University of Passau
Faculty of Computer Science and Mathematics
Chair of Digital Image Processing

Master Thesis

Unsupervised Feature Extraction Using Deep Neural Networks

submitted by

Faizuddin Nasaruddin

1. Examiner:	Univ.-Prof. Dr. Tomas Sauer
2. Examiner:	Univ.-Prof. Dr. Michael Granitzer
Date:	April 19, 2021

Contents

Abstract

Feature extraction has played an important role in the field of machine learning and deep learning as it assists in eliminating redundant information and thus making the task at hand computationally efficient. Extracting salient and discriminant features is the ultimate goal of feature extraction techniques.

Customized algorithms can be defined to extract informative features for a particular dataset. The algorithms specific to data can be expensive and time consuming as it requires domain expertise. This leads to innovation in the field of unsupervised automated feature extraction where key features are extracted from any given data automatically.

In this thesis, an unsupervised feature extraction method based on stacked Independent Subspace Analysis (ISA) is implemented. This approach uses hierarchical networks consisting of two layers to learn complex features from the data.

The primary goal of this thesis is to extract rich features for the given data using the proposed stacked ISA network. The stacked Independent Subspace Analysis network integrated with Multi Atlas-based Image segmentation has been validated for segmentation of connecting rods (Pleuel) in a dataset consisting of four 3-Dimensional CT scans of an engine belonging to a Ford Fiesta car. The CT scans were measured on a XXL CT system developed at the Fraunhofer EZRT in Fürth.

The results of this study provide a deeper insight of the stacked ISA approach and its usefulness in the Image segmentation space. This framework achieves an IoU score of 0.763 and a dice score of 0.865 when considering information from a patch of size $21 \times 21 \times 3$ reduced to dimension of 40 for the given dataset. Further benefits and influence of training parameters are also presented.

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Dr. Tomas Sauer, Chair of Digital Image Processing at the University of Passau for giving me the opportunity to work under his expert guidance. His continuous support and insightful ideas have helped me explore the nits and grits of this topic.

I would like to thank the second examiner Prof. Dr. Michael Granitzer, Chair of Data Science at the University of Passau for his time and effort.

My sincere appreciation and gratitude goes to Mr. Thomas Lang for his continuous and valuable advice and all the constructive feedback on my thesis.

I am indebted to my parents Mr. Nasaruddin Tajuddin and Mrs. Wahida Nasar and my sweet sister Ms. Fariya for showering me with love and care and also supporting and believing in me in all my endeavors.

I would like to thank all my friends and loved ones in and out of Passau for always being there and providing me with support and constant motivation throughout my journey.

My grandparents and family, thank you for the support, love, and prayers.

I am very grateful to all the sources of inspiration that have always encouraged me to move forward no matter how hard things get.

I would also like to thank the Creator, the Almighty for creating this amazing world and all the things it has to offer.

Lastly, special thanks to the internet for offering a plethora of information and with such ease.

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Real-world data is often Large and Complex. Using this data directly in Machine learning/Deep learning tasks would require high computational power and time. Often, useful representations and features are extracted from the data which will have reduced dimensionality and can be used in deep learning tasks efficiently.

Feature extraction is a method for extracting key features or creating new ones from a dataset containing raw information with the goal of reducing the dimensionality of the data while preserving important information required for a better understanding of the data for a particular use case. Feature extraction is applicable for all data domains.

Innovation and advances in imaging infrastructure and technologies have paved the way for digital data such as images and videos to play an important role in our life. The key for a good analysis of the image is dependent on the features extracted.

The research to recognize various patterns in images that could benefit Image analysis is being actively conducted. Traditionally, handcrafted features were used that required human labor with expertise on that particular dataset. This was not only expensive but also time-consuming. Some of the other feature extraction techniques that are hand-crafted are independent of the dataset at hand. Thus, they might not be effective for every dataset.

Unsupervised Automated Feature extraction improves upon this standard workflow by extracting meaningful features from the data automatically. This way it can adapt to any data at hand. This technique removes the need for domain experts and reduces the time required for feature extraction. Also, it doesn't require annotated data.

One such Unsupervised Automated Feature extraction method has been implemented in this thesis based on the methods proposed by Shu Liao et al [Lia+13b]. In this method, a stacked Independent Subspace Analysis (ISA) algorithm is implemented to automatically learn features from 3D images. The features obtained can then be used in deep learning tasks. With the help of automated feature extraction, data scientists can now spend more time on other parts of the deep learning framework. To show the performance of this method, the learnt features obtained from 3-Dimensional CT scans of an engine are integrated into a Multi Atlas-based Segmentation framework.

The curiosity lies in the exploration in this research field to unravel solutions in detail.

1.2 Research Questions

The domain of automated feature extraction is vast and to understand it, a lot of research goes into it. Along with feature extraction, the combination of atlas-based segmentation should be compatible. Further to address this problem statement, exploration in the field leads to few questions. In this section, the questions are discussed.

- Can the stacked ISA network learn features in an unsupervised manner?
- Is the stacked ISA approach effective for image segmentation when integrated with the Multi Atlas based Segmentation (MAS) method?
- Does the stacked ISA approach perform better than a single ISA?
- How does the patch size affect the segmentation result of the ISA approach?
- How does the output dimension of ISA layers affect the performance?

The goal of this thesis is to perceive the concept behind the technicalities, interpret them into significant model architecture and adapt them to the novel dataset that will be used in this work. This work promotes learning in this field, to find quantifiable solutions at the end of experiments and evaluation.

1.3 Challenges

The field of automated feature engineering is desirable to overcome the struggles of manual feature extraction. These struggles encourage researchers on developing a novel method for feature extraction. To achieve this, initial research must be towards understanding the existing manual feature extraction and automated feature extraction methods. Since excellent feature extraction techniques require a lot of expert knowledge, building an automatic feature extraction technique poses a challenge.

In order to visualize and annotate the high energy CT scan of the engine containing connecting rods (conrods), clearing out the unwanted information from the scan requires study about the physical features of the object itself. The dataset contains noise and beam hardening artefacts which degrade the quality of features obtained in the feature extraction technique. This has to be taken into account of and necessary remedies are to be performed.

The ubiquitous challenge of limited samples and class imbalance has to be considered and intelligently work with the features that are available. The ISA approach for Image segmentation is mainly used in the medical domain. Hence, an extensive literature survey to adapt the techniques to this dataset is required. Understanding the technicalities of the 3D slicer tool is needed to perform tasks like annotation and image registration.

There are no existing implementations for Multi Atlas segmentation using Stacked ISA, thus the entire framework was built from scratch which required understanding of several concepts related to the field. After building the model, some tuning will be required to obtain the best parameters, which in turn requires understanding the nuances of the model.

During extraction of feature signature for an entire image, patches have to be extracted for each voxel present in the image, the memory requirements for such data can go upwards

of 23GB in size for the dataset at hand. Also, performing operations on such data can be time-consuming. Optimizing the code is necessary to handle such data.

1.4 Outline

This thesis will proceed as follows:

Chapter 2: This chapter gives an overview of the fundamentals of deep learning, Image segmentation, and feature extraction followed by a comprehensive review of related works in the field.

Chapter 3: In this chapter, the stacked ISA approach is introduced. The implementation process is explained following which it is integrated into a Segmentation framework.

Chapter 4: Dataset information, metrics used for evaluation, and tools used are discussed in this chapter. The annotation procedure is also briefly described.

Chapter 5: This chapter entails numerical experiments. The results are presented and insight is drawn from them.

Chapter 6: In the final chapter, conclusions on the work are summarised and the future tasks are discussed.

2 Theoretical background

In this chapter, the background study on which the thesis is built upon on is included. This includes the fundamentals of Deep Learning, Image Segmentation and Feature engineering. A comprehensive review of related work on the thesis is also discussed.

2.1 Deep Learning

The human visual system has been tuned via evolution, which is now a major gateway to perception. In the human brain, there are physical vortexes that are accountable for the human vision [Nie]. The human brain is a supercomputer that is capable of complex image processing [Nie]. Computer vision is a very versatile field in science which enables computers to gain high-level knowledge [Lam]. It strives to achieve the perspective of a human visual system. This has led to research in the field of artificial intelligence.

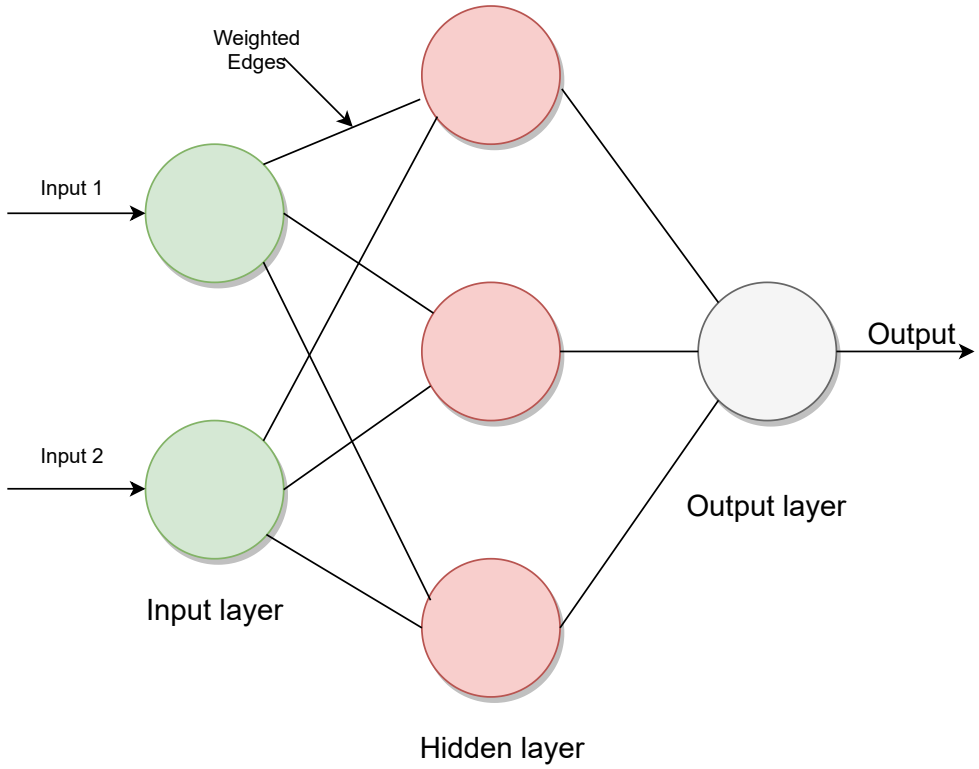


Figure 2.1: Basic neural network architecture.

The study of the human brain and its working has helped conceptualize the neurons in the brain into a neural network. This forms the basis of the majority of deep learning algorithms. These are also called Artificial Neural Networks (ANN). The basic neural network is as illustrated in Figure 2.1 and comprises units representing the neurons of a brain with an activation function and parameters.

This network consists of hidden layers which compute the relationship between the data. Figure 2.2 demonstrates what a simple unit looks like. Neural networks are capable of handling an enormous amount of data. With time, the ANN has advanced and is now being employed in autonomous cars, medical imaging, voice, text, face recognition, etc.

There are many challenges involved, especially when there is so much uncertainty with the human visual system itself [Bro]. Computers work well with a certain set of rules, for them to perform well in the open world, new techniques should evolve.

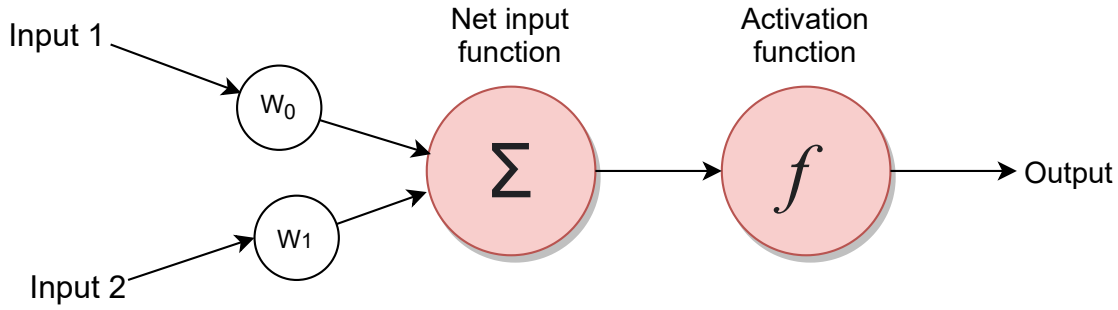


Figure 2.2: Neuron simple unit.

Deep learning was introduced by Walter Pitts and Warren McCulloch in the year 1943. Prior to this, there were major advancements called cybernetics and connectionism which were unpopular. Deep learning provided a breakthrough that tried and eliminated precious shortcomings [ban].

Advanced deep learning techniques are advancing due to their eminent benefits. Deep learning in computer vision was promoted through a highly promising convolutional neural network or CNN for various problems such as object detection, classification, and segmentation. Like ANN, CNN is also a brainchild of science and engineering. This too was inspired by the human visual system [Lin20]. Hubel and Weisel discovered that the animal brain has different types of cells which are responsible for perception. The simple cells are stimulated only when kept at certain locations, they are responsible for light and dark changes. Whereas, complex cells are less strict and respond effectively [Lin20]. The reception to the complex less is from the simple cells. From Figure 2.3, the structure of simple and complex cells can be observed.

Later Neocognitron was developed by Fukushima, which served as an inspiration for CNN. Figure 2.3 shows the comparison between that of a typical CNN. It can be observed on the left side that, since complex cells receive the information from simple cells they have more unvaried responses. On the right side of the image, it can be seen that the convolution layer is applied to the input image. Along with this, a filter is applied, a grayscale box which helps

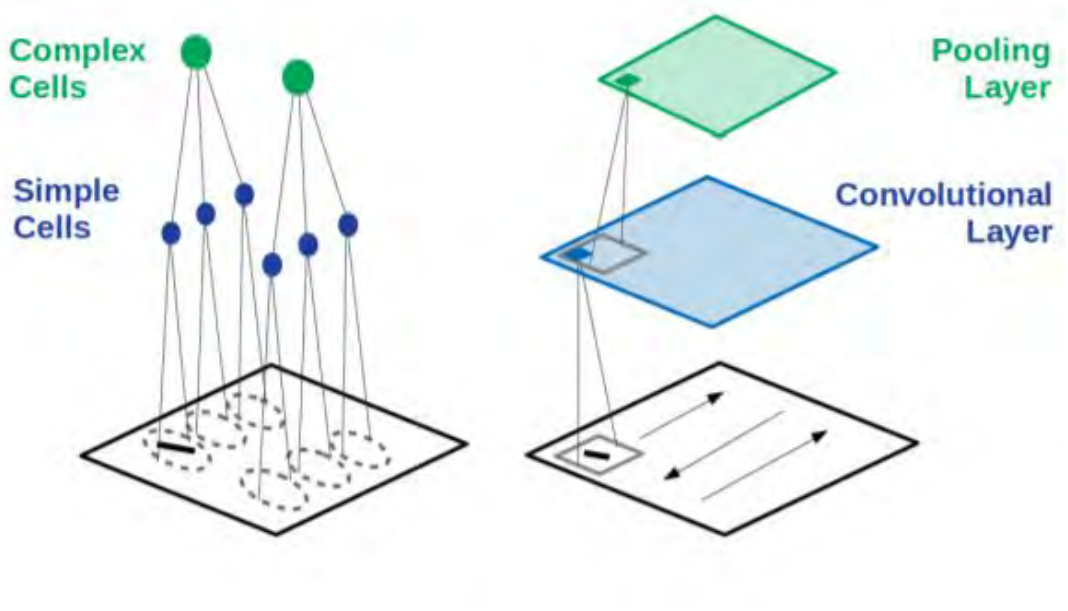


Figure 2.3: Comparision between simple cell present in the eyes with CNN layers [Lin20].

in learning the features. These are called feature maps. The pooling layer helps in selecting the feature from the map. There are different strategies involved. Selecting the maximum value from the feature map mimics the complex cells. This operation is generally known as *"max-pooling"*. The layers of CNN are explained in detail below.

A typical convolutional layer consists of multiple hidden layers, feature detection layers, classification parts, etc. As illustrated in the Figure 2.4 , the layered architecture can be observed. In the lower layers, the network learns primitive features such as edges etc, later on, it learns more prominent and distinguishing features appropriate to the classes. The major building blocks of CNN are :

- Convolutional layers.
- Pooling layers.
- Fully-connected layers.

As discussed convolutional layers contain filters called feature maps. This layer is the fundamental part of the whole network. The first convolutional layer extracts pixels values from the input patch which serves as feature maps to the upcoming layers. The area covered between movement of the filter at a time, on the input image is referred to as strides. The overlapping application is repeated several times with the help of filters across the whole image.

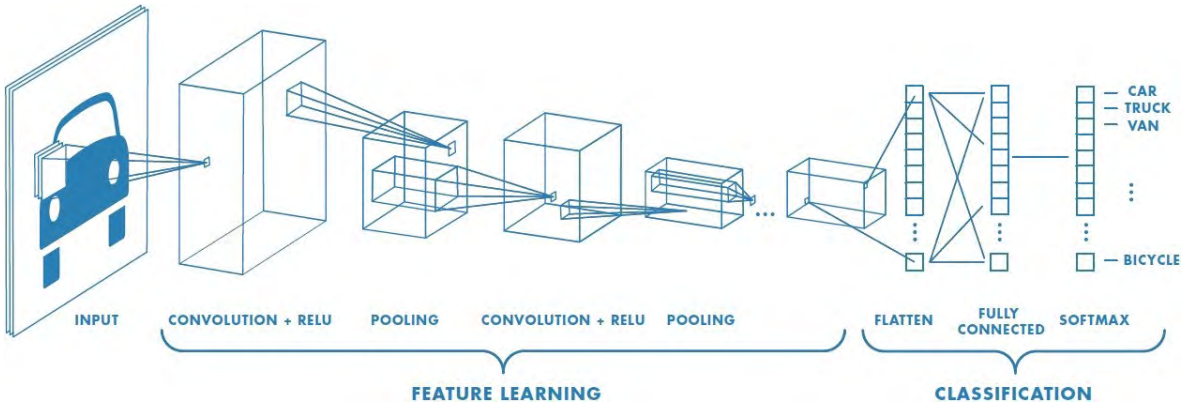


Figure 2.4: Typical structure of Convolutional neural network [Gup].

The pooling layers consolidate the features expressed from the previous layers. In general, the output features are susceptible to their location in the input. The pooling layer aims to make them robust by downsampling the features. By doing this any translational changes that occur to the features will be recorded, and important structural units will not get lost.

This is typically found after the convolution layer. There is an activation layer after the convolution layer which is responsible for activating the neurons. Any network without this would be a linear regression model. To assist the model to learn more complex features and introduce non-linearity, the activation functions are used.

There are many activation functions, such as sigmoid[1] which is an 'S' shaped graph. It is used mostly in binary classification problems, as it activates any neuron greater than "0.5" to "1" otherwise "0" [Gee]. There is a derived version of this known as the TanH[2] function which ranges from -1 to 1. It is better in centering the data, and can not solve the vanishing gradient[3] problem [Gee]. The popularly used activation function is ReLU[4], it is faster and computationally less expensive [Gee]. It helps the model to converge faster, and also helps with the problem of vanishing gradients as it does not produce small gradients.

Fully connected layers are used as a layer to learn the high-level features non-linearly. This acts as a gateway for the network to make predictions. Sometimes a softmax layer[5] is used at the end to predict the probabilities of each output.

2.2 Image Segmentation

Image segmentation is a process where the image pixels are partitioned into meaningful parts so as to achieve a better understanding of the image. It uses information such as intensity, texture, continuity of intensities and high level patterns to partition the image.

[1] https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function
[2] https://keras.io/api/layers/activations/
[3] https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484
[4] https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning
[5] https://keras.io/api/layers/activation_layers/softmax/

Image segmentation has an important role in medical image analysis. It is an essential and indispensable process in anatomy related studies. Analysis of brain scans, lung scans, prostate scans are few of the examples where image segmentation is needed.

Imaging techniques such as Computed tomography (CT) scans, Positron emission tomography (PET), Magnetic Resonance Imaging (MRI), etc help in diagnosing diseases. They are a part of routine diagnostics performed in health care facilities. These techniques help in visualizing the internal organs and structures, thus assessing for any abnormalities. As per clinical practice, an expert radiologist studies the scans of each patient and segments the affected regions manually. It is not only time consuming but also is highly dependent on the radiologist. Thus, it is subjected to high intra and inter rater variability [IDŞ16]. Also, only qualitative assessment can be done on manual segmentation.

Quantitative assessment would provide valuable information, however due to large variability in size, shape and location of subject in a scan, the assessment proved to be difficult. Thus, manual segmentation is not feasible for large studies. Many researchers are now focused on applying computer algorithms to automatically segment images. These involve Machine learning and Deep learning methods.

Machine learning approaches are heavily dependent on pre-processing of data whereas Deep learning methods are dependent on large amounts of data. Convolutional Neural Networks (CNNs) are the most popular in the field of Image segmentation. Review on several deep learning approaches are covered by Alom et al [Alo+19]. To achieve automated image segmentation, U-Net, atlas-based segmentation, Region-Based Convolutional Neural Network (RCNN) and several methods have been developed.

The recent resurgence of CNN is in the field of Segmentation, object detection, and classification. Segmentation requires more accurate supervision than any other computer vision task. This is an extension of object detection, which not only classifies the pixels but will identify the shape using image localization.

A fully convolutional network is known for its state-of-the-art results. It contains convolutional layers which use filters to produce feature maps. It allows skip connection which helps the flow of feature maps from lower to higher-level layers. This results in very precise segmentation [Nup].

Encoder-decoder-based segmentation consists of an encoder path that is made up of convolutional layers and the decoder path is made up of deconvolutional layers. The input feature vector plays an important part as it is responsible for output probabilities [Nup]. The most popular encoder-decoder architectures are U-Net and V-Net. These involve data augmentation techniques which help the learning from manual labels [Nup]. U-Net is a symmetric architecture that consists of an analysis and synthesis path for explicitly capturing *"what"* and *"where"* information. The V-Net is based on the Dice coefficient which can perform a whole MRI segmentation at a time.

Pyramid network-based models such as the Feature pyramid network (FPN) uses top-down and bottom-up pathways to create pyramids of features. Each level of the top-down pathway produces a prediction and these are merged using lateral connections [Nup].

Other than CNNs, RNNs i.e, Recurrent Neural Networks have performed well in image segmentation problems. It models short-term and long-term dependencies for better estimation.

Models such as ReSeg even use upsampling layers to maintain image resolution throughout the network and also Gated Recurrent Units (GRUs) to balance memory and computational power. A graph-based model Long short term memory (LSTM) creates an undirected graph by taking a superpixel [Nup].

The traditional manual segmentation techniques are outmoded because of time constraints and need for expert supervision [Ala+21]. Hence, the CNN, FCN for these computer vision tasks have inflated. Sometimes it is difficult to find fine segmentation experts in some domain which calls out for a better way. This is where automatic segmentation comes into the picture. There is a need for reducing effort, time and to help the user for easy detection in any field this is applied to. The automatic segmentation is not sensitive to user error or bias, thus making it a better choice.

This work deals with semantic segmentation, inspired from [Lia+13b]. It proposes an automatic image segmentation model in the 3D domain. The study on three generations of Automatic segmentation methods in [SA10] have classified the techniques to :

- Gray level features - the techniques which use gray level features come into this category. There are many methods based on this such as amplitude, edge, and region-based segmentation. These are controlled by the act of thresholding. Segmentation based on amplitude may be one of the simpler ways, but these fall behind as it is difficult to find a uniform thresholding value, sometimes getting affected by the unwanted noise. Edge-based on the other hand uses the variance in the grey values and marks boundaries to separate the pixels belonging to different classes. As the name suggests it detects edges, but there might be some frail edges, some posing as edges which might negatively affect the performance [SA10].
- Textual features - Based on the texture of different regions in the image, segmentation is done. The texture might be represented as smooth, grained depending on the pixel intensities, a sense of commonness among the pixels [SA10]. The texture extraction process can be done synthetically as well as statistically [SA10].
- Model-based Segmentation- It will make use of the patterns present in any structure at hand, and extract probabilistic features for the structure. These require manual attention to initially set the model parameters. These are susceptible to deformed models which can affect the generalization of the model.
- Atlas-based segmentation - This third-generation algorithm is highly robust, especially in the field of medical image segmentation. It is a guided approach that works on correlation. There will be a reference image that acts as the model image. Other input images will have to trace along with the model image. First, a procedure called image registration is done which aligns the spatial coordinates of the input image. Post this, the label information is read from the atlas to the image [SA10]. The two categories under Atlas-based segmentation are discussed in the upcoming section.

2.3 Atlas-based Segmentation

The two categories that fall under atlas based segmentation are discussed in the following subsections.

2.3.1 Single Atlas-based Segmentation

Approaches based on single atlas-based segmentation (SAS) involve one training image with manually labelled label map and registration algorithm that transforms the training image to target image i.e. the image to be segmented with respect to spatial correspondence, and the target image is segmented by propagating the training image label to the target image coordinates.

This however limits the performance for cases where there is large structural, anatomical or intensity difference between the atlas image and target image due to achieving poor registration. In SAS, if there are a collection of training images, the best training atlas is chosen based on the similarity between the registered training image and the target image intensities. The drawback of using this method is that it does not consider the useful information present in other training images. To overcome this problem, Multi Atlas-based Segmentation (MAS) methods have been proposed[RRM04][Alj+09].

2.3.2 Multi atlas-based segmentation

Similar to Single Atlas-based Segmentation, MAS comprises image registration, in addition, MAS has a label fusion method. The atlas images can be termed as training images and is used interchangeably. In the MAS framework, each training image is registered to the target image and training labels are then propagated to the target image which are then fused using label fusion methods to get the final segmentation. In few of the MAS methods, a collection of training images that have better similarity with the target image are selected instead of using all available training images [Zaf+18].

In certain use cases where the training images are not a representative sample of the population of test images, better training images can be synthesized that offer much closer representation of the test images like in [JYS12]. This technique helps to increase the accuracy by enriching the training images pool.

A different approach where one could obtain a large number of annotated atlases from non expert segmenters could work in some applications [IS15]. The computational power requirement for the MAS is linear with respect to the number of training images used. If the number of training images selected is halved, the speed of the MAS algorithm is expected to be doubled.

Label fusion is the main component of MAS as it combines the propagated training image labels to produce the segmentation result of the target image. Most common fusion techniques are Majority Voting and Weighted Voting.

In Majority Voting, the most occurring label from the corresponding atlas locations is chosen for each target location, thus utilizing information from all atlases. The drawback of Majority voting is that it does not consider intensity values of image voxels.

Majority Voting is extended to weighted voting where the training atlases are assigned weights based on their similarity with the target image. The weights used in weighted voting represent the entire atlas, therefore they cannot represent the varying spatial nature within the same atlas. To tackle this and use local information nearby the target voxels, patch based label fusion is introduced.

In recent years, several patch based methods have been proposed [Cou+11][San+15]. This method involves calculating the similarity of the patches in the corresponding locations and assigning weights to each of the voxels based on the similarity measure. Techniques like sparse coding, euclidean distance measures are used to calculate the similarity.

Sparse representation is a common tool used in applications like facial recognition, Image classification and image reconstruction. The idea is that the feature signature of the target voxel can be represented as a sparse combination of training image voxels. These sparse coefficients can then be used as the weights for the corresponding voxel in the label fusion method. The power of sparse representation is dependent on developing highly discriminated feature vectors.

2.4 Feature engineering

Data that will be in crude forms contain measurable attributes called features. These must be processed to calibrate the algorithm. Feature engineering has a strong influence on the prediction of the Classification and segmentation models. Unlike humans, the computer needs a set of instructions in the form of vital features that assists in learning. Some of the datasets are rarely available and expensive at times such as medical data. Feature engineering can utilize the dataset at hand and make sure that no important features are lost. On the other side, it can eliminate redundant features and further foster effective training.

Feature extraction is a dimensionality reduction process that will make the task at hand easier. They reduce the huge number of variables by selecting the features of utmost importance. This reduces the computational power and still retains the originality. This decreases the model's effort, thereby accelerating training.

Handcrafted feature methods are designed to extract features from data. These methods use standard algorithms which are defined by an expert in the field. Few of the handcrafted feature methods have been discussed in the upcoming section. These methods are not only time-consuming but also expensive as they require expert knowledge in the domain of the dataset. These customized methods are specific to a dataset and cannot be applied to data from different domains. To tackle this, automated feature extraction methods were developed.

Automated feature extraction is an unsupervised method that automatically learns representations from images on its own without the need for any additional data. This technique has become a novel approach in many computer vision tasks. It does not involve human supervision making it less error prone. This can be applied on all datasets and is also less time-consuming.

In this work, an unsupervised feature extraction method is implemented which is built upon Independent Subspace Analysis (ISA). The ISA is a feature extraction method on its own. It is a two-layer neural network that captures invariant features between input patches that it is trained on. The layered network helps to capture complex features from images.

The ISA network is an extended version of Independent Component Analysis (ICA). In ICA, each signal is decomposed into basis vectors that are independent of each other. ICA can be extended to ISA by considering multidimensional components where the components within each group are not necessarily independent [The06]. So, an ISA with group size 1

is equivalent to ICA since it has no dependent components within a group. ICA is often combined with Principal Component Analysis (PCA) for dimensionality reduction. In ISA, depending on the size of the subspace, the components within the same group are dependent. But the components in different subspaces must be independent. More on ISA is discussed in Chapter 3.

The Stacked ISA approach utilizes Principal Component Analysis for handling larger image patches. PCA is a dimensionality reduction technique that is used to represent features with higher dimensions into a small set of variables without losing much information. The newly created set is called principal components. In PCA, linear combinations of original feature vectors are made to achieve maximum variance with each linear combination being uncorrelated with each other. The first few components consist of most information. For example, if 30-dimensional data is reduced to 10 dimensions using PCA. Then, the first component of PCA consists of majority information, followed by the second component which consists of the remaining information, and so on.

The implementation of PCA involves calculating the covariance matrix of the original vector and then calculating the eigenvector and eigenvalues of this covariance matrix. The eigenvector with the highest corresponding eigenvalue is the principal component for the data. Depending on the number of dimensions required in the output features, the eigenvalues can be chosen. The original data is then transformed to reduced dimensions using the values obtained [JC16].

2.5 Related Work

Data in the world is in abundance. The processing of this data requires a lot of techniques. To extract meaningful information from the data is the ultimate goal. In deep learning or machine learning models, for the model to learn the behavior of the data, a well-defined dataset is required.

The key to success in any real-life algorithm is the way in which the data is represented. Value-added data is much appreciated as it aids the algorithm to perform well. To achieve this, the requirement of hand-crafted features are necessary. In the past few decades, research in feature engineering has been agile. As a result, many techniques will be discussed in this section.

Literature research in handcrafted features leads to [Low04], where the authors present a robust, distinctive feature extraction that can handle distortion, change in 3D viewpoint. The approach is named as Scale Invariant Feature Transform (SIFT), as the coordinates of the features are scale invariant. It provides dense, stable features from the provided image data. However, the process is quite slow and does not work well in low-powered devices.

Histograms of Oriented Gradients (HOG) were studied in [DT05], this technique is widely used for object recognition. It is well known for detecting humans in an image. In HOG, the image is divided into bins and for the pixels that come under these bins, an individual histogram is built. These histograms are concatenated to form the feature descriptor. HOG is great for detecting corners and edges in an image, however when it comes to textural classification, HOG is not considered a good choice. Also, HOG is highly sensitive to rotation.

In the field of computer vision, it is well acknowledged that multiresolution decomposition performs a great deal. Along the same lines, a wavelet theory is explored in [Mal89]. This can be employed in various computer vision tasks such as detection, signal coding, etc. They provide a condensed set of features from the novel image. The computational time is faster, as they use integral images that are in the form of look-up tables. It also ensures to remove redundancy as a compact feature set is produced, which helps eliminate repetitive features.

Authors of [Hec+06] have proposed a layered architecture that consists of a 3 layered architecture that extracts discriminative features that are used in protein cellular classification. This is a supervised technique and based on Local Binary Pattern (LBP) method. The original LBP tends to provide uneven distributions as all patterns do not have an equal part in the image. Many histogram distributions contain only the most dominant features and very less information is available about the less occurring features. So layered architecture was designed to eliminate these limitations. This approach provides robust, highly discriminative features which are precise in nature.

The feature extraction methods discussed in this section until now are hand-crafted features. Their effectiveness is dependent on the dataset they are applied to. They are unable to extract important features from all types of data i.e., they are suited for specific applications. Hand crafted features are extracted by programmed algorithms defined by experts.

With the increase in image analysis in the medical domain, more and more studies are being conducted to obtain better results in the task at hand. The significant variation in performance is observed for different feature extraction techniques which suggests the necessity of better feature extraction methods. This has led to increase in research in the field of feature engineering.

Stacked Independent subspace analysis network has been used in [Le+11] to extract invariant spatio-temporal features to assist in an action recognition classification problem. In [Ram+13], the Authors have integrated stacked ISA with semantic rules to perform action recognition between two complex scenarios. Qi dou et al have proposed cerebral microbleed detection method in Magnetic Resonance Images (MRI) using Stacked ISA along with SVM (Support Vector Machine) classifier[Dou+15]. This model has achieved high value of true positives and low values of false negative sensitivity and also has very few false alarms.

To evaluate the feature extraction techniques in an image segmentation framework, the most popular approach used is Multi Atlas based Segmentation (MAS) using Label propagation. Several studies[RRM04][Hec+06][Alj+09][AMO09] have been conducted on the effectiveness of this approach for Image segmentation.

In [RHS11], authors have proposed a Human brain segmentation method which is based on a label propagation framework that uses intensity similarities between atlas and the target images.

Liao et al have proposed a prostate segmentation method where representations are learned from the dataset using Subclass Discriminant Analysis (SDA). These features are then used in the Label propagation framework for image segmentation [Lia+13a].

A similar approach where Local Binary Texture(LBT) is used in MAS framework is proposed in [Zha+12]. The authors have used the features extracted to sparsely represent the target

voxel using a patch from the training images. The effectiveness in obtaining a good sparse representation resides on how discriminate the features are.

Feature extraction forms the core in such approaches. Therefore, Feature engineering has been an active research topic in the field of deep learning.

3 Methods

The workflow of Stacked ISA based image segmentation framework is illustrated in Figure 3.1. It is divided in to two parts. The training stage involves training the stacked ISA network using 3D images. In the testing stage, feature maps are created using the trained stacked ISA network which is then incorporated in the Atlas-based segmentation framework to perform image segmentation.

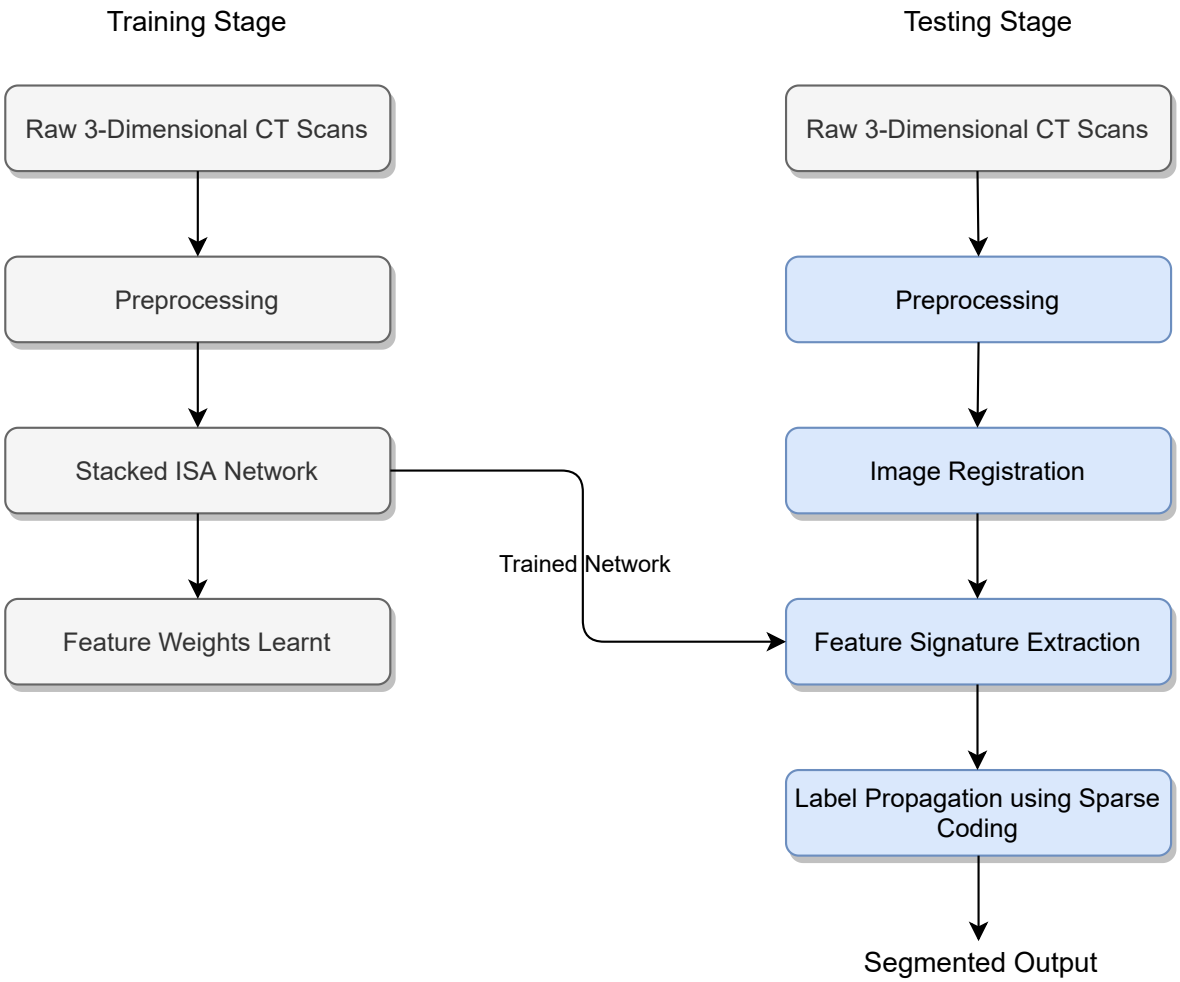


Figure 3.1: Workflow of Stacked ISA based image segmentation framework

3.1 Training

In this section, the stages of training that include Data preparation and Feature Extraction will be described in detail.

3.1.1 Data preparation

The 3D input images containing CT scans of con rods have values ranging between 0 to 50000 and are in raw format. These images were converted to nifti[1] format using the software ITK-SNAP[2] for easier access in the 3D slicer tool[3]. Once the image is loaded in the 3D slicer tool, it is rescaled between values 0 and 255 using the `RescaleIntensityImageFilter`[4] option present in the Simple Filters module of the 3D slicer tool.

This rescaled image is now loaded into Google Colab notebook[5]. Thresholding is performed on the rescaled image to reduce the CT artifacts like noise and beam hardening. Figure 3.2 depicts the raw image and the thresholded rescaled image. The image is then normalized to values between 0 and 1. Random sampling is performed with respect to three axes. The patch size of the 3D image and the number of samples per image are defined. The patches are stored as column vectors. Two sets of sampling are performed on the dataset. Each with a different patch size.

Further explanation on this will be discussed in the upcoming sections. Further pre-processing is performed on the smaller image patches. DC components are removed from the image patches after which it is whitened. For the whitening of the data, the PCA algorithm is used. The input of shape $f \times p$ is then fed to the ISA algorithm where p is the number of patches sampled and f is the patch area.

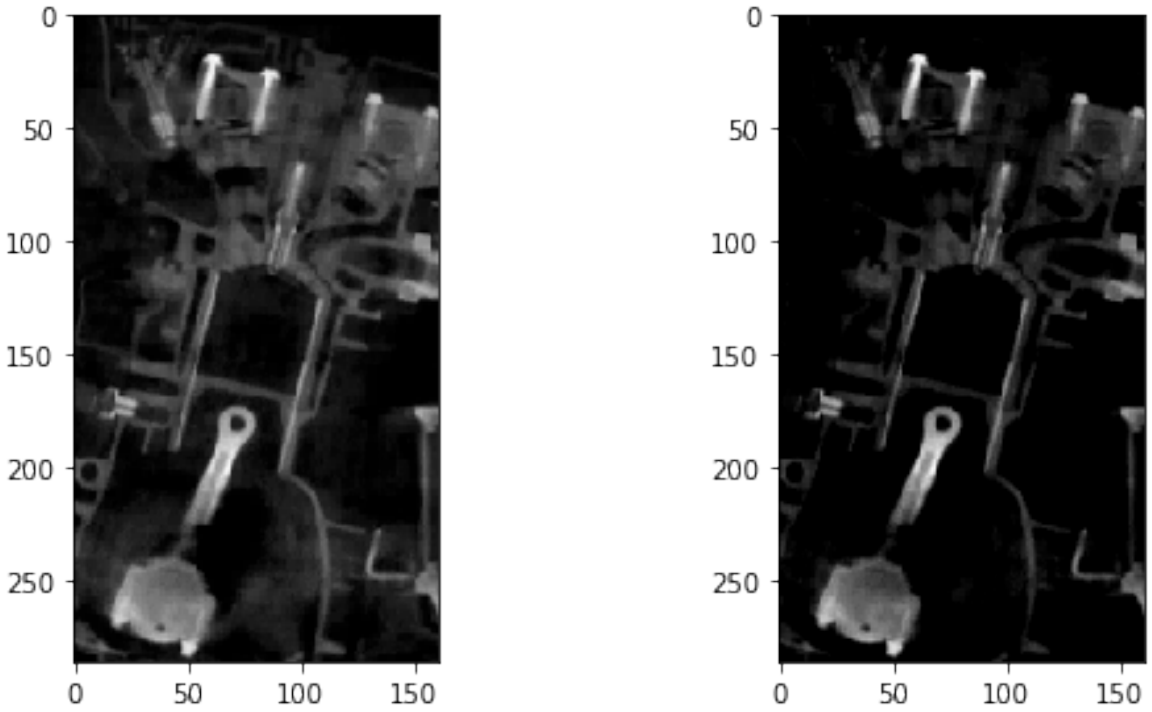
3.1.2 Feature Extraction

Before starting with stacked ISA, the basic Independent Subspace Analysis network which is often used in feature extraction for images is discussed. Then the stacked ISA network obtained through stacking and convolution is explained which is needed to scale the basic ISA network for larger images and learn high-level features.

3.1.2.1 Basic Independent Subspace Analysis Network

The basic ISA network consists of two layers. The first layer captures square non-linearity relationships among image patches. The units of the first layer are termed simple units. The second layer consists of pooling units that group the response from different simple units and capture square root non-linearity relationships. The basic network structure is illustrated in

[1] `https://nifti.nimh.nih.gov/`
[2] `http://www.itksnap.org/pmwiki/pmwiki.php`
[3] `https://www.slicer.org/`
[4] `https://www.slicer.org/wiki/Documentation/Nightly/Modules/SimpleFilters/`
[5] `https://colab.research.google.com/notebooks/intro.ipynb`



(a) Original Image (b) Pre-processed Image

Figure 3.2: A Slice of a 3D CT Scan

Figure 3.3.

The weights W corresponding to simple units in the first layer of the ISA network are learned. The weights V corresponding to pooling units in the second layer of the basic ISA network are fixed. Given an input pattern x^t, the goal of ISA is to estimate the filter matrix W and V by minimizing the energy function.

$$\begin{aligned} &\underset{W}{\text{minimize}} && \sum_{t=1}^{T} \sum_{i=1}^{m} p_i \left(x^t; W, V\right), \\ &\text{subject to } WW^T = \mathbf{I} \end{aligned} \tag{3.1}$$

$$\text{where } p_i \left(x^t; W, V\right) = \sqrt{\sum_{k=1}^{m} V_{ik} \left(\sum_{j=1}^{n} W_{kj} x_j^t\right)^2} \tag{3.2}$$

is the activation of each second layer[Le+11]. n, k, m are the input dimension, the number of simple units, and pooling units respectively. The general rule of ISA is that filters belonging to different subspaces should be independent whereas the filters within the same subspace

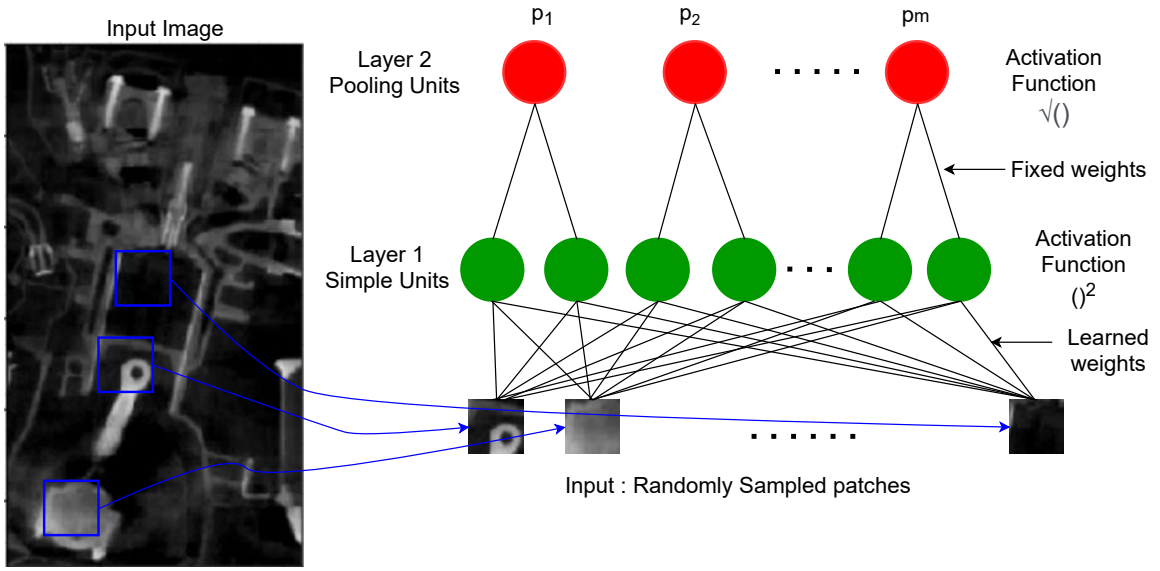


Figure 3.3: Basic ISA network structure [Le+11]

are not necessarily independent. With the help of the ISA algorithm, invariant features can be extracted by inspecting the underlying subspace structure of the input data. ISA is an unsupervised feature learning method as it learns features from unlabeled data[Le+11].

The ISA network is built like a two-layer neural network as described in Figure 3.3. The activation function of the first layer is the square function and for the second layer, the square root function is used. The reasons for such activation is due to the fact that the sum of squares is used to represent the strength of the subspace. Since the weight vectors are considered to be orthogonal, the sum of squares at the pooling layer represents the norm of the vector and the norm is considered as a measure for strength. Also, it is a good representation of complex cells in a visual cortex where they are often modeled by the classical 'energy model' i.e., the sum of squares[HHH09]. At the end of each layer, post-activation processing is performed by a low-high threshold. In this, a boolean function is used to limit the values between predefined high threshold and low threshold.

The pre-processed input patches obtained from the pre-processing stage are now fed to the network. The input patches obtained from the data preparation stage are now pre-processed. First, DC components are removed from the image patches. Then, it is whitened. For the whitening of the data, the PCA algorithm is used. This pre-processed data of size $f \times p$ where p is the number of patches and f is the patch area, is now ready to be fed to the ISA network. The subspace size s and the number of linear components n required to represent the feature vectors are defined.

A matrix W of size $n \times f$ is initialized with random initial values. This matrix corresponds to the weights of the first layer (simple units) in the ISA network. A subspace matrix is created where the i, j^{th} element is assigned 1 if it is present in the same subspace and zero if it falls in a different subspace. This matrix represents the second layer (Pooling units) of the ISA network where the weights are fixed. The matrix W is updated in each iteration based on the objective function given in Equation 3.1. The weight matrix when reshaped to the resolution

of input patches represents the filters learned from the input data. These filters are then used to extract features. Similar filters are grouped to span a subspace. Filters belonging to the same subspace can be dependent but filters of different subspace must be independent. The algorithm is run for a fixed number of iterations.

3.1.2.2 Stacked Independent Subspace Analysis (ISA) Network

The basic ISA network becomes less efficient and computationally expensive when input patches are large. When training the network with high dimensional data like 3D Images or videos, the time taken will be high. To overcome this disadvantage and to learn more complex features, the stacked ISA network was designed. It consists of hierarchical architecture as illustrated in Figure 3.4 and uses PCA and ISA as sub-units for unsupervised feature extraction. The PCA operation is used to whiten the data and reduce the dimension so as to provide a more compact representation of data. This makes sure that the ISA algorithm works with low dimensional inputs. In the low-level layer, basic image features like spots, edges are learned from input data. The high-level layer encodes more abstract higher-level image information [Lia+13b].

The implementation strategy is as follows. First, the ISA algorithm is trained on small input patches. This trained network is convolved with a larger region of image i.e., on the larger input patches. The output of the trained networks is concatenated and given as input to the next layer which is another ISA algorithm. The input to the second ISA algorithm is preprocessed by removing DC components and performing whitening of data using PCA. This reduces the dimension of the input data and makes sure that low dimensional data is passed to the second ISA algorithm. Thus, scaling the network for large input patches. The second layer is then trained for a fixed number of iterations.

The stacked network is trained in a greedy layerwise manner as done in [Ben+07]. Here, layer 1 is trained and then without affecting layer 1, layer 2 is trained. The number of iterations of the algorithm is defined. The network along with the filters for both the layers is stored in order to compute the feature maps of the input data required for segmentation/classification problems. The architecture of the stacked ISA network for 3D Image is illustrated in Figure 3.4.

The Figure 3.5 demonstrates a few of the filters learned from the first layer of the stacked ISA network. The dataset consists of 4 grayscale 3D Images and the number of patches per image is 2000, totaling 8000 image patches. The window size of each patch is $17 \times 17 \times 2$ for the first layer with output dimension of 60 and $21 \times 21 \times 3$ for the second layer of the stacked ISA network with output dimension of 40. It was observed that the change in objective function above 200 iterations was insignificant for the parameters discussed. Thus, both the layers were trained for 300 iterations.

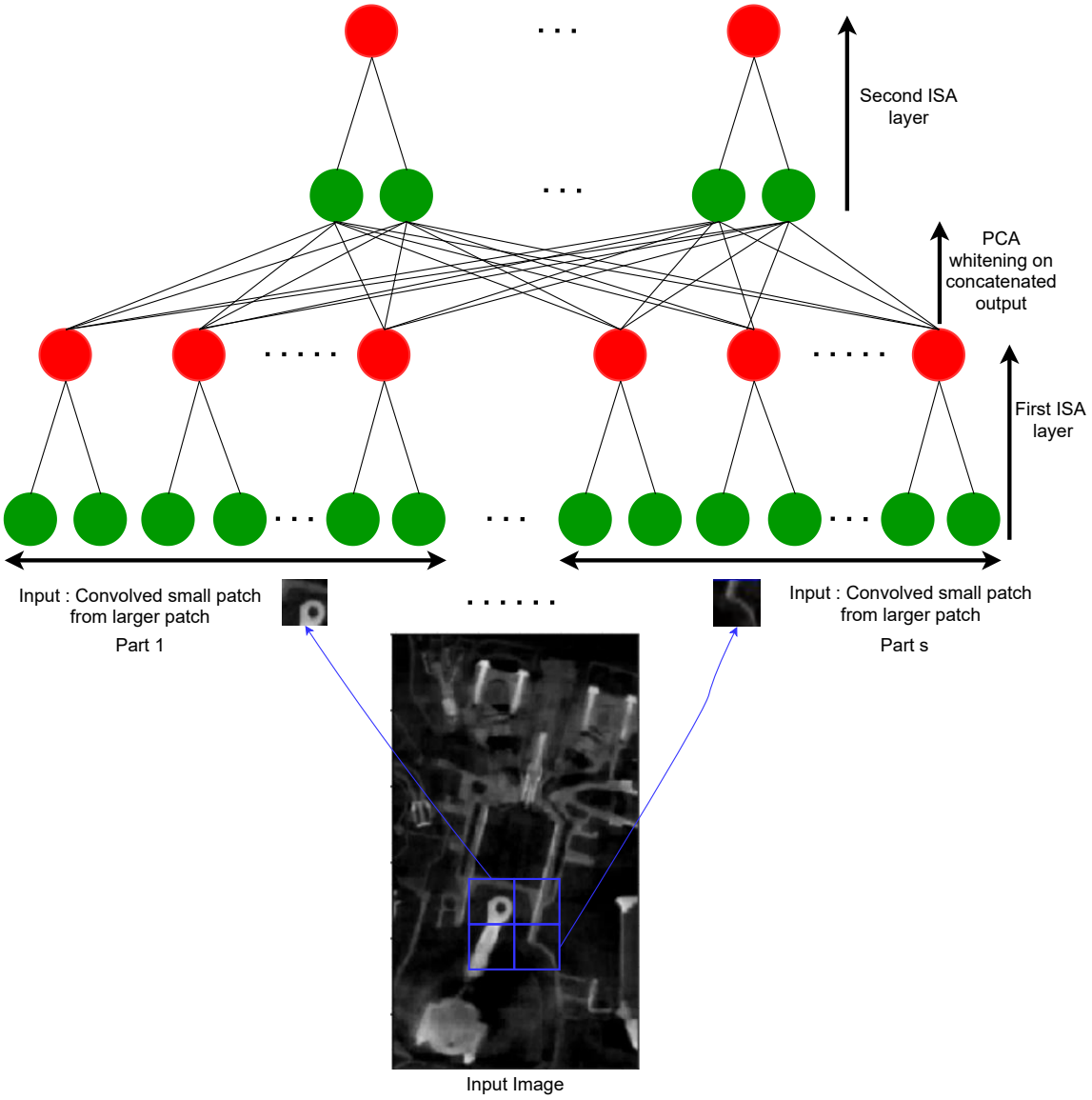


Figure 3.4: Stacked ISA network structure with large input patch decomposed to s smaller overlapping patches [Lia+13b] [Le+11]

3.2 Testing

The procedures followed in the testing phase i.e., Data preparation, Image registration, Feature Signature extraction and Image segmentation are described in this section.

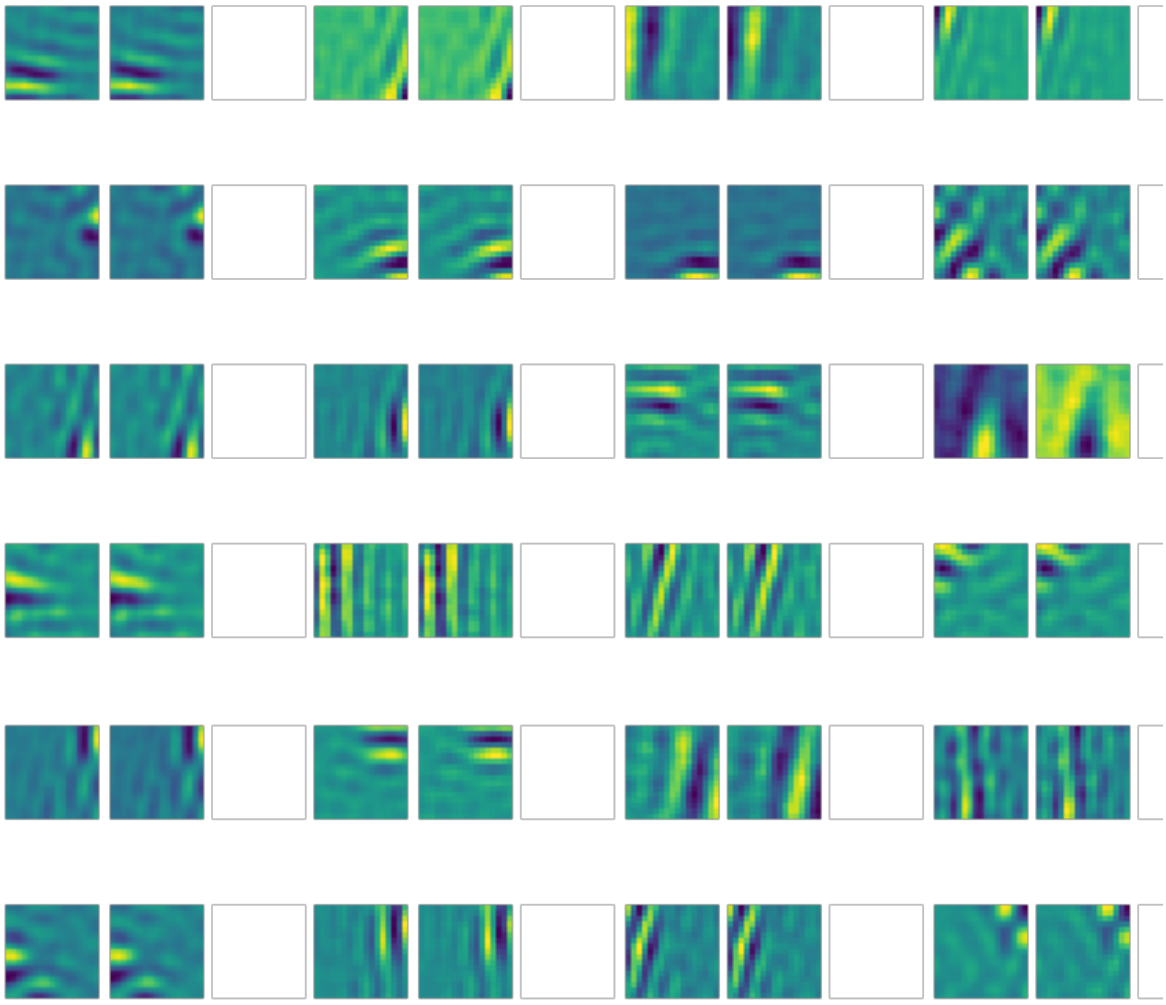


Figure 3.5: Few of the filters learned from the first layer of the stacked ISA network.

3.2.1 Data preparation

The 3D input images containing CT scans of connecting rods (conrods) in nifti format are loaded into the 3D slicer tool [6]. The images are rescaled between values 0 and 255 using the `RescaleIntensityImageFilter` option present in the Simple Filters module of the 3D slicer tool. The rescaled images are now ready for image registration.

3.2.2 Image registration

This technique is commonly used in image processing to align different images to a target image. The aim is to integrate the data so that it becomes comparable [Bro92]. Image registration can be classified into two types based on the transformation used. Transformations like translation, rotation, skewing, and scaling are performed in Rigid registration whereas

[6] `https://www.slicer.org/`

non-linear and elastic transformation is performed in non-rigid registration. The need for image registration as a key pre-processing step of the input data is to align images so as to ease the Atlas-based label propagation method. In this thesis, Rigid registration is performed on the 3D input images.

The image for which segmentation needs to be performed is considered as the target image I_{new}. The other images that will be used as input in the Label propagation framework are considered as training Images $I_t(t = 1, ..., T)$ where T is the total number of training images available. These training images can also be called atlases. Initially, all the image volumes are centered around the origin. Next, manual registration is performed using the transformation module of the 3D slicer tool on the training images keeping the target image as a reference. Following this, affine registration is performed using General Registration(BRAINS) module[7]. The registration procedures will try to maximize the similarity between the target image and the training images. Both the transforms from manual registration and affine registration are saved and applied to their corresponding label maps $L_t(t = 1, ..., T)$. The transformations are applied on the label maps using the Resample Image module[8] of the 3D slicer tool.

3.2.3 Feature Signature Extraction

The registered images, target image, and its corresponding label maps are loaded into Google Colab. Thresholding is performed on the images to reduce the CT artifacts like noise and beam hardening. The images are then normalized to values between 0 and 1. Feature maps are calculated for all images. A feature map of an image is generated by extracting feature signatures of each voxel in the image using the learned ISA network.

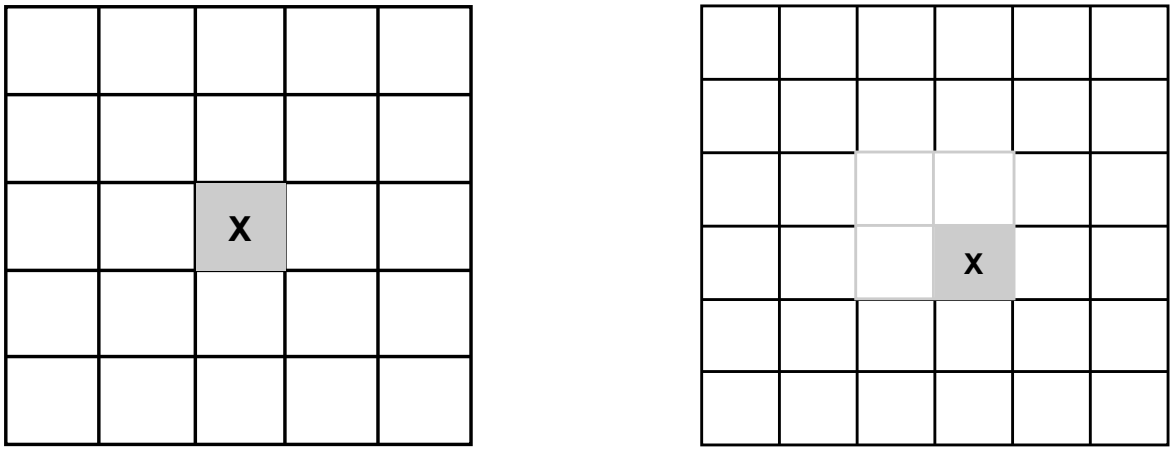
For each voxel in the image, a large patch is extracted with the selected voxel at the center. Or for a large patch extracted, the patch is assigned to the center voxel of that patch. For a patch with an axis being an even number i.e., there is no center. The center of such patch is considered to be the voxel with the highest coordinate in the centermost patch of size $2 \times 2 \times 2$. This is illustrated using Figure 3.6 with 2D blocks of sizes 5×5 and 6×6.

The patch size is the same as that used in the stacked ISA network. Padding operation is performed on every axis to include feature signatures for all voxels. The large patch is then traversed through the entire ISA network. Features learnt from both the layers of stacked ISA are concatenated and used as feature signatures. The feature map is now a 4D array with the last dimension size equal to the number of features of the last layer of stacked ISA.

3.2.4 Image Segmentation using Multi Atlas-based Sparse Label propagation

The technicalities involved in the process of multi atlas-based segmentation is discussed in this section.

[7] https://www.slicer.org/wiki/Documentation/4.4/Modules/Registration
[8] https://www.slicer.org/wiki/Documentation/4.1/Modules/BRAINSResample



(a) Center pixel X of a 5×5 block

(b) Center pixel X of a 6×6 block obtained by choosing the highest coordinate in the center most 2×2 block

Figure 3.6: Selection of center voxel for a patch in Feature Signature Extraction method

3.2.4.1 Label propagation and sparse coding

Automated segmentation is a much-needed improvement as it decreases the dependency of human experts and also saves time. Multi Atlas-based segmentation helps to evaluate a new image using the knowledge of experts on existing sample data. By having prior knowledge, the method is able to predict the label of each voxel in the new image.

The labels obtained from training images are propagated to the target image coordinates and are merged using a label fusion algorithm.

In this method, the label at the target voxel of the target image is calculated by combining the labels in the corresponding location in the training label maps. The intensity voxels are replaced by their corresponding feature signatures which are extracted using feature extraction techniques. The label fusion method combines this information to produce the segmentation output. To mitigate the mismatch in alignment caused by image registration, this technique fuses the information from a small 3D patch around the voxel. This reduces the effect of bad registration and introduces information about the neighborhood voxels. Thus, the more similar the patches are, the higher the weight it carries. Label propagation can be defined as follows.

Consider a target image I_{new} that needs to be segmented. Given T training images I_t and their label maps L_t registered to the target image I_{new}, the label at each voxel position $x \in I_{new}$ can be calculated using equation[Lia+13b]:

$$L_{new}(\boldsymbol{x}) = \frac{\sum_{t=1}^{T} \sum_{\boldsymbol{y} \in N_t(\boldsymbol{x})} w_t(\boldsymbol{x}, \boldsymbol{y}) L_t(\boldsymbol{y})}{\sum_{t=1}^{T} \sum_{\boldsymbol{y} \in N_t(\boldsymbol{x})} w_t(\boldsymbol{x}, \boldsymbol{y})} \tag{3.3}$$

Where $N_t(x)$ denotes the search neighbourhood in each training image. $W_t(x, y)$ represents

the weight of each training image voxel's $y \in I_t$ contribution to the target voxel $x \in I_{new}$. L_{new} is the probability label map obtained of the target Image I_{new}.

The weights are determined by considering the similarity of the patches. In this approach, sparse coding is used for obtaining sparse representation of feature signatures. These representations are used for assigning weights to the corresponding voxel in 3.3. The Sparse coefficient vector is estimated by minimizing equation[Lia+13b]:

$$E(\boldsymbol{\beta_x}) = \frac{1}{2} \|\boldsymbol{f_x} - \mathbf{A}\boldsymbol{\beta_x}\|_2^2 + \lambda \|\boldsymbol{\beta_x}\|_1 , \quad \boldsymbol{\beta_x} \geq 0 \tag{3.4}$$

f_x is the feature signature of the voxel x in target image I_{new}. For each voxel y in the search neighborhood of the target voxel in the training image, the feature signature is calculated and organized into a Matrix A. The feature signatures are arranged as column vectors. For each of the target voxel x, the search neighbourhood is a 3D patch denoted by M_x which has a patch size of $(2r+1) \times (2r+1) \times (2r+1)$ where r is the search radius of the search neighborhood. The center voxel is removed as only the contribution of the neighbourhood is considered. In total, there are $(2r+1)^3 - 1$ training image voxel per training image in the neighborhood of target voxel x. These feature vectors organised into Matrix A forms a dictionary that is then used in the equation. The weight $W_t(x, y)$ is then assigned the corresponding element in the optimal sparse coefficient vector β_x^{Opt}. In this thesis, the built-in `sparse_encode` method in the `sklearn.decomposition` library[9], is used for the sparse coding method.

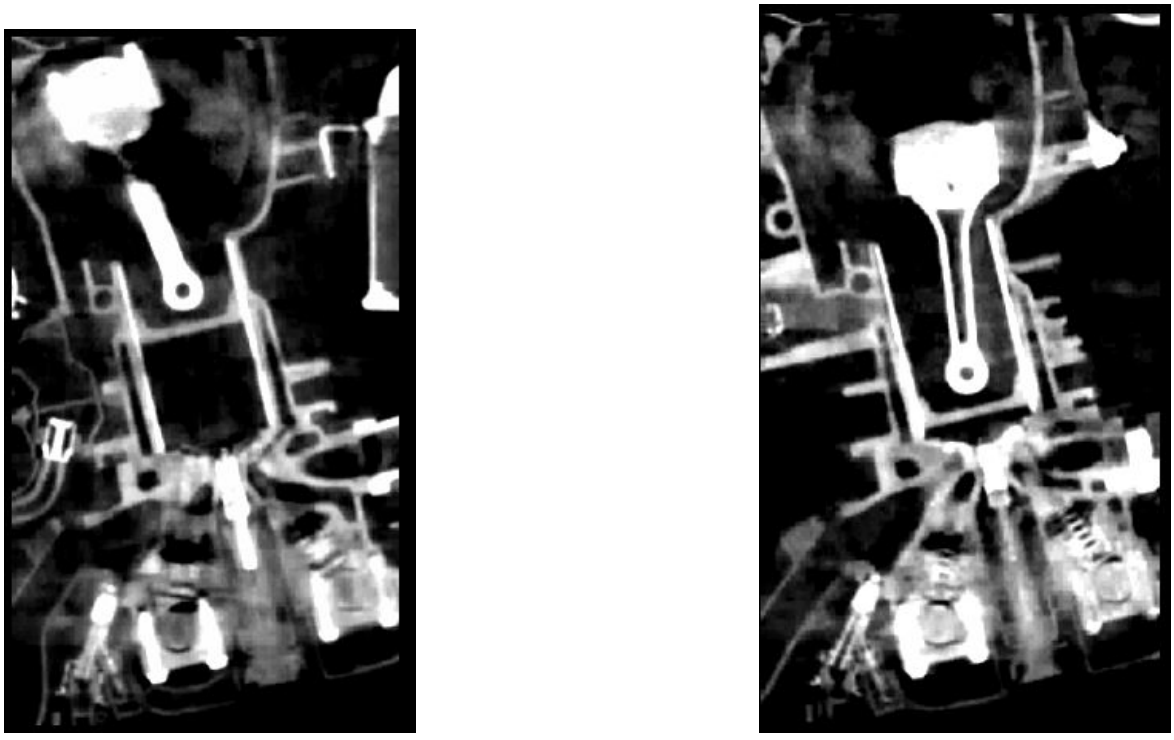
To summarise, after calculating the feature signatures for each voxel in the target image and their corresponding registered training images, the extracted feature signatures are integrated into the Label propagation method where the weights for each training image voxel are estimated using the sparse coding technique. The probability map obtained from label fusion is then used to estimate the target label. In this work, voxels with label probability $L_{new}(x) > 0.5$ are considered as belonging to the connecting rod region. Voxels with $L_{new}(x) \leq 0.5$ are determined as background. With this, the Label map of the target image is obtained.

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparseCoder.html`

4 Experimental Setup

4.1 Dataset

The dataset used in the evaluation of stacked ISA consists of four 3D CT scans of a combustion engine containing connecting rods also known as con rods. The con rods are responsible for converting the up and down linear motion of the piston to the rotational motion of the crankshaft which then operates the mechanism it is connected to. They are a crucial link of power transfer between the piston and crankshaft of an engine. It is required to move over thousands of times a minute and as smoothly as possible. Engines having multiple cylinders have multiple con rods[Wik]. The scans belong to the engine of a Ford Fiesta car which has a motor consisting of four cylinders, hence the four con rods. These scans are of different resolutions and the size of each individual volume element is 1.6 mm. The scans are measured on a XXL CT system developed at the Fraunhofer EZRT in Fürth [fra]. The visualization of some of the scans using the software tool 3D slicer is shown in Figure 4.1.



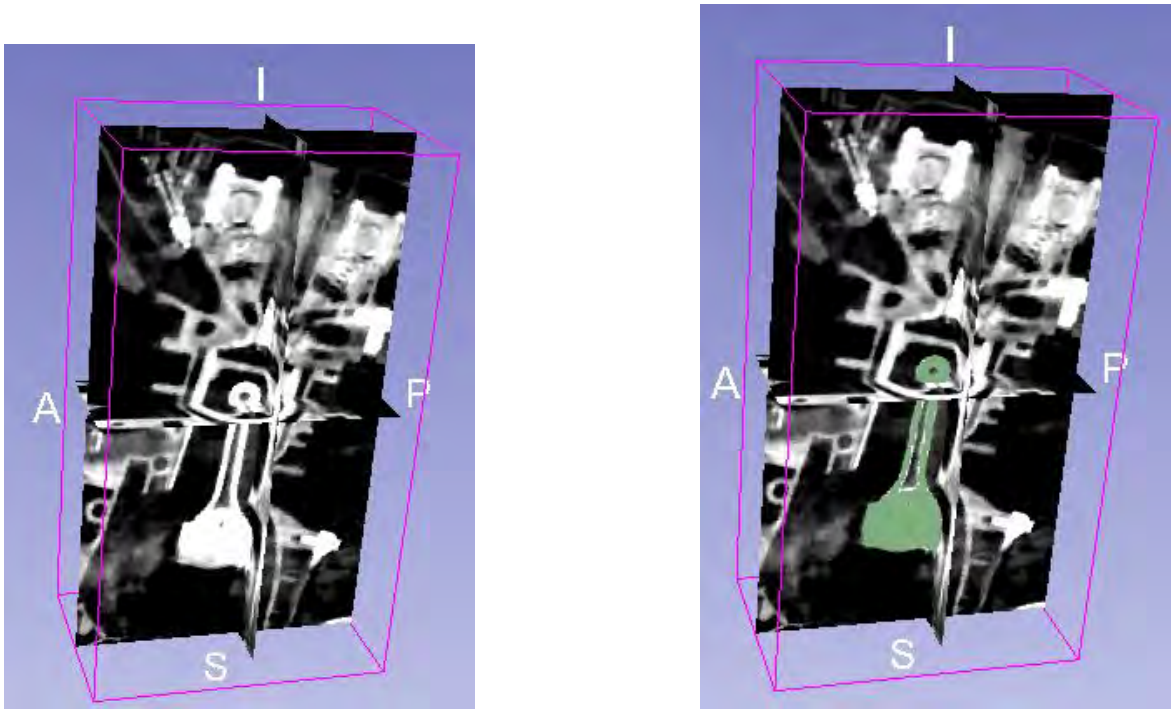
(a) Slice from xy plane of a scan with dimension $286 \times 161 \times 56$

(b) Slice from xy plane of a scan with dimension $286 \times 161 \times 64$

Figure 4.1: Visualization of conrod scans in 3D slicer tool

4.1.1 Annotation

The annotation of the scans is done using 3D slicer [1]. The segment editor module is used for manual annotation. It offers different ways to annotate the images. It provides an interface similar to painting applications but for annotating 3D voxels. Multiple overlapping and non-overlapping segments can be annotated. The image can be annotated both in 3D and 2D(slice by slice). To ease the annotation procedure, the threshold option is selected where the regions can be highlighted based on intensity values. The Intensity range is chosen such that only the region of interest is highlighted. With the help of masking, only the highlighted regions can be annotated without worrying about annotating the low-intensity regions. Figure 4.2 shows an orthogonal view of a 3D scan and its corresponding manually annotated segment. Figure 4.3 shows a 3D view of a segmented connecting rod.



(a) Orthogonal View of the scan without Label

(b) Orthogonal View of the scan with Label

Figure 4.2: Orthogonal view of conrod scans with dimension $286 \times 161 \times 64$ in 3D slicer tool

4.2 Software and Hardware

The environment in which the the framework is implemented :

- Google Colab

[1] https://www.slicer.org/

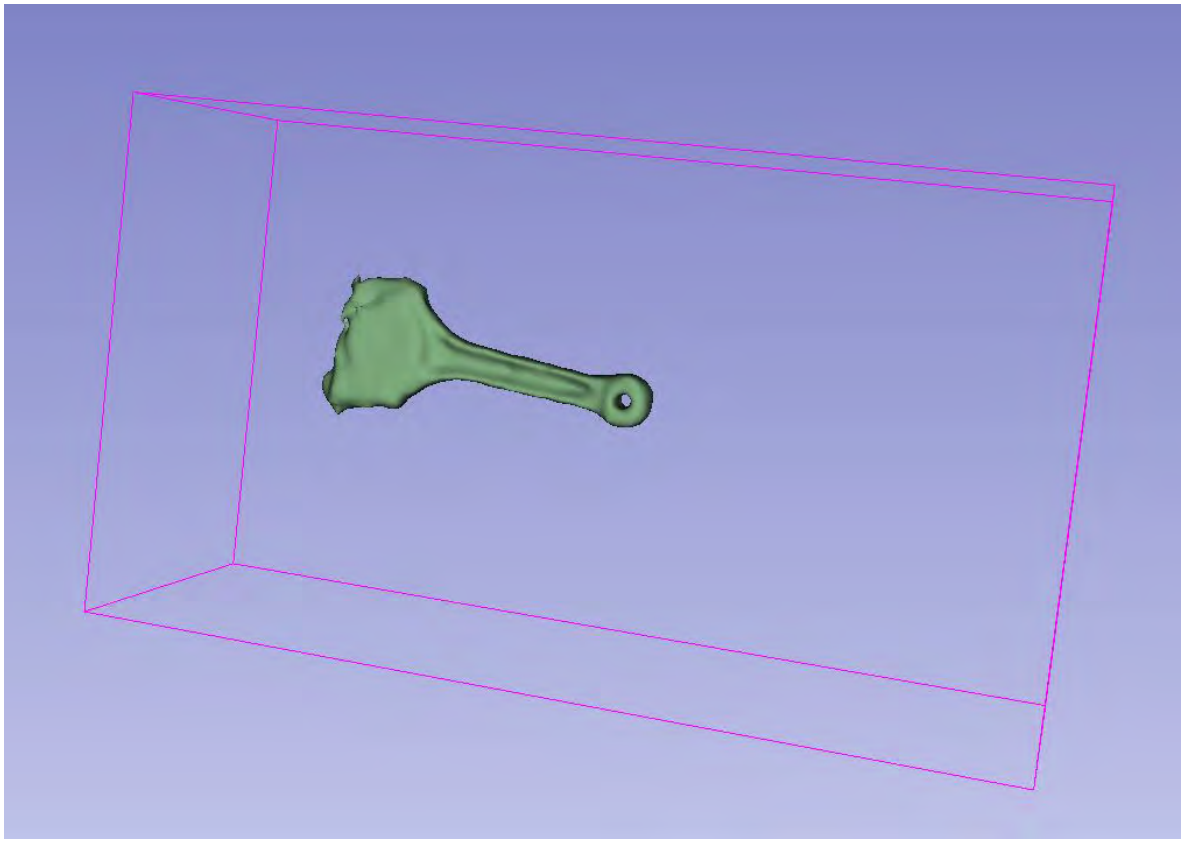


Figure 4.3: 3D view of a segmented conrod

- Python 3.7.10
- Libraries used : numPy[2], matplotlib[3], TensorFlow[4], Numba[5],scikit-learn[6]

Google Colab short for Google Colaboratory is a product from Google Research. It is a free cloud service with a dedicated Graphical Processing Unit (GPU) which can be used to perform Machine Learning and Deep Learning tasks. Python was chosen because of the useful libraries it has for deep learning and also due to the previous knowledge in the language. Open-source libraries and packages help in implementing the ideas at a higher level. NumPy is used for its ease of handling multi-dimensional arrays and for performing mathematical operations. Matplotlib is a data visualization library that offers inline visualization of plots. TensorFlow mainly focuses on deep learning related tasks. Numba helps translate python code to fast machine code that can be run on GPUs. scikit-learn is a very useful library in the field of machine learning.

3D Slicer is an open-source software for image analysis and visualization supported on Windows, Linux and macOS. It is mainly used in the medical domain for the study and analysis of

[2] https://numpy.org/
[3] https://matplotlib.org/
[4] https://www.tensorflow.org/
[5] https://numba.pydata.org/
[6] https://scikit-learn.org/stable/

3D scans belonging to internal organs and structures. It offers various functionalities. Interactive 3D volume visualization, Image registration, Image segmentation, among many others are supported by 3D slicer. It can also handle several formats of data [Sli].

Increase in research in the field of deep learning has led to an increase in the availability of GPUs and GPU compatible libraries. GPUs provide parallel computing capability for faster execution of operations on large datasets. Google colab has a Tesla K80 with 2496 CUDA cores and 12GB GDDR5 VRAM. The CPU is Intel Xeon Processors with 12.6GB RAM running at 2.3Ghz with 1 core and 2 threads.

4.3 Performance Metrics

For statistical analysis, the broadly known accuracy, precision, recall, and F1 score is used. Accuracy is a simple metric that provides the percentage of correct predictions over the total number of predictions. This will work the best when there are equal samples in the classes. Especially in the field of deep learning or image segmentation in general, most of the time, the problem of class imbalance is encountered. The accuracy metric alone is sometimes misleading as some models with high accuracy will not perform well on the test data. Hence, Precision, Recall, and F1 scores are used. These can be measured using the equations :

$$\text{Precision } = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \tag{4.1}$$

$$\text{Recall } = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{4.2}$$

$$\text{F1 score } = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision + Recall}} \tag{4.3}$$

Classifying the model's prediction into true positives, false positives, true negatives, false negatives are not very clear when it comes to a segmentation output. A more intuitive way of looking into the pixels that belong to a certain region is required.

Metrics that are best suited for Image segmentation models are IoU, Dice ratio, etc. In this work, to evaluate the model, quantitative assessments such as IoU, Mean dice ratio, Hausdorff Distance and Average Surface distance are used.

IoU short for Intersection Over Union, calculates the region of overlap between the predicted mask and the ground truth mask. It is the ratio of the number of pixels common to both target and prediction, to the pixels that are present in both. It is given in equation 4.4 [Rez+19]. It is also known as the Jaccard index.

$$IoU = \frac{\text{Ground Truth } \cap \text{ Predicted Mask}}{\text{Ground Truth } \cup \text{ Predicted Mask}} \tag{4.4}$$

The mean Dice ratio is also a similarity measure that can be calculated using equation 4.5 [TH15]. It measures the overlap of pixels between the target and the prediction divided by

the number of pixels in each set. Both IoU and Dice ratio are influenced by the size of the dataset. The range of both these metrics falls between 0 to 1 with 0 as no overlap and 1 being a good prediction with complete overlap[Gro].

$$\text{Dice Ratio} = 2 * \frac{\text{Ground Truth } \cap \text{ Predicted Mask}}{\mid \text{Ground Truth } \mid + \mid \text{Predicted Mask } \mid} \tag{4.5}$$

Distance-based metrics based on surface distance are used to calculate the degree of accuracy incorporating the real spatial resolution of the 3D scans. The comparison is made on the Predicted Mask(PM) with respect to the ground truth(GT). The surface distance for a voxel x belonging to one Image to a set of voxels A is given as [YV15]:

$$d(x, A) = \min_{y \in A} d(x, y) \tag{4.6}$$

Where d is the euclidean norm between x and y. Average Surface Distance(ASD) and Hausdorff Distance are based on this distance measure. Calculating the surface distance for all the voxels on the boundary of the GT to the voxels on the boundary of the PM gives the total surface distance denoted as $d(B_{GT}, B_{PM})$. The mean of the total surface distance $d(B_{GT}, B_{PM})$ and $d(B_{PM}, B_{GT})$ is the average surface distance(ASD) given by [YV15]:

$$ASD = \frac{1}{|B_{PM}| + |B_{GT}|} \times \left(\sum_{x \in B_{PM}} d\left(x, B_{GT}\right) + \sum_{y \in B_{GT}} d\left(y, B_{PM}\right) \right) \tag{4.7}$$

Hausdorff distance(HD) is the largest difference between the surface distance. It can also be stated as the maximum distance from a point in a set A to the closest point in set B :

$$\text{HD} = \max\left[d(B_{GT}, B_{PM}), d(B_{PM}, B_{GT})\right] \tag{4.8}$$

The Hausdorff distance is highly affected by outliers. Since the ASD is averaged over all distances, it is expected to be stable [TH15]. Both these metrics are measured in mm and incorporate the spacing of voxels. The lower the distance measures, the better the accuracy.

5 Results

In this chapter of the thesis, the details of various experiments performed are provided. The outcomes of each of the experiments are also presented. All the experiments performed are evaluated using Leave-One-Out Cross-Validation (LOOCV). In LOOCV, the number of times the algorithm is run is equal to the number of samples in the dataset. In this cross validation method, one sample is used for testing and the rest samples are considered as train dataset. Thus, it is expensive to perform, but it provides reliable results.

The main parameters involved in this study are patch size for layer 1 and layer 2 of the stacked ISA, the output dimension of each ISA. Patch size represents the size of the patch extracted from an image in x. The Output dimension for each layer is the number of components in the pooling layer of the ISA. They are evaluated using the robust performance metrics IOU score, Dice ratio for 3D segmentation. The Average Surface Distance (ASD) and Hausdorff distance (HD) measures are also measured. The experiments are performed for each of the parameters mentioned below.

- The patch size used in training the layers of ISA
- With respect to x and y axis keeping z constant
- With respect to z axis keeping x and y constant
- The output neurons used in each layer of ISA
- Using only the first layer of ISA

The parameters mentioned below are fixed during each of these experiments.

- The number of iterations required for training the ISA is set to 300 for both layers.
- The number of patches extracted from each image for training is 2000.
- The neighbor radius used in equation 3.3 is set to 2.
- The subspace size of the first ISA layer is 1 and the second ISA layer is 2. This represents the size of subspace where the components within the subspace can be dependent.

The workflow consists of a training stage and a testing stage. The flow of the architecture along with the parameters used in this work, is explained using an example in the two algorithms 1 and 2 [Lan+15]. The parameters in the algorithm represent the model constraints and arguments that are particular to this work.

Table 5.1 shows the parameters used in this example.

Stacked ISA Layer	Patch Size			Patches Extracted per Image	Subspace size	Output Dimension	Iterations
	X	Y	Z				
1	17	17	2	2000	1	60	300
2	21	21	3	2000	2	40	300

Table 5.1: Parameters chosen

Algorithm 1 Training stage:

Input: Sample N patches from each image

$$X_{first} = [x_1, x_2, ..x_i..x_n]$$
$$\text{where } x_i \in R^{17\times17\times2}$$

—Training the first ISA layer—

Train and apply PCA_1 on the Input X_{first} with dimension of 60

$$PCA_1 = PCA(X_{first}, 60)$$
$$X_{PCA1} = PCA_{apply}(X_{first}, PCA_1)$$

Train ISA on X_{PCA1} with subspace size as 1 and store the ISA model as ISA_1

$$ISA_1 = ISA_{train}(X_{PCA1}, 1)$$

—Training only the second ISA layer using a greedy training approach—

Input: Sample N patches from each image

$$X_{second} = [x_1, x_2, ..x_i..x_n]$$
$$x_i \in R^{21\times21\times3}$$

X_{second} is convolved with a stride of [4,4,1] i.e., it is decomposed into 8 patches each of size $R^{17\times17\times2}$

PCA_1 followed by ISA_1 is applied to convolved input X_{conv}.

$$X_{ISA_1} = ISA_{apply}(ISA_1, PCA_{apply}(PCA_1, X_{conv}))$$

The outputs from the decomposed 8 patches are concatenated(denoted as $Xout_1$)

$Xout_1$ is the input to second layer.

Train and apply PCA_2 on the Input $Xout_1$ with dimension of 40.

$$PCA_2 = PCA(Xout_1, 40)$$
$$X_{PCA2} = PCA_{apply}(Xout_1, PCA_2)$$

Train and apply ISA on X_{PCA2} with subspace size as 2 and store the ISA model as ISA_2

$$ISA_2 = ISA_{train}(X_{PCA2}, 2)$$
$$Xout_2 = ISA_{apply}(ISA_2, X_{PCA2})$$

Output: ISA_1, PCA_1 and ISA_2, PCA_2

Algorithm 2 Testing stage :

Input : Target image I_{new} and training images $I_t(t = 1, ..., T)$, where T is the number of training images registered on I_{new}

—Extracting feature signature from the images using trained ISA model—-

Zero Padding is performed on the image as required.

for all voxel in images :

    Patches of size $21 \times 21 \times 3$ are extracted.

for all patch in patches extracted :

    Steps 5 to 8 from algorithm 1 is performed and output $Xout_1$ is obtained

    PCA_2 obtained from algorithm 1 is applied to $Xout_1$.

$$X_{PCA2} = PCA_{apply}(Xout_1, PCA_2)$$

    ISA_2 is applied to X_{PCA2}.

$$Xout_2 = ISA_{apply}(ISA_2, X_{PCA2})$$

    X_{PCA2} and $Xout_2$ are concatenated to form the feature vector.

$$fv = [X_{PCA2}, Xout_2]$$

X_{PCA2} is the PCA reduced output of the first ISA layer. These feature vectors obtained for each voxel represent the feature signature of that particular voxel. The feature signatures is extracted for all images.

— Label Propagation using sparse coding—

for all voxel x in I_{new} :

    for all Training image I in $I_t(t = 1, ..., T)$:

        For neighbour radius r of 2, a patch of size $5 \times 5 \times 5$ is extracted in the I the target voxel x location at the center.

        The center voxel is removed from the patch.

    The remaining patch from all I containing feature signatures $f_i(i = 1, .., T * (2r + 1))$ in column vector format are organised into a matrix A.

    The feature signature of x is stored as f_t.

    Using matrix A and f_t, the sparse coefficient vector β_x is calculated using equation 3.4.

    Each element of β_x is a coefficient vector corresponding to the elements in A.

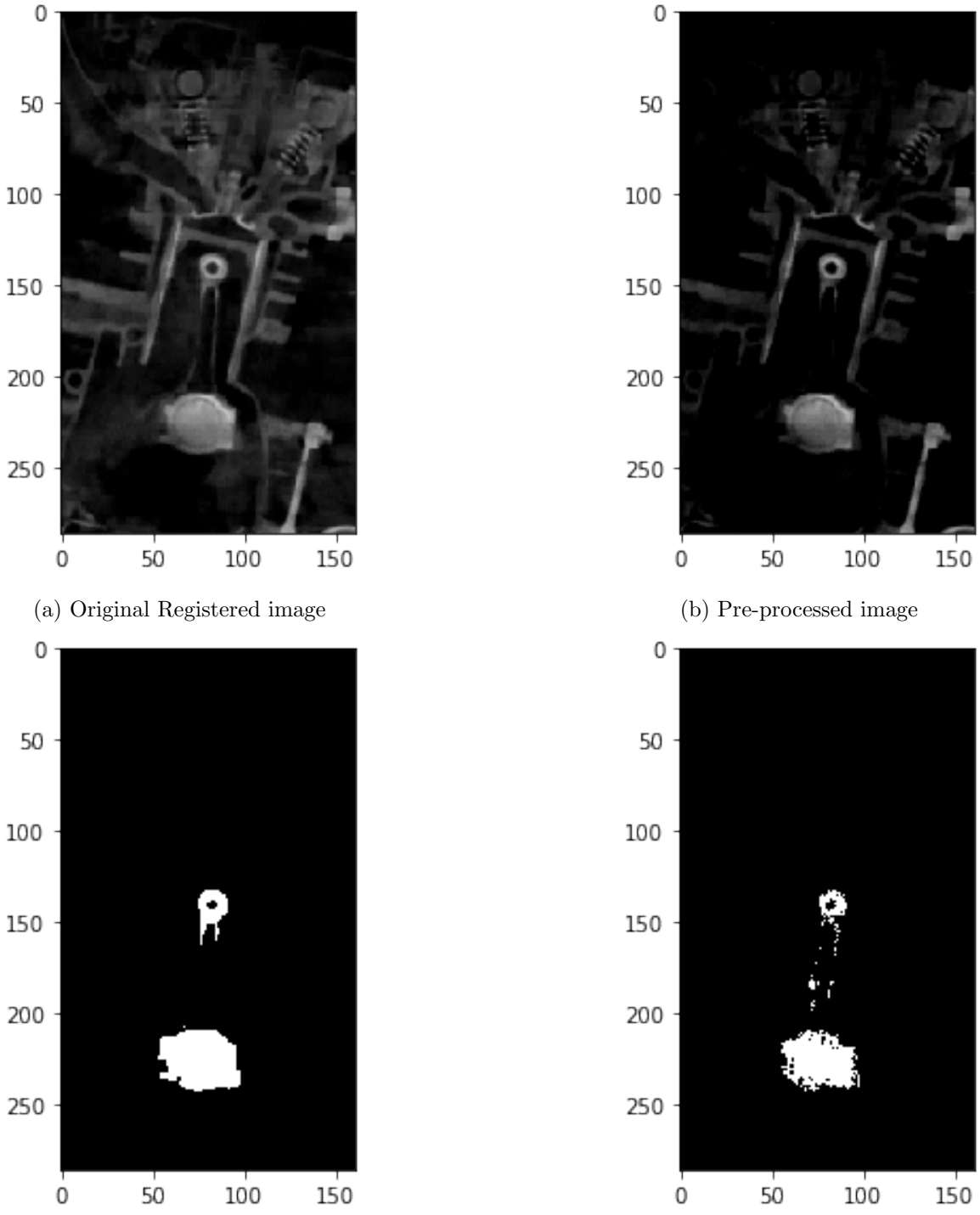
    The optimal coefficient vector values obtained by solving the sparse coding equation 3.4 is used to set to the graph weight $W_i(x, y)$ which is used in equation 3.3.

    Label probability at voxel x is calculated using equation 3.3.

    Label confidence is defined as :

$$L(x) > 0.5 \Rightarrow \text{Class 1}$$
$$L(x) \leq 0.5 \Rightarrow \text{Class 0}$$

The Predicted Label map is compared with the Actual target mask and evaluated using various performance metrics



(a) Original Registered image

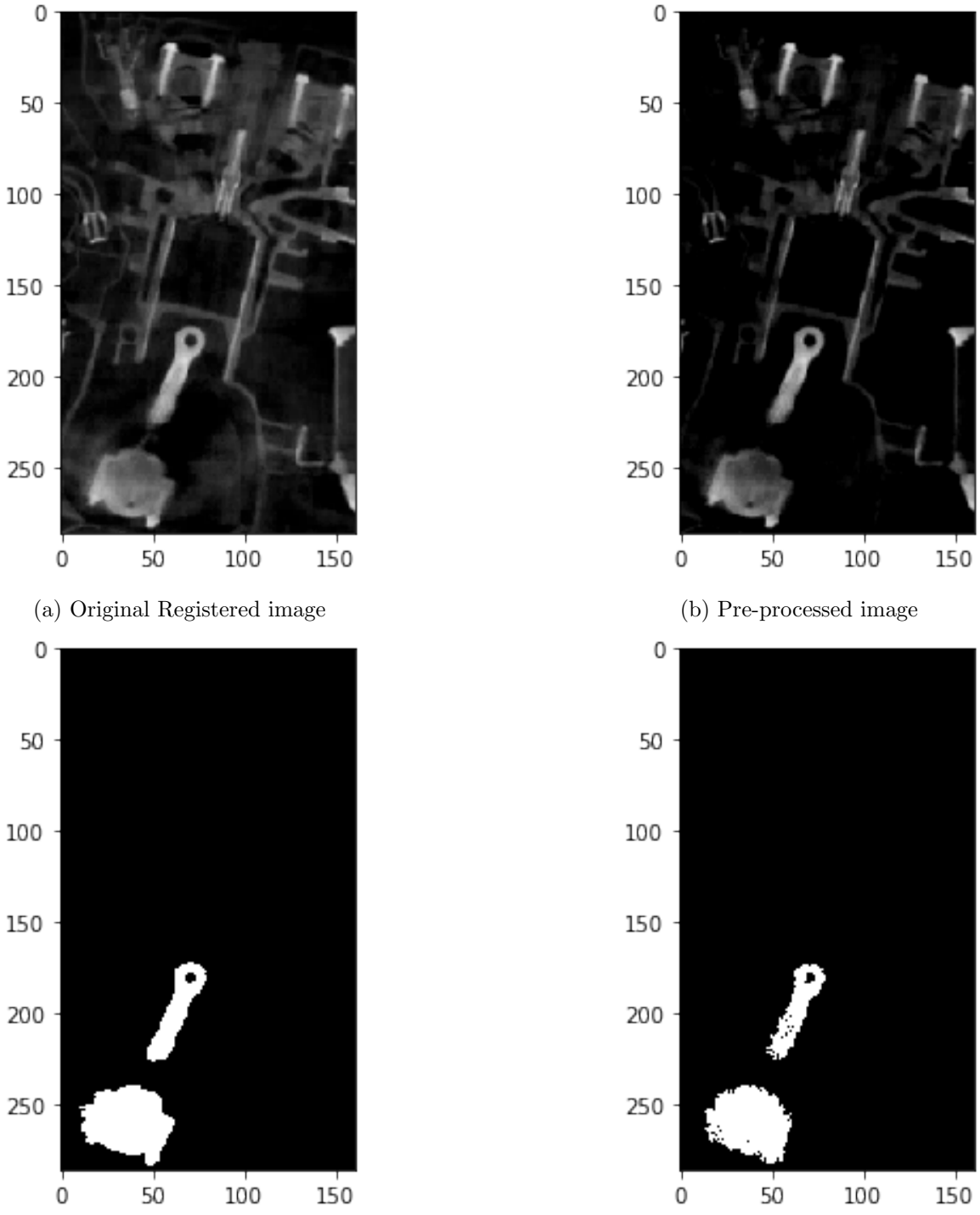
(b) Pre-processed image

(c) Actual mask

(d) Predicted mask

Figure 5.1: Subject 1, slice 22

The Figure 5.1,5.2,5.3,5.4 shows the output label maps predicted using this architecture and the actual label map for its corresponding target image. The parameters used for this



(a) Original Registered image

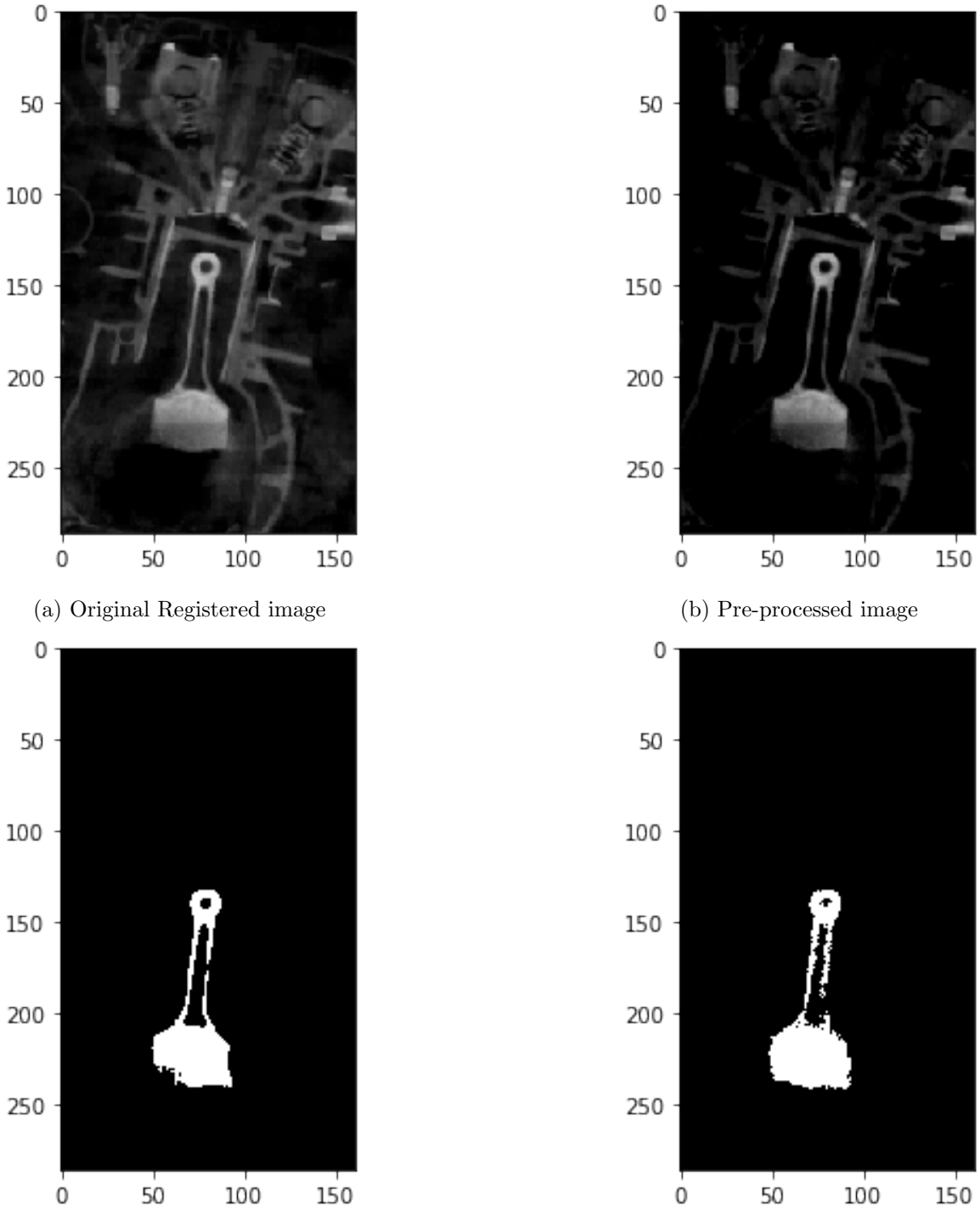
(b) Pre-processed image

(c) Actual mask

(d) Predicted mask

Figure 5.2: Subject 2, slice 26

demonstration are those mentioned in the above example. It can be observed that for the slices from subjects 2, 3 and 4 the output label map is very close to ground truth whereas, in



(a) Original Registered image

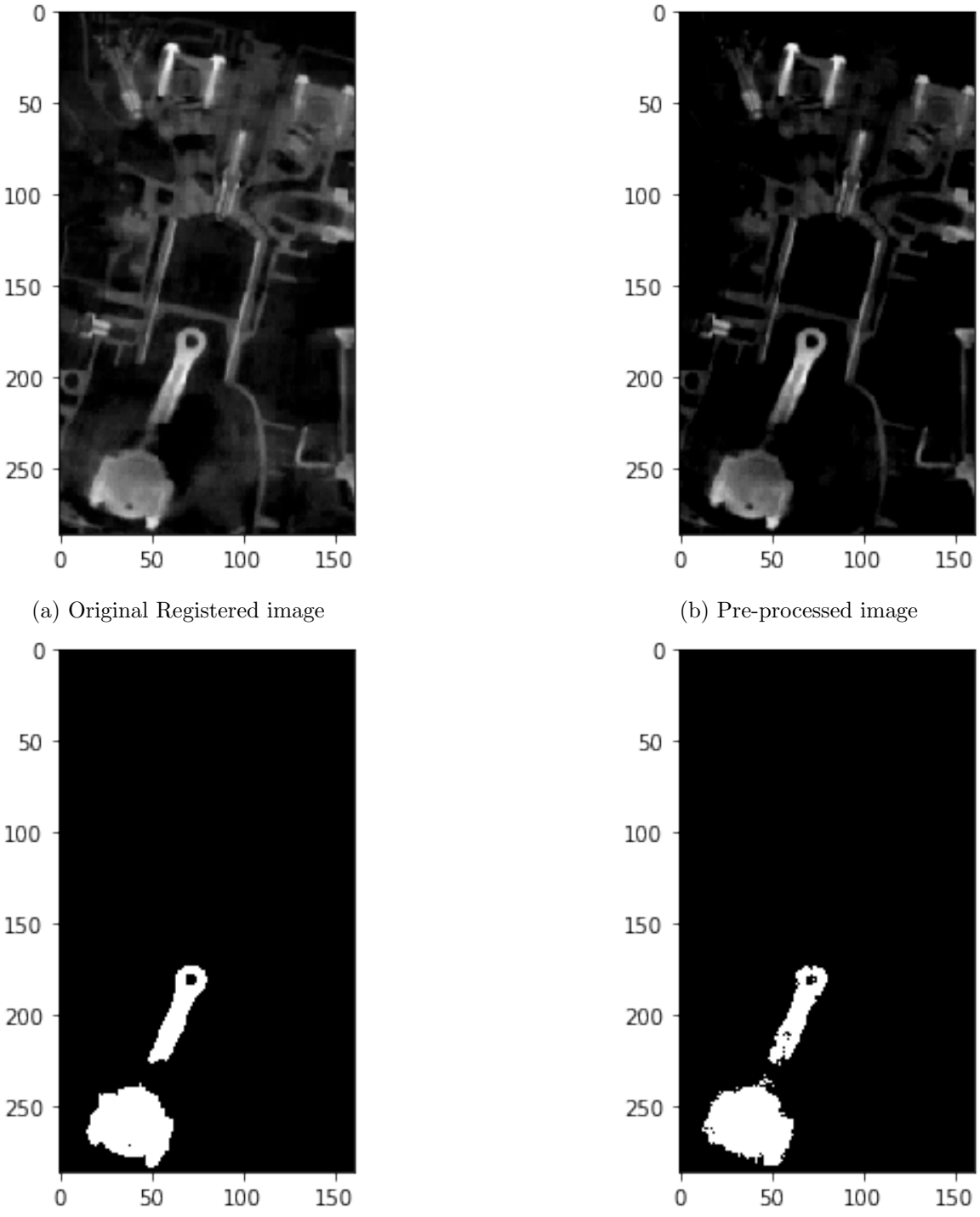
(b) Pre-processed image

(c) Actual mask

(d) Predicted mask

Figure 5.3: Subject 4, slice 77

the first subject, the predicted mask is less dense and slight mismatch with the actual mask.



(a) Original Registered image

(b) Pre-processed image

(c) Actual mask

(d) Predicted mask

Figure 5.4: Subject 3, slice 28

This could be due to the reason that slice 22 is the edge slice containing the connecting rod.

IOU score (Mean ±SD)	Dice Ratio (Mean ±SD)	Average Surface Distance (Mean ±SD)	Hausdorff Distance (Mean ±SD)
0.763 ±0.014	0.865 ±0.009	1.22 ±0.057	17.66 ±5.11

Table 5.2: Performance Measures

For illustration purposes, consider one of the images from the dataset as the target image. This target image has dimensions $286 \times 161 \times 56$. The registered training images also have the same dimension. The time taken to extract the feature vectors for each image is 4.26 minutes. The label propagation step takes around 3.35 minutes. With 3 registered training images and one target image, the total time taken for segmenting an image of dimension is 20.39 minutes. The training time of the first and second layers of stacked ISA is 37.8 seconds and 15.8 seconds respectively. Table 5.2 lists the mean values and standard deviation for various performance metrics obtained for the predicted label map.

The filters learned in the first layer of ISA are illustrated in the Figure 5.5. As can be observed, there are a variety of patterns learned. Some of them represent filters learnt through Gabor filters with different frequencies and orientations. Some represent much more complex features. The subspace size is set to 1, thus all the filters are independent of each other. The two adjacent filters represent the two slices in the z axis.

The Figure 5.6 illustrates the filters learned in an ISA network for an image of dimension $21 \times 21 \times 3$ with the number of linear components set to 60 and subspace size of 2. Filters adjacent and close to each other belong to the same group. Each row shows three sets of grouped filters. Each set of filters is in 3D but only the center slice is considered for visualization purposes. Patterns can be observed between the filters belonging to the same group. It can be seen in row one-column one that the two sets of filters learned have translational invariance. In row four-column one it can be seen that complex change-like intensity values have been learned. In row one-column two, rotational invariance can be observed. Other groups seem to have captured other complex features.

5.1 Effect of Image patch size

Two sets of evaluations are performed in this section. The patch size is considered to be $x \times y \times z$.

- Varying x and y while keeping z constant(corresponding output dimension of the ISA is increased linearly to accommodate the new patch size)
- Varying z axis while keeping x and y constant(corresponding output dimension of the ISA is increased linearly to accommodate the new patch size)

Varying x and y while keeping z constant
In this experiment, the performance is measured with respect to changes in x and y axis of the

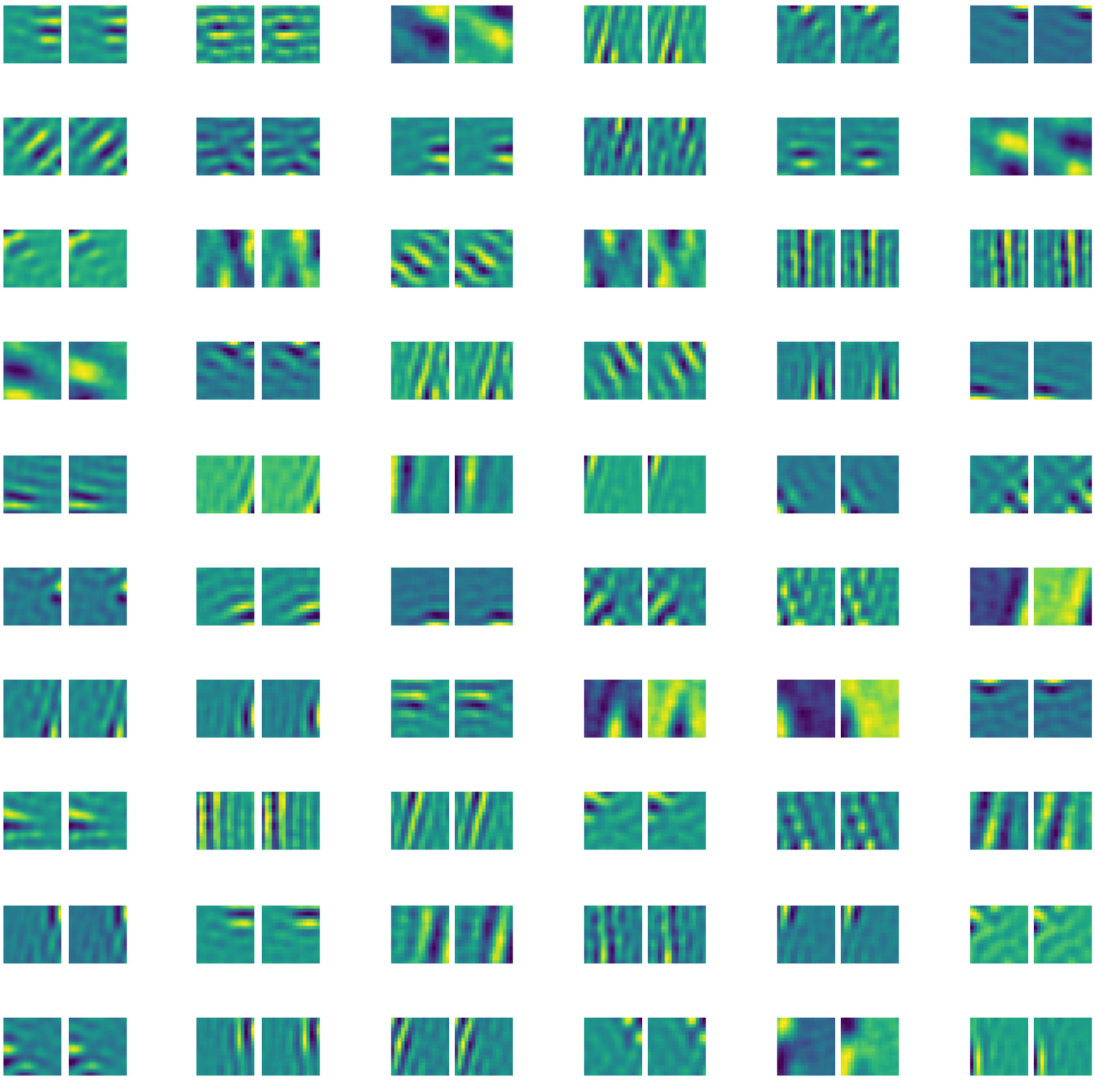


Figure 5.5: All Filters learned from the first layer of the stacked ISA network.

patch size. The experiment ID along with the corresponding parameters used is mentioned in the Table 5.3.

From the results in 5.8, it can be observed that with the increase in the patch size x and y, the segmentation results obtained are better. After a point ie., at D, the increase in both IOU and Dice is less when compared with point A to point C. The slight increase at higher patches can be considered but at the cost of increased computational power. Table 5.4 provides the Average Surface Distance(ASD) and Hausdorff distance(HD) in Mean ±Standard deviation format.

Varying z axis while keeping x and y constant

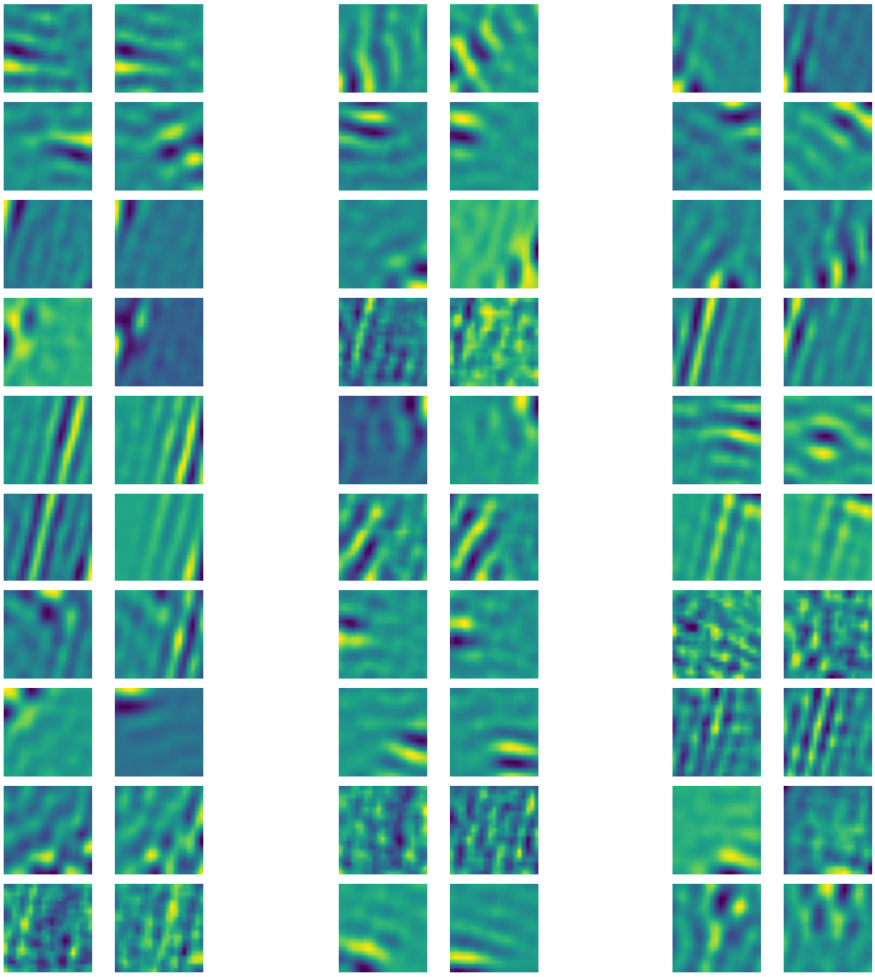


Figure 5.6: Filters learned from the first layer of the stacked ISA network with a subspace size of 2.

In this study, the performance of the framework is compared to changes in the patch size with respect to the z axis. Table 5.5 provides the experiment ID along with the corresponding parameters used.

From the results in Table 5.6, it can be observed that with increase in the patch size z, the segmentation results obtained are better. It is important to note that the increase of patch size z from 2 to 3 is a 50% increase in the overall patch size for the first ISA and the increase of patch size z from 3 to 4 is a 33.33% increase in the overall patch size for the second ISA which increases the computational power needed by a big number. A trade-off between better segmentation result and computational cost is important, however, in some use cases better segmentation is required even at the cost of high computational power.

Experiment ID	Stacked ISA Layer	Patch Size			Output Dimension
		x	y	z	
A	1	13	13	2	34
	2	17	17	3	22
B	1	15	15	2	45
	2	19	19	3	30
C	1	17	17	2	60
	2	21	21	3	40
D	1	19	19	2	72
	2	23	23	3	48
E	1	21	21	2	90
	2	25	25	3	60
F	1	23	23	2	100
	2	27	27	3	66
G	1	25	25	2	120
	2	29	29	3	80

Table 5.3: Experiment ID and its corresponding parameters (A-G).

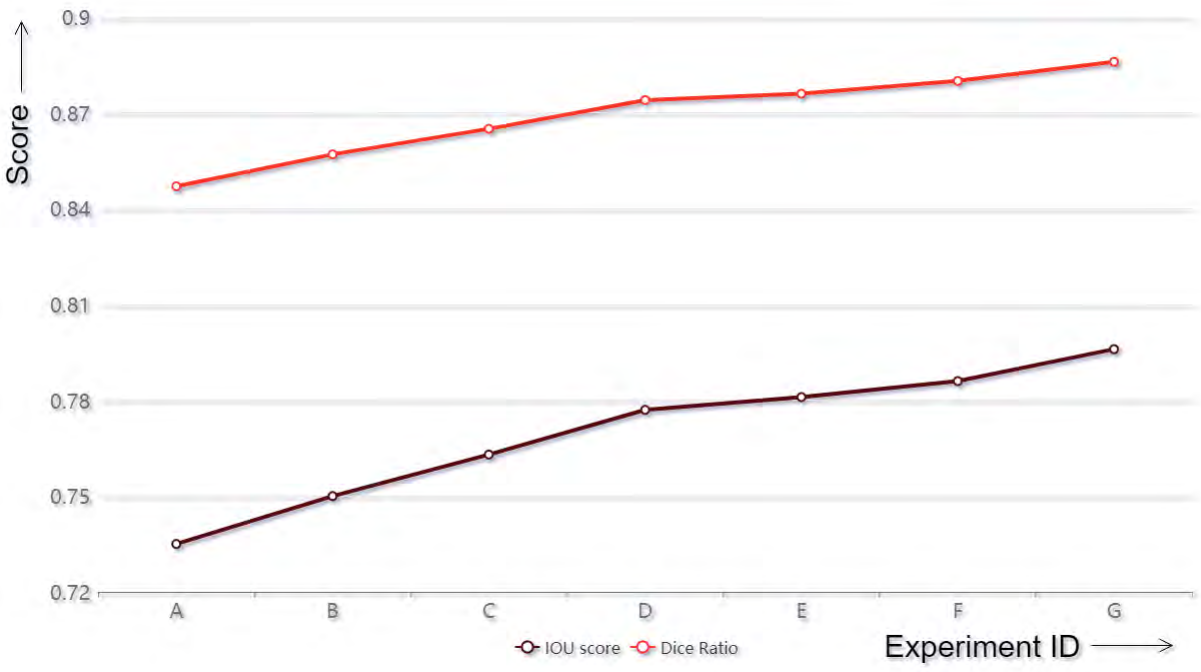


Figure 5.7: Effect of patch size on segmentation (A-G).

Experiment ID	Average Surface Distance (Mean ±SD) (in mm)	Hausdorff Distance (Mean ±SD)(in mm)
A	1.33 ±0.058	17.96 ±3.918
B	1.24 ±0.091	17.71 ±4.36
C	1.22 ±0.057	17.66 ±5.11
D	1.17 ±0.06	17.49 ±5.78
E	1.14 ±0.06	17.6 ±5.18
F	1.1 ±0.09	16.55 ±5.4
G	1.07 ±0.07	16.78 ±5.63

Table 5.4: Performance measure for corresponding experiments (A-G).

Experiment ID	Stacked ISA Layer	Patch Size			Output Dimension
		x	y	z	
H	1	17	17	1	28
	2	21	21	2	20
I	1	17	17	2	60
	2	21	21	3	40
J	1	17	17	3	86
	2	21	21	4	58
K	1	17	17	4	116
	2	21	21	5	78

Table 5.5: Experiment ID and its corresponding parameters (H-K).

Experiment ID	IOU score (Mean ±SD)	Dice Ratio (Mean ±SD)	Average Surface Distance (Mean ±SD) (in mm)	Hausdorff Distance (Mean ±SD) (in mm)
H	0.723 ±0.018	0.839 ±0.012	1.387 ±0.038	17.97 ±3.69
I	0.76 ±0.014	0.865 ±0.009	1.225 ±0.057	17.665 ±5.116
J	0.785 ±0.019	0.879 ±0.012	1.124 ±0.074	17.09 ±5.285
K	0.785 ±0.023	0.879 ±0.014	1.095 ±0.07	17.51 ±4.43

Table 5.6: Performance measure for corresponding experiments (H-K).

5.2 Effect of the output dimensions

For this experiment, the patch size of both the ISA are set constant and the output dimension is varied. The ratio between the output dimension of the first layer and the second layer is always around 1.5. The patch size used for the first and second layer is $17 \times 17 \times 2$ and $21 \times 21 \times 3$ respectively. The Table 5.7 provides the experiment ID along with the corresponding parameters used.

Experiment ID	Stacked ISA Layer	Patch Size			Output Dimension
		x	y	z	
L	1	17	17	2	40
	2	21	21	3	26
M	1	17	17	2	50
	2	21	21	3	34
N	1	17	17	2	60
	2	21	21	3	40
O	1	17	17	2	70
	2	21	21	3	46
P	1	17	17	2	80
	2	21	21	3	54
Q	1	17	17	2	90
	2	21	21	3	60
R	1	17	17	2	100
	2	21	21	3	66

Table 5.7: Experiment ID and its corresponding parameters (L-R).

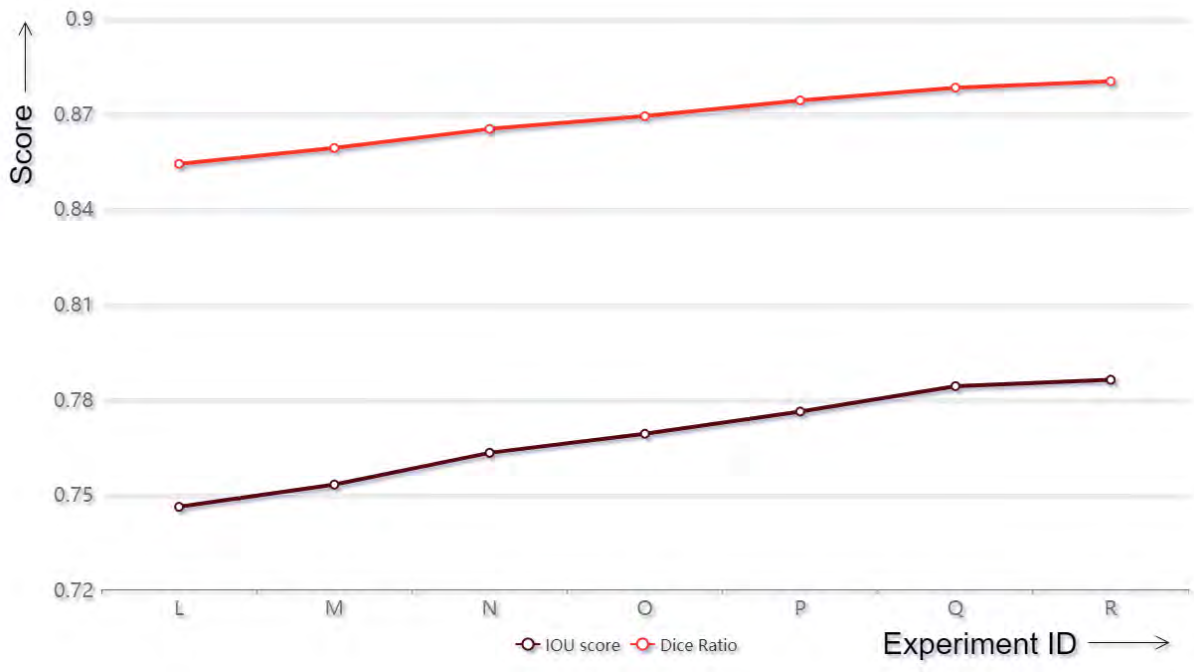


Figure 5.8: Effect of output dimension on Segmentation (L-R).

From the results in Figure 5.8 and Table 5.8 , it can be observed that with the increase in the

output dimension, the performance gradually increases. The increase in performance from L to Q almost seems linear. The ASD measure also improves with the increase in the output dimension. From Q to R, the increase is insignificant suggesting that there are excess features which the model cannot benefit from. As the output dimensions increase, the computational power required also increases. A trade-off between the two has to be considered.

Experiment ID	**Average Surface Distance (Mean ±SD) (in mm)**	**Hausdorff Distance (Mean ±SD)(in mm)**
L	1.298 ±0.078	18.01 ±4.54
M	1.253 ±0.054	17.62 ±5.13
N	1.22 ±0.057	17.66 ±5.11
O	1.19 ±0.085	17.89 ±4.78
P	1.175 ±0.051	17.43 ±4.38
Q	1.149 ±0.045	17.187 ±4.78
R	1.129 ±0.03	17.5 ±4.08

Table 5.8: Performance measure for corresponding experiments (L-R).

5.3 Performance of Single ISA network

The performance of a stacked ISA network is compared with a single ISA network. The experiment ID along with the corresponding parameters used is mentioned in the table 5.9. The performance results can be seen in table 5.10.

As expected, the stacked ISA network performs better than a single ISA. Different combinations of patches are used in the single ISA for better understanding of the performance. Applying large patches on a single ISA is computationally expensive. Using the same large patch in a stacked ISA provides better results and is computationally less expensive as the large patch is reduced to small patches and is used as the input to the first ISA. The reduced output obtained from the first ISA is of a lower dimension, therefore is computationally light on the second ISA. This is the main motivation for implementing a stacked ISA.

5.4 Discussion

In this section, the results for the research questions described in Chapter 1 are presented. Analysis of the unsupervised feature extraction methods based on stacked ISA is performed by integrating it to a Multi Atlas-based segmentation framework. Both qualitative and quantitative results have been discussed.

Experiment ID	Stacked ISA Layer	Patch Size			Subspace Size	Output Dimension
		x	y	z		
S(Stacked ISA for reference)	1	17	17	2	1	60
	2	21	21	3	2	40
T(Single ISA)	NA	17	17	2	1	60
U(Single ISA)	NA	21	21	3	1	40
V(Single ISA)	NA	21	21	3	1	130
W(Single ISA)	NA	21	21	3	2	130

Table 5.9: Experiment ID and its corresponding parameters (S-W).

Experiment ID	IOU score (Mean ±SD)	Dice Ratio (Mean ±SD)	Average Surface Distance (Mean ±SD) (in mm)	Hausdorff Distance (Mean ±SD) (in mm)
S	0.76 ±0.014	0.865 ±0.009	1.225 ±0.057	17.665 ±5.116
T	0.69±0.02	0.81±0.018	1.748 ±0.276	17.198±3.77
U	0.708±0.02	0.829±0.018	1.668 ±0.274	17.061±4.02
V	0.682±0.023	0.811±0.016	1.913 ±0.301	17.511 ±4.992
W	0.647±0.025	0.785±0.0192	2.012 ±0.315	16.40 ±4.494

Table 5.10: Performance measure for corresponding experiments (S-W).

The stacked ISA network is able to learn rich features without the need of labeled data i.e., it can automatically learn features from the 3D images without requiring expert knowledge. These features when used for Image segmentation in MAS provided great results thus confirming the effectiveness of the features. The results obtained for the stacked ISA network outshine that of a single ISA suggesting the importance of a hierarchical network.

Increasing the patch size with respect to spatial axes has an impact on the performance in a positive way. Increasing the output dimension but keeping the patch size constant also increases the performance of the MAS Method. This is due to the fact that an increase in patch size and output results in the storage of more feature information related to the input data. The increase in patch size and output dimension goes hand in hand with the increase in computational power needed. Based on the requirements, a trade-off has to be considered for an efficient feature extraction solution.

The segmentation results for all the experiments performed are obtained using leave-one-out cross validation. This cross validation method helps to avoid biased results. Overall, it can be seen that the stacked ISA approach implemented in this thesis for image segmentation is effective for the dataset at hand.

6 Conclusion and Future Work

In this thesis, an unsupervised Feature extraction method based on a stacked Independent Subspace Analysis (ISA) approach is implemented. This technique automatically learns discriminant features from image patches given. The stacked approach helps in scaling this technique to larger patches. It consists of two layers that are used to learn hierarchical complex features. The features learnt from both layers are combined to form the feature signature required for the task.

Multi Atlas-based Segmentation(MAS) is integrated to evaluate the capability of stacked ISA. Experiments were carried out with various parameter combinations and the results and analysis are discussed. These experiments were run on four 3D CT scans of an engine with the object of interest being the connecting rod. For a two-layered network with the larger patch size being $21 \times 21 \times 3$ and the output reduced to 40 dimensions, IoU score of 0.763 and a dice score of 0.865 is achieved using the MAS method.

The segmentation process takes around 20.39 mins for an image with dimension $286 \times 161 \times 56$ and with 3 training images. Both the qualitative and quantitative analysis suggests good segmentation accuracies. This encourages the research of feature extraction techniques that not only provide good results but also do not require domain-related expert knowledge.

Extracting feature signatures for each voxel in an image is a time-consuming process and is limited by the power of the hardware it runs on. With technological innovations leading to an increase in computational power of GPUs and increasing memory capacities, faster segmentation can be achieved. However, during this time, the resolutions of images also increase which then increases the computational constraint.

Further optimizations could be performed to reduce the segmentation time. The stacked ISA technique could be incorporated with machine learning techniques for Image segmentation as it boils down to voxel classification problem. Since images usually have a high dimension, the training dataset consisting of feature signatures of each voxel could be enormous. To tackle this, one could try a filtering technique to select only the highly probable class voxels for the machine learning task.

Bibliography

[Ala+21] Nasser Alalwan et al. "Efficient 3D Deep Learning Model for Medical Image Semantic Segmentation". In: *Alexandria Engineering Journal* 60.1 (2021), pp. 1231–1239.

[Alj+09] Paul Aljabar et al. "Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy". In: *Neuroimage* 46.3 (2009), pp. 726–738.

[Alo+19] Md Zahangir Alom et al. "A State-of-the-Art Survey on Deep Learning Theory and Architectures". In: *Electronics* 8.3 (2019). ISSN: 2079-9292. URL: https://www.mdpi.com/2079-9292/8/3/292.

[AMO09] Xabier Artaechevarria, Arrate Munoz-Barrutia, and Carlos Ortiz-de-Solórzano. "Combination strategies in multi-atlas image segmentation: application to brain MR data". In: *IEEE transactions on medical imaging* 28.8 (2009), pp. 1266–1277.

[ban] Varun bansal. *The Evolution of Deep Learning — Towards Data Science*. https://towardsdatascience.com/the-deep-history-of-deep-learning-3bebeb810fb2. (Accessed on 04/03/2021).

[Ben+07] Yoshua Bengio et al. "Greedy layer-wise training of deep networks". In: *Advances in neural information processing systems* 19 (2007), p. 153.

[Bro] Jason Brownlee. *A Gentle Introduction to Computer Vision in Deep Learning for Computer Vision*. https://machinelearningmastery.com/what-is-computer-vision. (Accessed on 04/08/2021).

[Bro92] Lisa Gottesfeld Brown. "A survey of image registration techniques". In: *ACM computing surveys (CSUR)* 24.4 (1992), pp. 325–376.

[Cou+11] Pierrick Coupé et al. "Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation". In: *NeuroImage* 54.2 (2011), pp. 940–954.

[Dou+15] Q. Dou et al. "Automatic cerebral microbleeds detection from MR images via Independent Subspace Analysis based hierarchical features". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2015, pp. 7933–7936. DOI: 10.1109/EMBC.2015.7320232.

[DT05] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[fra] fraunhofer. *High energy computed tomography*. https://www.iis.fraunhofer.de/en/ff/zfp/tech/hochenergie-computertomographie.html#3. (Accessed on 04/16/2021).

[Gee] GeeksforGeeks. *Activation functions in Neural Networks - GeeksforGeeks*. https://www.geeksforgeeks.org/activation-functions-neural-networks/. (Accessed on 04/08/2021).

[Gro] Maarten Grootendorst. *9 Distance Measures in Data Science — Towards Data Science*. https://towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa. (Accessed on 04/05/2021).

[Gup] Divam Gupta. *A Beginner's guide to Deep Learning based Semantic Segmentation using Keras — Divam Gupta*. https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html. (Accessed on 04/03/2021).

[Hec+06] Rolf A Heckemann et al. "Automatic anatomical brain MRI segmentation combining label propagation and decision fusion". In: *NeuroImage* 33.1 (2006), pp. 115–126.

[HHH09] Aapo Hyvrinen, Jarmo Hurri, and Patrick O. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.* 1st. Springer Publishing Company, Incorporated, 2009. ISBN: 1848824904.

[IDŞ16] Ali Işın, Cem Direkoğlu, and Melike Şah. "Review of MRI-based brain tumor image segmentation using deep learning methods". In: *Procedia Computer Science* 102 (2016), pp. 317–324.

[IS15] Juan Eugenio Iglesias and Mert R Sabuncu. "Multi-atlas segmentation of biomedical images: a survey". In: *Medical image analysis* 24.1 (2015), pp. 205–219.

[JC16] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[JYS12] Hongjun Jia, Pew-Thian Yap, and Dinggang Shen. "Iterative multi-atlas-based multi-image segmentation with tree-based registration". In: *NeuroImage* 59.1 (2012), pp. 422–430.

[Lam] Harshall Lamba. *Understanding Semantic Segmentation with UNET — by Harshall Lamba — Towards Data Science*. https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47. (Accessed on 04/06/2021).

[Lan+15] Zhenzhong Lan et al. *The Best of Both Worlds: Combining Data-independent and Data-driven Approaches for Action Recognition*. 2015. arXiv: 1505.04427 [cs.CV].

[Le+11] Quoc V Le et al. "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis". In: *CVPR 2011*. IEEE. 2011, pp. 3361–3368.

[Lia+13a] Shu Liao et al. "Automatic prostate MR image segmentation with sparse label propagation and domain-specific manifold regularization". In: *International Conference on Information Processing in Medical Imaging*. Springer. 2013, pp. 511–523.

[Lia+13b] Shu Liao et al. "Representation learning: a unified deep learning framework for automatic prostate MR segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2013, pp. 254–261.

[Lin20] Grace W Lindsay. "Convolutional neural networks as a model of the visual system: past, present, and future". In: *Journal of cognitive neuroscience* (2020), pp. 1–15.

[Low04] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[Mal89] Stephane G Mallat. "A theory for multiresolution signal decomposition: the wavelet representation". In: *IEEE transactions on pattern analysis and machine intelligence* 11.7 (1989), pp. 674–693.

[Nie] Michael Nielsen. *Neural networks and deep learning Chapter 1 Using neural nets to recognize handwritten digits*. `http://neuralnetworksanddeeplearning.com/chap1.html`. (Accessed on 04/03/2021).

[Nup] Er Nupur. *Image Segmentation Using Deep Learning: A Survey — by Er Nupur — The Startup — Medium*. `https://medium.com/swlh/image-segmentation-using-deep-learning-a-survey-e37e0f0a1489`. (Accessed on 04/04/2021).

[Ram+13] Karinne Ramirez-Amaro et al. "Enhancing human action recognition through spatio-temporal feature learning and semantic rules". In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE. 2013, pp. 456–461.

[Rez+19] Seyed Hamid Rezatofighi et al. "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression". In: *CoRR* abs/1902.09630 (2019). arXiv: `1902.09630`. URL: `http://arxiv.org/abs/1902.09630`.

[RHS11] Françcois Rousseau, Piotr A Habas, and Colin Studholme. "A supervised patch-based approach for human brain labeling". In: *IEEE transactions on medical imaging* 30.10 (2011), pp. 1852–1862.

[RRM04] Torsten Rohlfing, Daniel B Russakoff, and Calvin R Maurer. "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation". In: *IEEE transactions on medical imaging* 23.8 (2004), pp. 983–994.

[SA10] Neeraj Sharma and Lalit M Aggarwal. "Automated medical image segmentation techniques". In: *Journal of medical physics/Association of Medical Physicists of India* 35.1 (2010), p. 3.

[San+15] Gerard Sanroma et al. "A transversal approach for patch-based label fusion via matrix completion". In: *Medical image analysis* 24.1 (2015), pp. 135–148.

[Sli] Slicer. *3D Slicer image computing platform — 3D Slicer*. `https://www.slicer.org/`. (Accessed on 04/16/2021).

[TH15] Abdel Aziz Taha and Allan Hanbury. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool". In: *BMC medical imaging* 15.1 (2015), pp. 1–28.

[The06] Fabian Theis. "Towards a general independent subspace analysis". In: *Advances in Neural Information Processing Systems* 19 (2006), pp. 1361–1368.

[Wik] Wikipedia. *Connecting rod - Wikipedia*. `https://en.wikipedia.org/wiki/Connecting_rod`. (Accessed on 04/16/2021).

[YV15] Varduhi Yeghiazaryan and Irina Voiculescu. *An Overview of Current Evaluation Methods Used in Medical Image Segmentation*. Tech. rep. RR-15-08. Oxford, UK: Department of Computer Science, 2015, p. 22.

[Zaf+18] Paolo Zaffino et al. "Multi atlas based segmentation: should we prefer the best atlas group over the group of best atlases?" In: *Physics in Medicine & Biology* 63.12 (June 2018), 12NT01. DOI: `10.1088/1361-6560/aac712`. URL: `https://doi.org/10.1088/1361-6560/aac712`.

[Zha+12] Daoqiang Zhang et al. "Sparse Patch-Based Label Fusion for Multi-Atlas Segmentation". In: *Proceedings of the Second International Conference on Multimodal Brain Image Analysis*. MBIA'12. Nice, France: Springer-Verlag, 2012, pp. 94–102. ISBN: 9783642335297. DOI: `10.1007/978-3-642-33530-3_8`. URL: `https://doi.org/10.1007/978-3-642-33530-3_8`.

A List of Acronyms

ISA Independent Subspace Analysis
MAS Multi Atlas-based Segmentation
SAS Single Atlas-based Segmentation
3D Three Dimensional
2D Two Dimensional
CT Computer Tomography
MRI Magnetic Resonance Imaging
PET Positron Emission Tomography
CNN Convolutional Neural Network
LBP Local Binary Pattern
SVM Support Vector Machine
IoU Intersection Over Union
FPN Feature Pyramid Network
ANN Artificial Neural Network
LSTM Long short term memory
GRU Gated recurrent units
HOG Histograms of oriented gradients
FCN Fully Convolutional Network
ICA Independent Component Analysis
PCA Principal Component Analysis
SIFT Scale Invariant Feature Transform
CV Computer Vision
ReLU Rectifies Linear Unit
RCNN Region-Based Convolutional Neural Network
RAM Random Access Memory
LBT Local Binary texture
CUDA Compute Unified Device Architecture
VRAM Video RAM

ITK Insight Segmentation and Registration Toolkit

GPU Graphics Processing Unit

GB GigaByte

IoU Intersection over Union

GT Ground Truth

PM Predicted Mask

HD Hausdorff Distance

ASD Average Surface Distance

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie, dass ich die in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Passau, 19. April 2021

_____

Faizuddin Nasaruddin