**CHAPTER 1**

**INTRODUCTION**

* 1. **IMAGE PROCESSING**

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. The purposes of image processing are visualization, image sharpening and restoration, image retrieval, measurement of pattern and image recognition. The various image processing techniques are image pre-processing, image enhancement, image segmentation, feature extraction and image classification.

In image pre-processing the image data is improved by suppressing unwanted distortions. Image enhancement is the process of improving the quality of a digitally stored image by manipulating the image with software. Image segmentation is the process of partitioning a digital image into multiple segments. Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. Image classification refers to the process of categorizing all pixels in a digital image into one of several land covers classes.

Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities.

**1.2 DEEP LEARNING**

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, machine vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Most modern deep learning models are based on artificial neural networks, specifically, Convolutional Neural Networks (CNN)s, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own.

* 1. **SYSTEM OVERVIEW:**

Pancreas Segmentation is done using Abdominal CT-Scans. The various stages of process are been executed for accuracy. The first step of the process is Image Segmentation. The Abdominal CT-Scan is analysed and processed. Masking is used to process denoising mechanism which is been done to reduce the image noise and improve the image segmentation results. The designed U-NET architecture is providing greater efficiency in the process of segmentation. This can really help in finding abnormalities in the pancreas which is burdensome to doctors. They can also be used to diagnose some crucial disease such as pancreatitis, pancreatic cancer.

* 1. **SCOPE OF THE PROJECT:**

Segmentation and Masking of the Abdominal CT scan is crucial and vital in finding or locating the pancreas. Proportion of Pancreas is also important as it differs from each individual. Abnormalities in and around the pancreas can also be analyzed. These can help diagnose some crucial disease such as pancreatic cancer, pancreatitis.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] Dr.Bindhu V,”BIOMEDICAL IMAGE ANALYSIS USING SEMANTIC SEGMENTATION”, 2019.**

Semantic Segmentation is a very active area of research in the examining the medical images. The failure in the conventional segmentation methods to preserve the full resolution throughout the network led to the research’s that developed methods to protect the resolution of the images. The proposed method involves the semantic segmentation model for the biomedical images by utilizing the encoder/decoder structure to down sample the spatial resolution of the input data and develop a lower resolution feature mapping that are very effective at distinguishing between the classes and then perform the up samples to have a full-resolution segmentation map of the biomedical images reducing the diagnostic time.

The frame work put forth utilizes a pixel to pixel fully trained cascaded convolutional neural network for the task of image segmentation. The evaluation biomedical image analysis using the semantic segmentation shows the performance improvement achieved by the minimization of the time required in testing and the augmentation in the analysis performed by the radiologist.

The biomedical segmentation produces a very high performance with the rational amount of training epochs at a rational training time. The down-sample and up-sample method reduces the image resolution which might affect the accuracy of the segmentation.

**[2] Ozan Oktay, Jo SchlemperLoic Le, Folgoc Matthew, LeeMattias Heinrich,” Attention U-Net Learning Where to Look for the Pancreas”, 2018.**

A novel attention gate (AG) model for medical imaging that automatically learns to focus on target structures of varying shapes and sizes. Models trained with AGs implicitly learn to suppress irrelevant regions in an input image while highlighting salient features useful for a specific task. This enables us to eliminate the necessity of using explicit external tissue/organ localisation modules of cascaded convolutional neural networks (CNNs). AGs can be easily integrated into standard CNN architectures such as the U-Net model with minimal computational overhead while increasing the model sensitivity and prediction accuracy. The proposed Attention U-Net architecture is evaluated on two large CT abdominal datasets for multi-class image segmentation. Experimental results show that AGs consistently improve the prediction performance of U-Net across different datasets and training sizes while preserving computational efficiency. The code for the proposed architecture is publicly available.

**[3] Dr. Olaf Ronneberger Mr.Philipp Fischer and Dr.Thomas Brox,” U-Net: Convolutional Networks for Biomedical Image Segmentation“,2015**

There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, The authors have present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. They show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) They won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU.

**[4] Ange Lou, Shuyue Guan, Murray Loew*,*” DC-U Net: Rethinking the U-Net Architecture with Dual Channel Efficient CNN for Medical Image Segmentation”,2020.**

Recently, deep learning has become much more popular in computer vision area. The Convolution Neural Network (CNN) has brought a breakthrough in images segmentation areas, especially, for medical images. In this regard, U-Net is the predominant approach to medical image segmentation task. The U-Net not only performs well in segmenting multimodal medical images generally, but also in some tough cases of them. However, the authors have discovered that the classical U-Net architecture has limitation in several aspects. Therefore, we applied modifications: 1) designed efficient CNN architecture to replace encoder and decoder, 2) applied residual module to replace skip connection between encoder and decoder to improve based on the-state-of-the-art U-Net model. Following these modifications, they designed a novel architecture--DC-UNet, as a potential successor to the U-Net architecture. They created a new effective CNN architecture and build the DC-UNet based on this CNN. The evaluation of their model on three datasets with tough cases and have obtained a relative improvement in performance of 2.90%, 1.49% and 11.42% respectively compared with classical U-Net. In addition, they used the Tanimoto similarity to replace the Jaccard similarity for grey-to-grey image comparisons.

**[5] Jun Gao, Qian Jiang, Bo Zhou, Daozheng Chen,”Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis“, 2019.**

Computer-aided detection or diagnosis (CAD) has been a promising area of research over the last two decades. Medical image analysis aims to provide a more efficient diagnostic and treatment process for the radiologists and clinicians. However, with the development of science and technology, data interpretation manually in the conventional CAD systems has gradually become a challenging task. Deep learning methods, especially convolutional neural networks (CNNs), are successfully used as tools to solve this problem. This includes applications such as breast cancer diagnosis, lung nodule detection and prostate cancer localization. In this overview, the current state-of-the-art medical image analysis techniques in CAD research are presented, which focus on the convolutional neural network (CNN) based methods. The commonly used medical image databases in literature are also listed. It is anticipated that this paper can provide researchers in radiomics, precision medicine, and imaging grouping with a systematic picture of the CNN-based methods used in CAD research.

**[6] Holger R. Roth, Amal Farag, Le Lu, Evrim B. Turkbey, and Ronald M. Summers,“ Deep convolutional networks for pancreas segmentation in CT imaging”, 2015.**

Automatic organ segmentation is an important prerequisite for many computer-aided diagnosis systems. The high anatomical variability of organs in the abdomen, such as the pancreas, prevents many segmentation methods from achieving high accuracies when compared to other segmentation of organs like the liver, heart or kidneys. Recently, the availability of large annotated training sets and the accessibility of affordable parallel computing resources via GPUs have made it feasible for "deep learning" methods such as convolutional networks (ConvNets) to succeed in image classification tasks. These methods have the advantage that used classification features are trained directly from the imaging data. They authors present a fully-automated bottom-up method for pancreas segmentation in computed tomography (CT) images of the abdomen. The method is based on hierarchical coarse-to-fine classification of local image regions (super pixels). Super pixels are extracted from the abdominal region using Simple Linear Iterative Clustering (SLIC). An initial probability response map is generated, using patch-level confidences and a two-level cascade of random forest classifiers, from which super pixel regions with probabilities larger 0.5 are retained. These retained super pixels serve as a highly sensitive initial input of the pancreas and its surroundings to a Conv-Net that samples a bounding box around each super pixel at different scales (and random non-rigid deformations at training time) in order to assign a more distinct probability of each super pixel region being pancreas or not. They evaluate our method on CT images of 82 patients (60 for training, 2 for validation, and 20 for testing). Using ConvNets we achieve average Dice scores of 68%+-10% (range, 43-80%) in testing. This shows promise for accurate pancreas segmentation, using a deep learning approach and compares favourably to state-of-the-art methods.

**[7] Dr.S. Kannan, Vairaprakash, Gurusamy, G.Nalini” Review on Image Segmentation Techniques”,2015.**

Many image segmentation techniques are available in the literature. Some of these techniques use only the grey level histogram, some use spatial details while others use fuzzy set theoretic approaches. Most of these techniques are not suitable for noisy environments. Some works have been done using the Markov Random Field (MRF) model which is robust to noise, but is computationally involved. Neural network architectures which help to get the output in real time because of their parallel processing ability, have also been used for segmentation and they work fine even when the noise level is very high. The literature on colour image segmentation is not that rich as it is for grey tone images. This paper critically reviews and summarizes some of these techniques. Attempts have been made to cover both fuzzy and non-fuzzy techniques including colour image segmentation and neural network based approaches. Adequate attention is paid to segmentation of range images and magnetic resonance images. It also addresses the issue of quantitative evaluation of segmentation results.

**[8] Rina Komatsu and Tad Gonsalves” Effectiveness of U-NET in Denoising RGB images “,2019.**

Digital images often contain “noise” which takes away their clarity and sharpness. Most of the existing denoising algorithms do not offer the best solution because there are difficulties such as removing strong noise while leaving the features and other details of the image intact. Faced with the problem of denoising, they tried solving it with a Convolutional Neural Network architecture called the “U-Net”. This paper deals with the training of a U-Net to remove 3 different kinds of noise: Gaussian, Blockiness, and Camera shake. Their results indicate the effectiveness of U-Net in denoising images while leaving their features and other details intact.

**[9] Jonathan Long, Evan Shelhamer, Trevor Darrell.,“ Fully Convolutional Networks for Semantic Segmentation “,2011.**

Convolutional networks are powerful visual models that yield hierarchies of features. They explain that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Their key insight is to build "fully convolutional" networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. They define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. Their work on this paper is to adapt contemporary classification networks (AlexNet, the VGG net, and GoogLeNet) into fully convolutional networks and transfer their learned representations by fine-tuning to the segmentation task. Then define a novel architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations. Their fully convolutional network achieves state-of-the-art segmentation of PASCAL VOC (20% relative improvement to 62.2% mean IU on 2012), NYUDv2, and SIFT Flow, while inference takes one third of a second for a typical image.

**[10] Pedro M.Sosa.,” Twitter Sentiment Analysis using Combined LSTM –CNN Models,”2017.**

An advanced model which is based on the LSTM-CNN model presented by Pedro M. Sosa for twitter sentiment analysis. They combined the encoder-decoder framework with the regular LSTM-CNN framework. In this model, LSTM can `remember' forward information of the sequence and multilayer CNN can catch and learn local information sufficiently. Meanwhile, the multilayer CNN is also regarded as an encoder and a two-layer deconvolution part is the corresponding decoder. This encoder-decoder framework is used to reconstruct the input matrix, this process of the reconstruction of input matrix by decoder makes the features learning in CNN much more intrinsic and effective. As the result, the more effective the feature learning is, the higher accuracy rate the classifier will achieve. Furthermore, their framework can also be used for other classification issues besides sentiment analysis. This work will make sense in fields such as machine learning and natural language processing.

**[11] The Effectiveness of Data Augmentation in Image Classification using Deep Learning**, **Jason Wang -** **Stanford University**

In this paper, The author Explore's and compare’s multiple solutions to the problem of data augmentation in image classification. Previous work in image augmentation has demonstrated the effectiveness of data augmentation through simple techniques, such as cropping, rotating, and flipping input images. They artificially constrain our access to data to a small subset of the ImageNet dataset, and compare each data augmentation technique in turn. One of the more successful data augmentations strategies is the traditional transformations mentioned above. They have also experiment with GANs to generate images of different styles. Finally, propose a method to allow a neural net to learn augmentations that best improve the classifier, which we call neural augmentation. The authors discuss the successes and shortcomings of this method on various datasets. Dataset includes GTSRDC Road traffic signs dataset provided by traffic cooperation department of the German country.

**[12] José V. Manjón ,” MRI Preprocessing”, Springer International Publishing Switzerland 2017 53 L. Martí-Bonmatí, A. Alberich-Bayarri (eds.), Imaging Biomarkers, 2017**

Standardized space making possible meaningful comparisons has done to set the images using image acquisition and image observation. The preprocessing techniques have been developed using denoising and inhomogeneity correction. Current state-of-the-art denoising methods were based on patch wise image preprocessing approaches exploiting sparseness or self-similarity properties of the medical images or both. The method can also be useful if a super resolution step has been applied to increase the resolution of image processes.

**[13] Matthew D. Zeiler and Rob Fergus,” Visualizing and Understanding Convolutional Networks”, 2016.**

Large Convolutional Network models have recently demonstrated impressive classification performance on the ImageNet benchmark Krizhevsky et al. [18]. However there is no clear understanding of why they perform so well, or how they might be improved. In this paper the authors explore both issues. They introduce a novel visualization technique that gives insight into the function of intermediate feature layers and the operation of the classifier. Used in a diagnostic role, these visualizations allow us to find model architectures that outperform Krizhevsky et al. on the ImageNet classification benchmark. They also perform an ablation study to discover the performance contribution from different model layers. They show that, ImageNet model generalizes well to other datasets: when the SoftMax classifier is retrained, it convincingly beats the current state of the art results on Caltech-101 and Caltech-256 datasets.

**[14] Bharath Hariharan1 , Pablo Arbelaez ´ 1 , Lubomir Bourdev1,2 , Subhransu Maji1 and Jitendra Malik,” Semantic Contours from Inverse Detectors”,2018.**

The paper is about to study the challenging problem of localizing and classifying category-specific object contours in real world images. For this purpose, the authors have presented a simple yet effective method for combining generic object detectors with bottom-up contours to identify object contours. They also provide a principled way of combining information from different part detectors and across categories. In order to study the problem and evaluate quantitatively our approach, they present a dataset of semantic exterior boundaries on more than 20, 000 object instances belonging to 20 categories, using the images from the VOC2011 PASCAL challenge.

**[15] Survey over image thresholding techniques and quantitative performance evaluation, Mehmet Sezgin , Bu¨lent Sankur.**

They conducted an exhaustive survey of image thresholding methods, categorize them, express their formulas under a uniform notation, and finally carry their performance comparison. The thresholding methods are categorized according to the information they are exploiting, such as histogram shape, measurement space clustering, entropy, object attributes, spatial correlation, and local grey-level surface. 40 selected thresholding methods from various categories are compared in the context of non-destructive testing applications as well as for document images. The comparison is based on the combined performance measures. They identified that the thresholding algorithms that perform uniformly better over non-destructive testing and document image applications.

**CHAPTER 3**

**SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM:**

Analyzing the medical image in image processing is the most important research area. Capturing the image are analyzed to identify different medical imaging problems is the common factor in this field. Robust organ segmentation is a prerequisite for computer-aided diagnosis (CAD), quantitative imaging analysis, pathology detection and surgical assistance. Some of the organs in the human body have high anatomical variability, so segmentation of such organs is very complex. The existing system segments the pancreas with the considerations of spatial relationships of splenic, portal and superior mesenteric veins with the pancreas. The proposed system uses macro super-pixels for fast and deep labeling and segmentation process. The existing system is an automated bottom-up approach for pancreas segmentation with the consideration of spatial relationships with the veins in abdominal computed tomography (CT) scans. The method generates dynamic cascaded and macro super-pixel segmentation information’s by classifying image patches at different resolutions. Fast organ analysis using Dense-SIFT algorithm.

* + 1. **Disadvantages of the Existing System:**
* The result is based on the image quality. It sometimes needs a measure of 'how much' the image qualifies as a clarity and density of pixels. (e.g., a low-quality CT scan on the mix might affect the overall accuracy.)
* The number of parameters leads to under-fitting of the developed model for getting just that little better result.
* One still needs higher storage to connect the resulting model and to extract the complete result, the model must be executed and stored with the dataset for obvious improvisations.
* Also due to the Dense-SIFT algorithm the execution is Still quite slow (SURF provides similar performance while running faster) and the model Generally doesn't work well with lighting changes and blur.
  1. **PROPOSED SYSTEM:**

The aim of proposed system is to develop a system of improved facilities and accuracy in predicting the pancreas from the abdominal CT scan. The proposed system works on the Deep Learning architecture specifically designed for the medical image segmentation and provides better accuracy with memory and space constrains. The model can overcome all the limitations of the existing system and works with minimal RAM support for the system. The system can also be executed over the cloud ecosystem making it platform independent and removes the problematic limitations of memory constrains. The proposed system reduces the manual work required to setup each system to execute the algorithm and also hardware independency of the model makes its stand over the universal implication. The model can also be upgraded with additional images to increase the overall accuracy of the model on the production environment.

* + 1. **Advantages of the Proposed System:**
* The U-Net model is a specific model developed for medical image segmentation and a light weight model. It can process through the images faster than other stable Deep Learning Architecture models.
* The space complexity of the U-Net architecture model is much less than the compared model.
* The data processing stage provides a better advantage to provide well segregated and annotated image for the model to process.
* The accuracy of the proposed system is slightly greater than the existing system.
  1. **REQUIREMENTS SPECIFICATION:**

**3.3.1 Hardware Requirements**:

* Hard Disk: 15GB and above.
* Cloud drive 30GB and above.
* Processor: intel core i3 5th gen and above.
* RAM:  4GB and above.

**3.3.2 Software Requirements**

* Windows operating system 7 and above
* Google Colab with any web browser to support.
* Python Version 3.0.0, TensorFlow 2.0 and above.

**3.4 LANGUAGE SPECIFICATION**

**3.4.1 PYTHON 3.0.0 and above:**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python-3.

The Python 2 language, i.e., Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains C Python, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and C Python development.

**3.4.2 TensorFlow 2.0:**

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 in 2015. TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (archived and development has ceased). Third-party packages are available for C#, Haskell, Julia, MATLAB, R, Scala, Rust, OCaml, and Crystal.

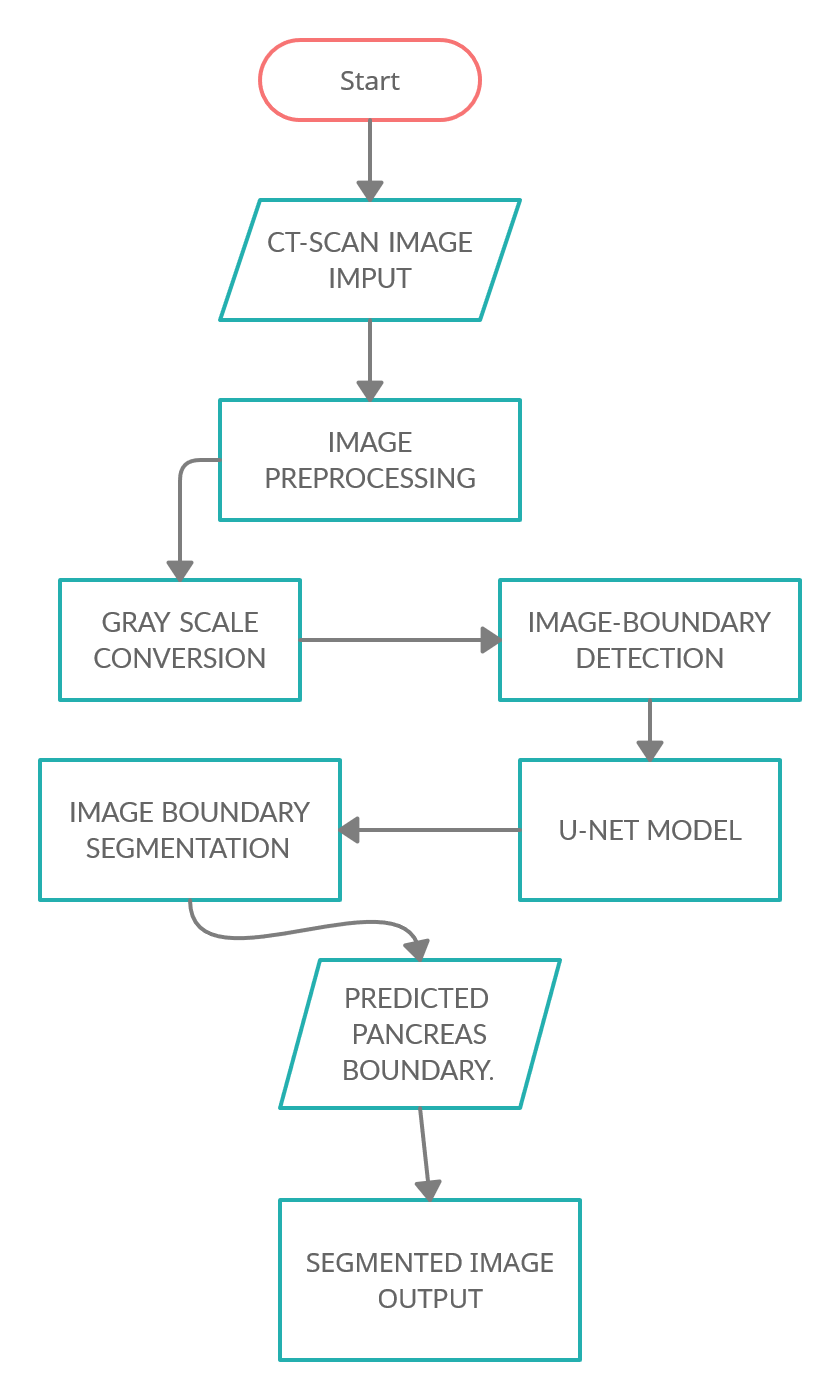
TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs. TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs. TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

**CHAPTER 4**

**SYSTEM DESIGN:**

**4.1 SYSTEM ARCHITECTURE**:

****

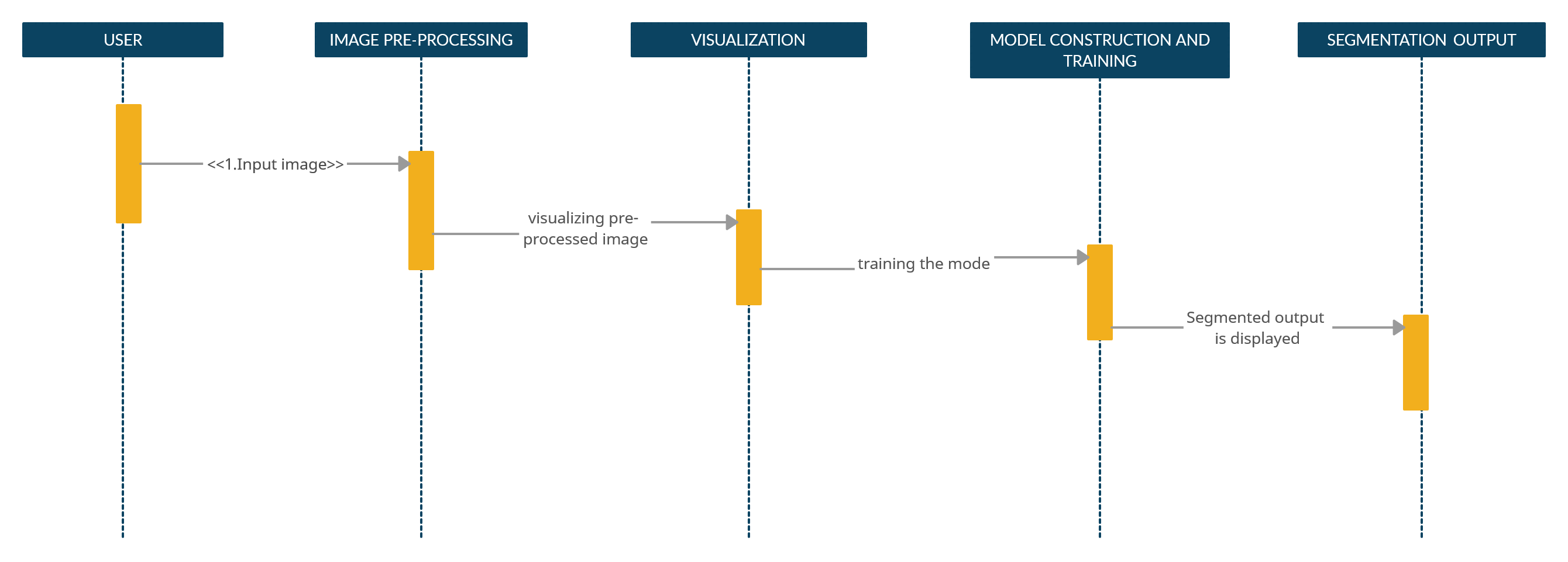
**Figure 4.1 Architecture of proposed system**

In the figure 4.1, the input is given as an CT scan (computed tomography) image combines a series of X-ray images taken from different angles around your body and uses computer processing to create cross-sectional images (slices) of the bones, blood vessels and soft tissues inside your body. CT scan images provide more-detailed information than plain X-rays do.

In order to obtain an efficient segmented image, the input image is been pre-processed by some procedures involved. By preprocessing, the input is been detected by grey scale conversion technique and binary mask conversion. In the image boundary detection module, the image is annotated using a tqdm progress bar and the test image is visualized randomly to view the mask and boundary image.

Then the train, test, and validation set image folders are split and moved to the u-net model for training. The u-net model trains the required model batch wise for the specified epochs and returns the best model to test with the test image folder. The model is saved and then loaded again back to the kernel and the image segmentation takes place. The final predicted segmentation n image is displayed as the output with higher level accuracy from both test and validation set images. By these techniques involved the output is been developed as the segmented image.

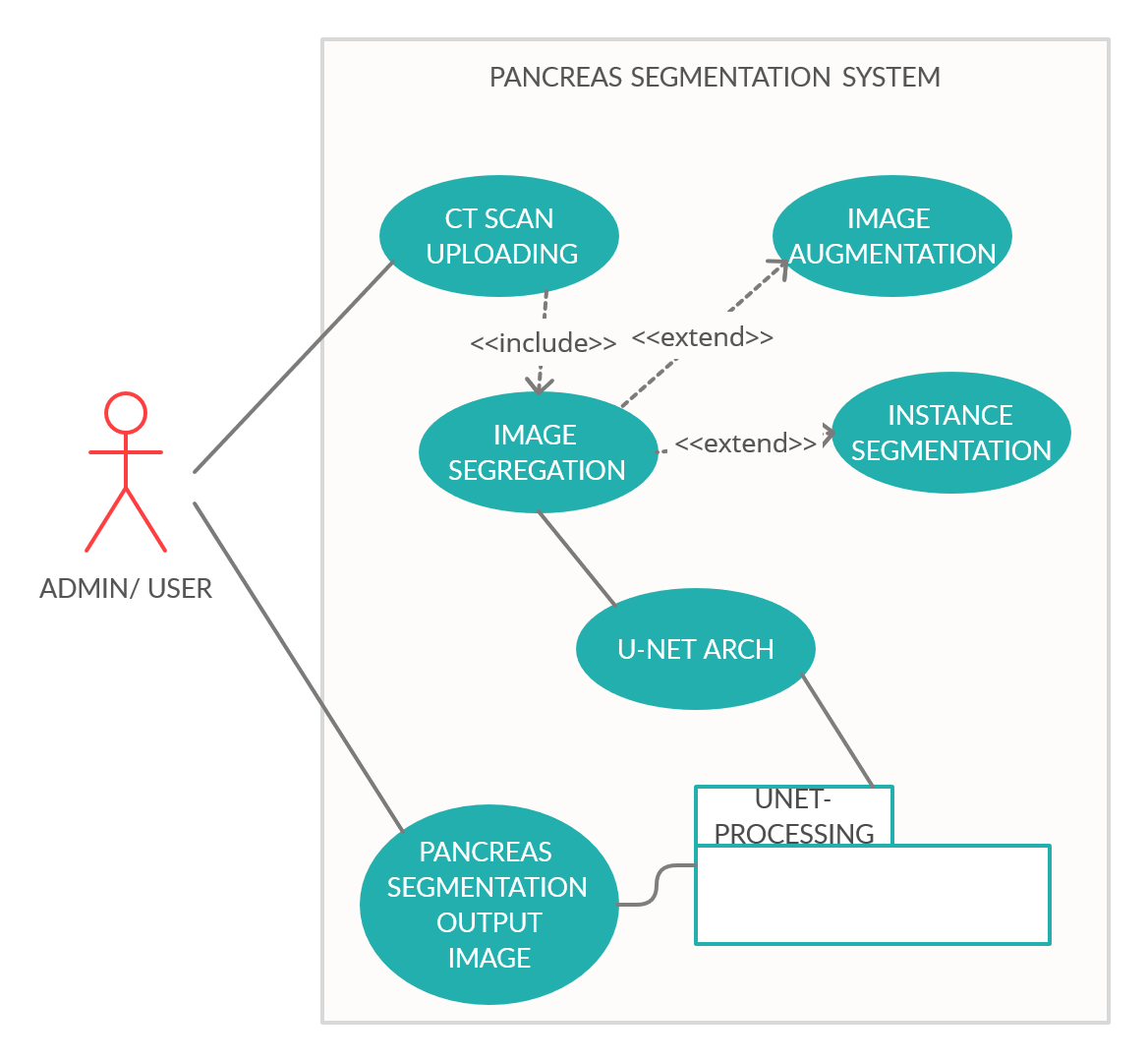
**4.2 SEQUENCE DIAGRAM:**

****

**FIGURE 4.2 SEQUENCE DIAGRAM**

In Figure 4.2, the input image is given by the user to the image pre-processing module. The image pre-processing module converts normal image to grey image and annotates the training images and their respective mask images under different folders but in the same sequence. Once the image is loaded, the training image with mask is given as input to u-net architectural model. The u-net model trains the images and develops the required model for the image segmentation to take place in the next module. The final module checks the model accuracy with the test set and also provides the output of the validation set image in a random order. Once the final module is complete, the segmentation output of the pancreas is displayed.

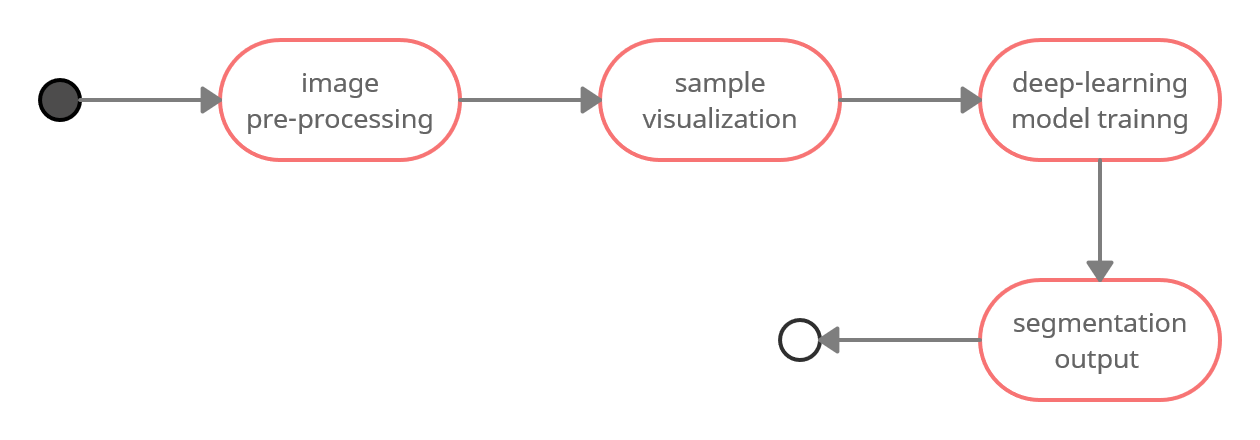
* 1. **USE CASE DIAGRAM:**

****

**FIGURE 4.3 USE CASE DIAGRAM**

In the Figure 4.3, the input image is given to the image segregation module, where the images are spilt into train, test, and validation sets. Once the previous module is complete, the images are loaded into the system and its progress is tracked using tqdm bar. Then, the train folder is provided as input to the deep learning model. Secondly the u-net provides a trained model which is finally utilized to generate the output from test and validation set. The final segmentation output is displayed.

* 1. **STATE DIAGRAM:**

****

**FIGURE 4.4 STATE DIAGRAM**

In the Figure 4.4, the taken image is subjected to different states. Initially, it is subjected to input state which takes the image for further processing. After the completion of input state, the image is subjected to sample visualization state. At this state, the image is initially displayed along with its mask image. The preprocessed image is given to deep learning model. The model processes the training images in the set and provides a model. The model is then subjected to the result state and it provides with the segmentation of the final image form the test and validation set. The result is given to the output state which will display it to the user.

**CHAPTER 5**

**MODULE DESCRIPTION**

**5.1 MODULES:**

The modules are:

* + Dataset Preparation.
  + Image pre-processing and Visualization initialization.
  + Constructing U-NET model and training it.
  + Segmentation – Visualization of output.

**PRE-PROCESSING:**

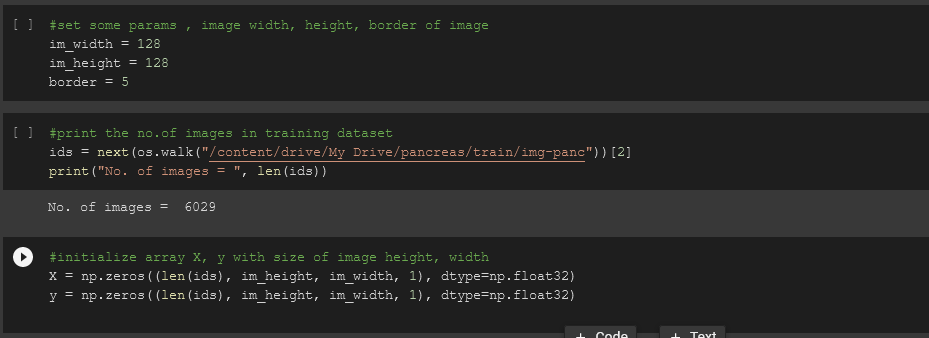
Pre-processing is a common name for operations with images at the lowest level of abstraction - both input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g., rotation, scaling, and translation) are classified among pre-processing methods here since similar techniques are used.

Image Preprocessing is used to:

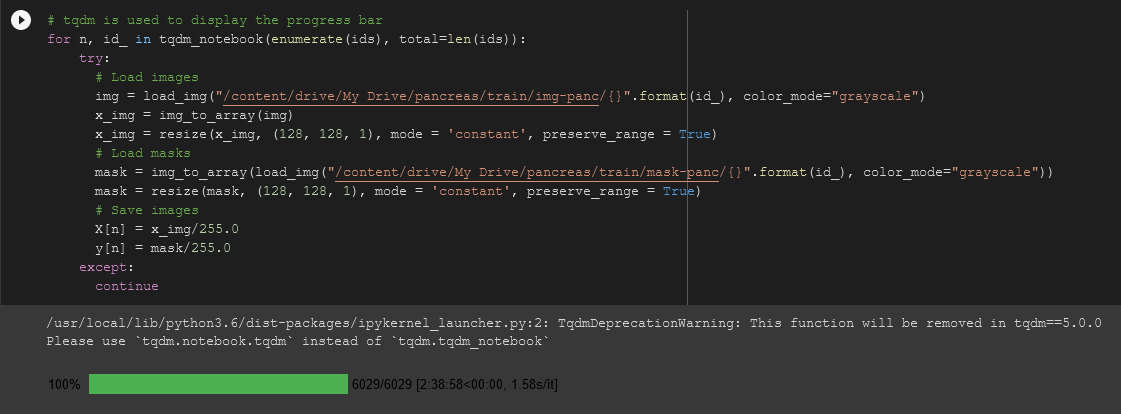
* Remove noise.
* Sharpen contrast.
* Highlight contours.
* Enhance image.
* Produce mask images.
* Sequential ordering of images.

The pre-processing step is accompanied by the **Dataset Preparation** module and the module initiates the flow of the project.

**5.1.1 Dataset Preparation:**

****

**Fig 5.1. Dataset preparation and primary parameter declaration.**



**Fig 5.2 Image annotation and pre-processing code snippet.**

In this module, the images collected from various sources are pooled together into one single folder before processing any images. After collecting the necessary counts of images, the total folder is processed to find the total number of images. Then, the basic necessary parameters such as image height, width and border are initialized to convert all images into similar dimensions. (as in fig. 5.1)

Furthermore, multiple one-dimensional arrays are initialized with float (32) as their datatype to simplify the image processing carried out by the deep learning model. The array plays an important role in the time necessary to process all the images together in the u-net model. Initially, the arrays are populated with zeros, once the images are converted then they are populated with the image vector points.

Secondly, the images are enumerated and progress is tracked using a tqdm bar (as in fig. 5.2). Every image is individually loaded, resized and saved back into the folder in the same sequence it was retrieved. The same process takes place for the mask images. The images are converted into Grey-scale images, which helps in the future process to colour code and output the segmented images. The images are resized to maintain uniformity. Finally, the collection of images is split into 3 different sub folders namely, train, test and validation image sets.

* + 1. **Image pre-processing and Visualization initialization:**

The images from the training set and test set are visualized on random basis to check if the images are named and annotated according to their respective mask images from the folder. To do this, a random image is chosen from the dataset and first it is put through the pancreas threshold code. If the image passes the threshold test, then a size of the image is enlarged temporally and is provided as an output to the function. Before displaying the result of the module, boundary (contour) in the original image separating Pancreas and non-Pancreas areas is implemented for easy visualization. After that, the respective mask image is also displayed in the following method.

Sample code:

# Visualize any randome image along with the mask

ix = random.randint(0, len(X\_train))

has\_mask = y\_train[ix].max() > 0 # Pancreas indicator

fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (20, 15))

ax1.imshow(X\_train[ix, ..., 0], cmap = 'plasma', interpolation = 'bilinear')

if has\_mask: # if Pancreas

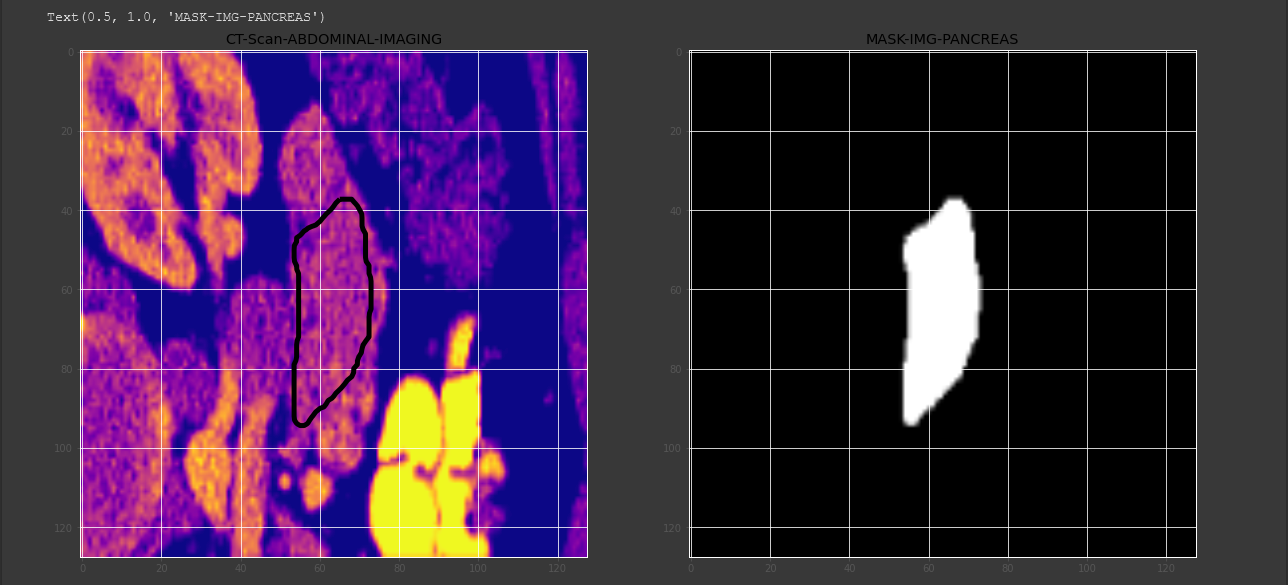
# draw a boundary(contour) in the original image separating Pancreas and non-Pancreas areas

ax1.contour(y\_train[ix].squeeze(), colors = 'k', linewidths = 5, levels = [0.5])

ax1.set\_title('CT-Scan-ABDOMINAL-IMAGING')

ax2.imshow(y\_train[ix].squeeze(), cmap = 'grey', interpolation = 'bilinear')

ax2.set\_title('MASK-IMG-PANCREAS')



**Fig.5.3 initial visualization result**

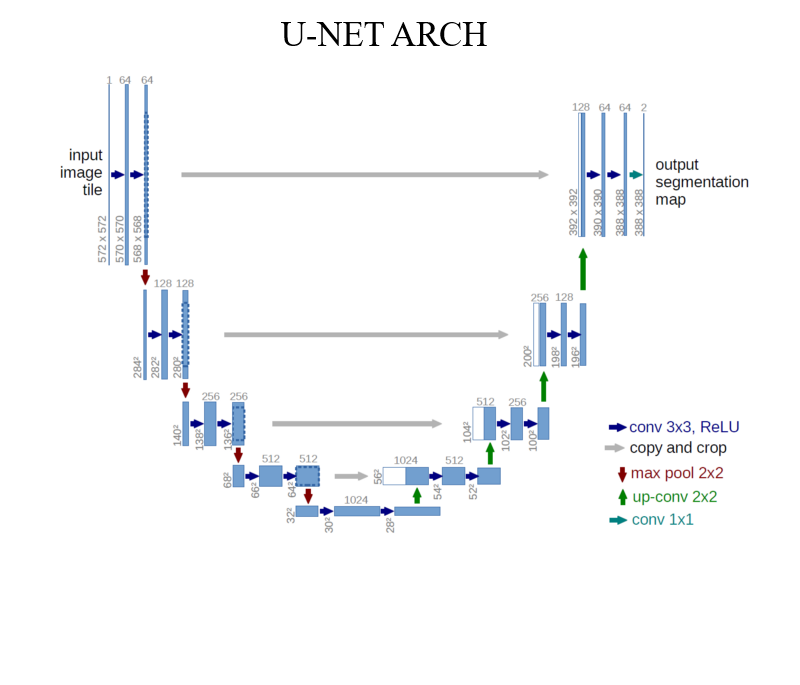
In the figure 5.3, The left part of the image depicts the original abdominal CT – Scan presented in a plasma interpretation for boundary visualization simplification. Thus, it provides much simpler view to any person, even without prior knowledge of any medical imaging reading and understanding. The left image is the mask image of the pancreas, which is processed with binary filter and edge boundary detection for a closed regional image display as the mask image output. The respective mask image is displayed automatically, since it was pre-processed and annotated in a sequential order.

* + 1. **Constructing U-NET model and training it:**

A static 2-D convolutional block is created with “relu” activation function, and its function is to add 2 convolutional layers with parameters passed to it. The model was tested with 4 different types of activation functions such as “relu, LeakyRelu, sigmoid, tanh” etc., and the best model accuracy was based on relu activation function.

A nine-layer U-NET architecture model is developed with the 2-D convolutional block function created before. The first four layers are used as Contraction path and the last four layers are used for Expansive path. The images are down-sampled from 128\*128 to 1\*1 and each pixel is evaluated closely. Thus, proving u-net arch is the best architecture for medical image segmentation. Then, finally the 1\*1 is up sampled back to 128\*128 using the expansive path.

The model is trained with 30 Epochs with specific condition to exit if the accuracy of the model does not improve over a period of time. The call-back function is created to avoid excess usage of RAM when not required and to reduce the time of execution. It also provides another benefit of reducing the implementation time cost when deployed and trained the model over the cloud infrastructure of Google Cloud Colab for machine learning. The final model is saved and reloaded into the script for testing and output visualization. The image is also tested for the threshold, such that only the images form the test and validation set which crosses it would definitely have the clear pancreas segmentation. This also helps as a cost cutting mechanism in cloud deployment.



**Fig 5.4 U-Net Architecture Diagram.**

**Input design**:

The design of input (as in figure 5.4) focuses on controlling the image provided to the u-net model. The input is a seismic image extracted from seismic imaging technique. this image is further passed into the u-net block which in turn access multiple unit CNN blocks into the system. These units process the image as described in the architecture. The image is down-sampled from a high resolution of 255 \* 255 to a unit cell block. The block is analyzed and processed in that format for simplicity and to reduce time complexity.

**Output Design:**

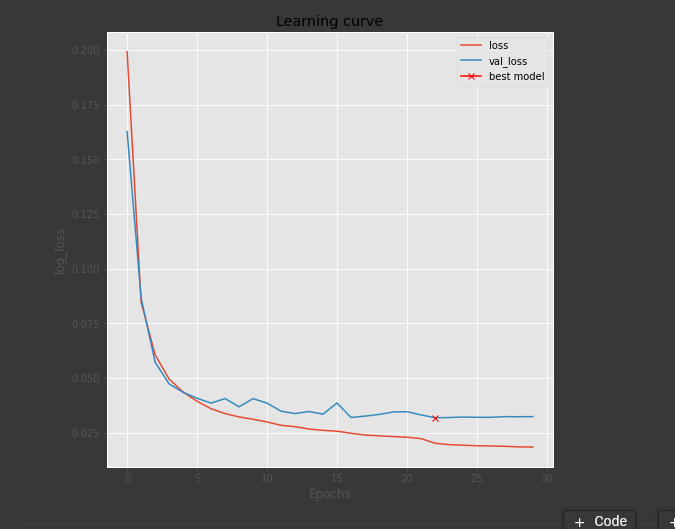
A quality output is one, which meets the requirements of the end user and presents the information clearly. In output is design in such a way that the accuracy of the model is improvised for each iteration of model training. The down-sampled image of resolution dimension 1 \* 1 is up-sampled again after processing in its unit size.

The image is further classified on basis of the training data provided and the salts boundary is drawn. The depicted output is displayed with an inbuild visualizing technique provided by the processing programming language itself. The boundary splits the salt and no salt region by image instance segmentation.

**5.1.4 Segmentation – Visualization of Output:**

The trained model is saved and the model weight is also stored, in-order to use the model for any future usage. The trained model’s loss and validation loss is plotted in a 2-D graph chart (fig 5.5) to provide a clear picture of the model’s process. The threshold predictions on the test and validation dataset are processed and completed.

A final function to plot the data and to check the prediction is developed. The segmented images are displayed in 4 different visualization models [ Seismic, normal, border prediction, and binary model]. The final segmentation of the pancreas from the abdominal CT Scan is displayed as the resultant output (refer fig 5.6, 5.7). A random image is also visualized from the validation set to verify its validity over the training and test images.



**Fig. 5.5 Training and Validation loss plot.**

Sample code :

#static convolutional neural network model to callback as fn for U-Net model

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:

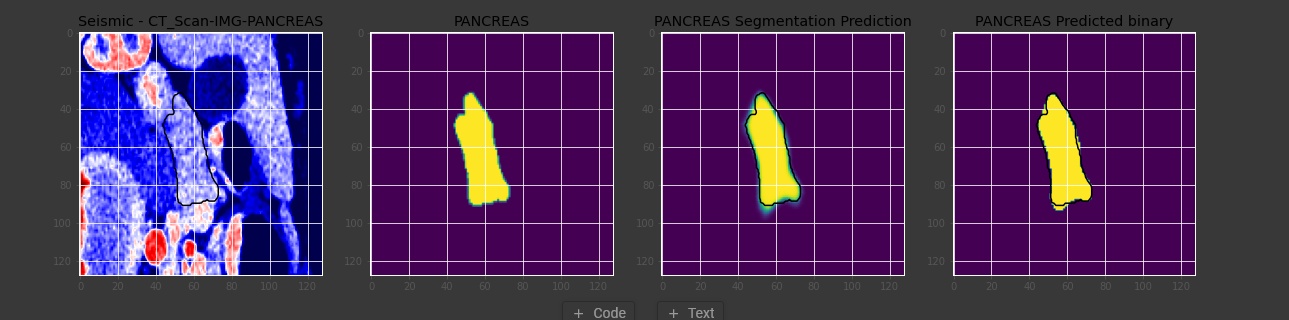
x = BatchNormalization()(x)

x = Activation('relu')(x)

return x

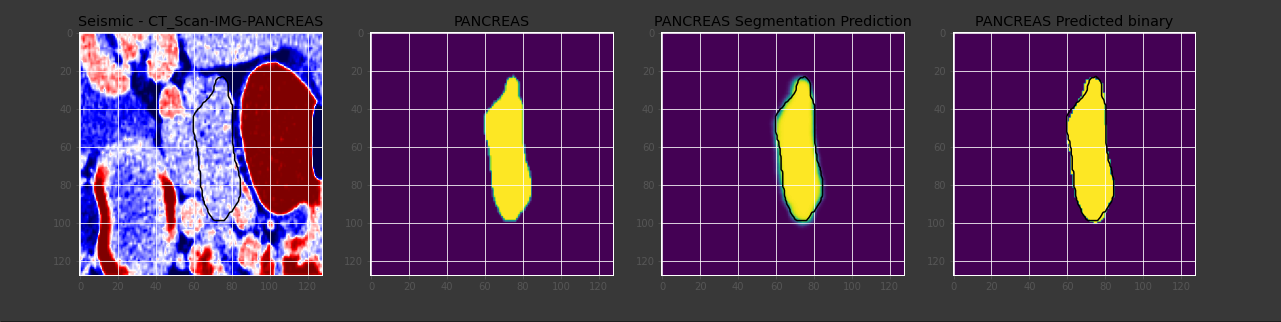
**Final Output:**

* 1. **Test set image:**



**Fig 5.6 Final output visualization 1.**

* 1. **Validation set image:**

****

**Fig 5.7 Final output visualization 2.**

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

* 1. **CONCLUSION**

For an image, pre-processing is necessary. Initially, an image is converted to Grey image using Grey scale image converter. The pre-processing steps include intensity normalization, background removal by thresholding method and masking. Using U-NET architecture, those masked images are then segmented based on the scans with boundary segmentation method. Here CT-Scans are converted into Seismic images as it provides more clear resolution which helps segmenting it more efficiently. A Nine-layer U-NET architecture is defined with attributes such as filters and dropout. The model is compiled with Adam optimizer and binary\_crossentropy as loss function. A call-back function with maximum epoch of 30 is created to save the best model. After saving the best model in h5 extension which can be used after to evaluate and predict. Threshold level of 0.5 is set so that the seismic image is of required clarity for segmentation. Plotting of data/image is done to display the image and segmented image and also the predicted binary.

**6.2 FUTURE ENHANCEMENT**

This project is based on segmentation of pancreas in Abdominal CT-Scan so as future work, we are trying to enhance segmentation into prediction model which helps in diagnosing crucial diseases such as pancreatitis, pancreatic cancer. Further enhancement is to deploy the model in Cloud based system and developing application with tight constraints so that every doctor can only know about their patient’s record using their mobile number without modifying the data. Furthermore, Natural Language Processing (NLP) can be used to extract text from scanned medical prescription providing effective embedded terms and categories about patient data which help physicians make better decisions as it reduces medical errors and improved treatment method.

**APPENDIX 1**

**SAMPLE CODING:**

**U-NET ARCHICTURE:**

#standard u-net arch

def get\_unet(input\_img, n\_filters = 16, dropout = 0.1, batchnorm = True):

    """Function to define the UNET Model"""

    # 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

**SEGMENTATION OUTPUT FUNCTION :**

#function to plot the data and to check the predictions.

def plot\_sample(X, y, preds, binary\_preds, ix=None):

    """Function to plot the results"""

    if ix is None:

        ix = random.randint(0, len(X))

    has\_mask = y[ix].max() > 0

    fig, ax = plt.subplots(1, 4, figsize=(20, 10))

    ax[0].imshow(X[ix, ..., 0], cmap='seismic')

    if has\_mask:

        ax[0].contour(y[ix].squeeze(), colors='k', levels=[0.5])

    ax[0].set\_title('Seismic - CT\_Scan-IMG-PANCREAS')

    ax[1].imshow(y[ix].squeeze())

    ax[1].set\_title('PANCREAS')

    ax[2].imshow(preds[ix].squeeze(), vmin=0, vmax=1)

    if has\_mask:

        ax[2].contour(y[ix].squeeze(), colors='k', levels=[0.5])

    ax[2].set\_title('PANCREAS Segmentation Prediction')

    ax[3].imshow(binary\_preds[ix].squeeze(), vmin=0, vmax=1)

    if has\_mask:

        ax[3].contour(y[ix].squeeze(), colors='k', levels=[0.5])

    ax[3].set\_title('PANCREAS Predicted binary');

**Static CNN model to callback as function for U-Net model:**

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:

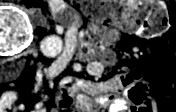
x = BatchNormalization()(x)

x = Activation('relu')(x)

return x

**APPENDIX 2**

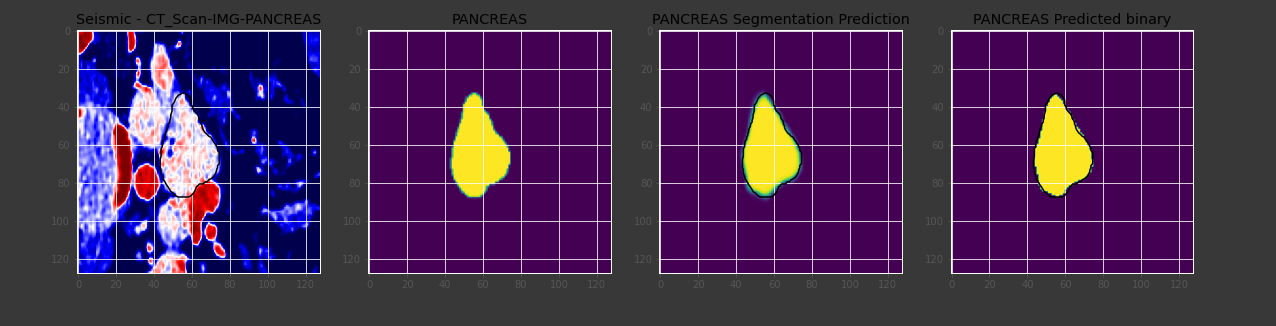
**SCREENSHOTS:**

****

**Fig. A2.1, Sample of original CT scan image.**

****

**Fig, A2.2, Sample of its respective mask image.**

****

**Fig, A2.3, Final output visualization in a random order.**

**REFERENCES**

[1] Bashar, Abul. "SURVEY ON EVOLVING DEEP LEARNING NEURAL NETWORK ARCHITECTURES." Journal of Artificial Intelligence 1, no. 02 2019

[2] Christ, Patrick Ferdinand, Mohamed Ezzeldin A. Elshaer, Florian Ettlinger, Sunil Tatavarty, Marc Bickel, Patrick Bilic, Markus Rempfler et al. "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields." In International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 415-423. Springer, Cham, 2016.

[3] Bai, W., Sinclair, M., Tarroni, G., Oktay, O., Rajchl, M., Vaillant, G., Lee, A.M., Aung, N., Lukaschuk, E., Sanghvi, M.M., et al.: Human-level CMR image analysis with deep fully convolutional networks.(2017)

[4] Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B.R., Pereira, S.P., Clarkson, M.J., Barratt, D.C.: Towards image-guided pancreas and biliary endoscopy: Automatic multi-organ segmentation on abdominal CT with dense dilated networks.(2017)

[5] Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS, pp. 2852–2860 (2012)

[6] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2014), arXiv:1411.4038 [cs.CV].

[7] Gao, J., Jiang, Q., Zhou, B., & Chen, D. (2019). Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis: An overview. Mathematical Biosciences and Engineering, 16(6), 6536

[8] Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Rad, A. E., Rehman, A., & Uddin, M. (2014). Medical image segmentation methods, algorithms, and applications. IETE Technical Review, 31(3), 199-213

[9] Tingxi Wen, Hanxiao Wu, Yu Du, Chuanbo Huang, Faster R-CNN with improved anchor box for cell recognition. Mathematical Biosciences and Engineering, 2020, 17(6): 7772-7786. doi: 10.3994/mbe.2020395

[10] Shuai Cao, Biao Song . Visual attentional-driven deep learning method for flower recognition. Mathematical Biosciences and Engineering, 2021, 18(3): 1981-1991.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, pp. 1097–1105, 2012.

[12] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, “Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network,” in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013, pp. 246–253, Springer, 2013.

[13] Jay Acharya, Sohil Gadhiya and Kapil Raviya, “Segmentation Techniques for Image Analysis: A Review”, International Journal of Computer Science and Management Research, Vol 2 Issue 1, January 2013, Pg. 1218-1221.

[14] Olaf Ronneberger, Philipp Fischer & Thomas Brox, (2015) “U-net: Convolutional Networks for Biomedical Image Segmentation”, International Conference on Medical image computing and computer-assisted intervention (MICCAI), Springer, Vol. 9351, pp234-241

[15] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In ICML, 201.

[16] Dinesh D. Patil and Sonal G. Deore , “Medical Image Segmentation: A Review”, International Journal of Computer Science and Mobile Computing, 2, pp. 22-27, 2013.

[17] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014, pages 818–833. Springer, 2014

[18] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In International Conference on Computer Vision (ICCV), 2011.

[19] M. Sundermeyer, R. Schl¨uter, and H. Ney, “LSTM neural networks for language modeling.,” in Interspeech, pp. 194–197, 2012.

[20] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014), arXiv:1409.1556 [cs.CV]

[21] Mehmet Sezgin and Bulent Sankur, “Survey over image thresholding techniques and quantitative performance evaluation”, Journal of Electronic Imaging 13(1), 146–165 (January 2004).