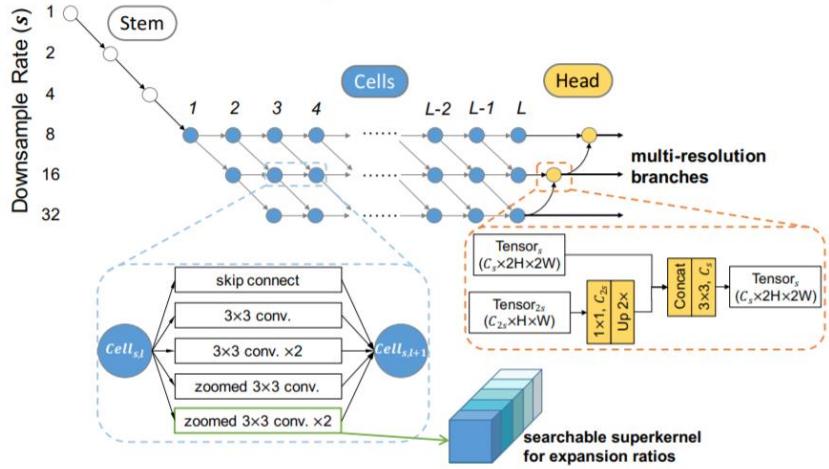


Figure 1: The multi-resolution branching search space for FasterSeg, where we aim to optimize multiple branches with different output resolutions. These outputs are progressively aggregated together in the head module. Each cell is individually searchable and may have two inputs and two outputs, both of different downsampling rates (s). Inside each cell, we enable searching for expansion ratios within a single superkernel.

Neural Architecture Search:

Neural architecture search (NAS) is a technique for automating the design of artificial neural networks, a widely used model in the field of machine learning. NAS has been used to design networks that are on par or outperform hand-designed architectures. Methods for NAS can be categorized according to the search space, search strategy and performance estimation strategy used:

- The search space defines the type(s) of ANN that can be designed and optimized.
- The search strategy defines the approach used to explore the search space.
- The performance estimation strategy evaluates the performance of a possible ANN from its design (without constructing and training it).

The search space consists of the following operators:

- skip connection • 3×3 conv. • 3×3 conv. $\times 2$
- “zoomed conv.”: bilinear downsampling + 3×3 conv. + bilinear upsampling
- “zoomed conv. $\times 2$ ”: bilinear downsampling + 3×3 conv. $\times 2$ + bilinear upsampling

Architecture Collapse Problem - Low latency is desirable yet challenging to optimize. Previous works observed that during the search procedure, the supernet or search policy often fall into bad “local minimums” where the generated architectures are of extremely low latency but with poor accuracy, especially in the early stage of exploration. In addition, the searched networked tend to use more skip connections instead of choosing low expansion ratios. This problem is termed as “architecture collapse”. The potential reason is that, finding architectures with extremely low latency (e.g. trivially selecting the most light-weight operators) is significantly easier than discovering meaningful compact architectures of high accuracy.

Latency Optimization with Finer Granularity - Quantitatively compared the influence of operators (O), downsample rate (s), and expansion ratios (χ) towards the supernet latency, by adjusting one of the three aspects and fixing the other two. Taking O as the example, we first uniformly initialize β and γ , and calculate $\Delta\text{Latency}(O)$ as the gap between the supernet which dominantly takes the slowest operators and the one adopts the fastest. Similar calculations were performed for s and χ .

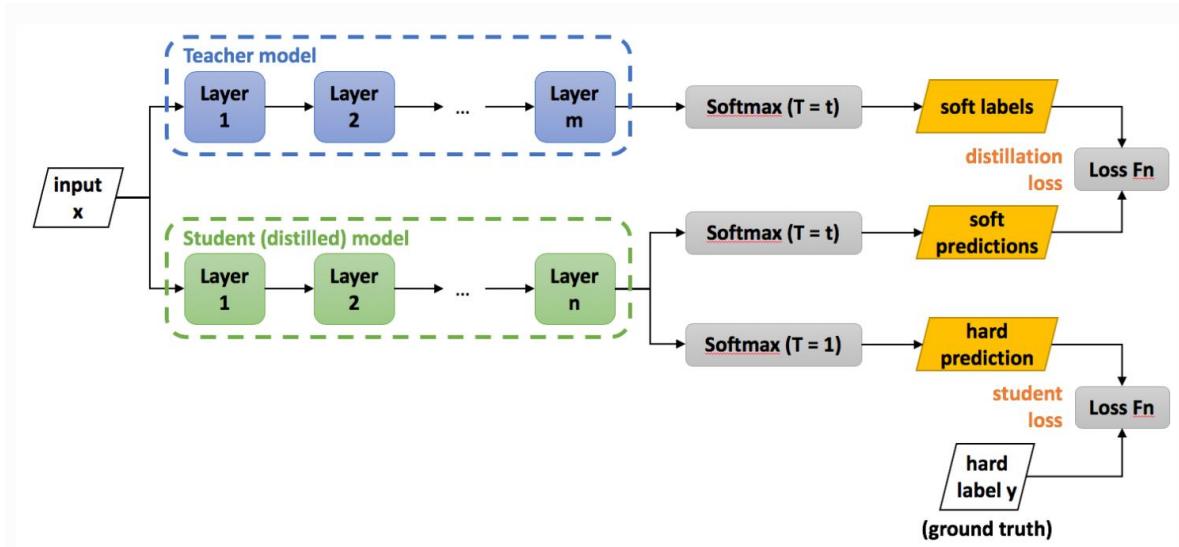
Based on this observation, a regularized latency optimization is proposed leveraging different granularities of the search space. Decoupled the calculation of supernet’s latency into three granularities of the search space (O, s, χ), and regularize each aspect with a different factor:

$$\text{Latency}(O, s, \chi) = w_1 \text{Latency}(O|s, \chi) + w_2 \text{Latency}(s|O, \chi) + w_3 \text{Latency}(\chi|O, s)$$

Knowledge Distillation:

Knowledge distillation is model compression method in which a small model is trained to mimic a pre-trained, larger model (or ensemble of models). This training setting is sometimes referred to as "teacher-student", where the large model is the teacher and the small model is the student (we'll be using these terms interchangeably).

In distillation, knowledge is transferred from the teacher model to the student by minimizing a loss function in which the target is the distribution of class probabilities predicted by the teacher model. That is - the output of a softmax function on the teacher model's logits.



Dataset:

The Cityscapes Dataset focuses on semantic understanding of urban street scenes.

Type of annotations

- Semantic
- Instance-wise
- Dense pixel annotations

Complexity

- 30 classes

Diversity

- 50 cities
- Several months (spring, summer, fall)

- Daytime
- Good/medium weather conditions
- Manually selected frames
 - Large number of dynamic objects
 - Varying scene layout
 - Varying background

Volume

- 5000 annotated images with fine annotations

Metadata

- Preceding and trailing video frames. Each annotated image is the 20th image from a 30 frame video snippets (1.8s)
- Corresponding right stereo views
- GPS coordinates
- Ego-motion data from vehicle odometry
- Outside temperature from vehicle sensor

Benchmark suite and evaluation server

- Pixel-level semantic labeling
- Instance-level semantic labeling

Group	Classes
flat	road · sidewalk · parking+ · rail track+
human	person* · rider*
vehicle	car* · truck* · bus* · on rails* · motorcycle* · bicycle* · caravan*+ · trailer*+
construction	building · wall · fence · guard rail+ · bridge+ · tunnel+
object	pole · pole group+ · traffic sign · traffic light
nature	vegetation · terrain
sky	sky
void	ground+ · dynamic+ · static+

* Single instance annotations are available. However, if the boundary between such instances cannot be clearly seen, the whole

crowd/group is labeled together and annotated as group, e.g. car group.

+ This label is not included in any evaluation and treated as void

Network Evaluation - FasterSeg on Cityscapes is evaluated on validation and test sets. Used the original image resolution of 1024×2048 to measure both mIoU and speed inference. Even under the maximum image resolution, the FasterSeg network gives superior performance. This high FPS is over 1.3× faster than human-designed networks. Meanwhile, the FasterSeg still maintains competitive accuracy, which is 73.1% on the validation set and 71.5% on the test set. This accuracy is achieved with only Cityscapes fine-annotated images, without using any extra data.

Method	mIoU (%)		FPS	Resolution
	val	test		
ENet (Paszke et al., 2016)	-	58.3	76.9	512×1024
ICNet (Zhao et al., 2018)	67.7	69.5	37.7	1024×2048
BiSeNet (Yu et al., 2018a)	69.0	68.4	105.8	768×1536
CAS (Zhang et al., 2019)	71.6	70.5	108.0	768×1536
Fast-SCNN (Poudel et al., 2019)	68.6	68.0	123.5	1024×2048
DF1-Seg-d8 (Li et al., 2019)	72.4	71.4	136.9	1024×2048
FasterSeg (ours)	73.1	71.5	163.9	1024×2048

mIoU and inference FPS on Cityscapes validation and test sets.

A novel multi-resolution NAS framework, leveraging successful design patterns in handcrafted networks for real-time segmentation. The NAS framework can automatically discover FasterSeg, which achieved both extremely fast inference speed and competitive accuracy. The search space is intrinsically of low-latency and is much larger and challenging due to flexible searchable expansion ratios. More importantly, it successfully addressed the “architecture collapse” problem, by proposing the novel regularized latency optimization of fine-granularity. Also demonstrated that by seamlessly extending to teacher-student co-searching, the NAS framework can boost the student’s accuracy via effective distillation.

Conclusion:

Explored and gained knowledge about Neural Architecture Search and latency regularization techniques. Trained a teacher-student network using softmax labels on MNIST data to gain concrete knowledge about Knowledge Distillation.

Paper: <https://openreview.net/pdf?id=BJgqQ6NYvB>

Code: <https://github.com/TAMU-VITA/FasterSeg>

References

- [1] Yongjin Zhou et al., D-UNet: a dimension-fusion U shape network for chronic stroke lesion segmentation ([arXiv:1908.05104](https://arxiv.org/abs/1908.05104) [eess.IV]), 2019 Aug
- [2] SZUHvern github source code implemented with keras (<https://github.com/SZUHvern/D-UNet>)
- [3] Jose Bernal, Kaisar Kushibar, Daniel S Asfaw, Sergi Valverde, Arnau Oliver, Robert Martí, and Xavier Lladó. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial intelligence in medicine*, 2018.
- [4] Julie A. Fiez, Hanna Damasio, and Thomas J. Grabowski. Lesion segmentation and manual warping to a reference brain: Intra- and interobserver reliability. *Human Brain Mapping*, 9(4):192–211, 2000.
- [5] Z Lao, D Shen, D Liu et al., "Computer-assisted segmentation of white matter lesions in 3d mr images using support vector machine", *Academic radiology*, no. 3, pp. 300-313, 2008.
- [6] J Mitra, P Bourgeat and J Fripp, "Lesion segmentation from multimodal mri using random forest following ischemic stroke", *NeuroImage*, pp. 324-335, 2014.
- [7] A Krizhevsky, I Sutskever and G E Hinton, "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, pp. 1097-1105, 2012.
- [8] M E Payne, D L Fetzer, J R MacFall et al., "Development of a semi-automated method for quantification of mri gray and white matter lesions in geriatric subjects", *Psychiatry Research Neuroimaging*, pp. 63-77, 2002.
- [9] S Datta, B R Sajja, R He et al., "Segmentation and quantification of black holes in multiple sclerosis", *Neuroimage*, pp. 467-474, 2006.
- [10] F Dutil, M Havaei, C Pal and H Larochelle, "A convolutional neural network approach to brain lesion segmentation", *Ischemic Stroke Lesion Segmentation*, a. pp. 13, 2013.
- [11] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147, 2016.
- [12] Ito K L, Kim H, Liew S L. A comparison of automated lesion segmentation approaches for chronic stroke T1-weighted MRI data[J]. bioRxiv, 2018: 441451.
- [13] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(12): 2481-2495.
- [14] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2881-2890.

- [15] Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]. Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.
- [16] Mosinska A, Marquez-Neila P, Koziski M, et al. Beyond the pixelwise loss for topology-aware delineation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 3136-3145.

References

- [1] Yongjin Zhou et al., D-UNet: a dimension-fusion U shape network for chronic stroke lesion segmentation (arXiv:1908.05104 [eess.IV]), 2019 Aug
- [2] SZUHvern github source code implemented with keras
(<https://github.com/SZUHvern/DUNet>)