Prostate MR Image Segmentation
September 2022

Supervisor - Frank C Langbein

Author - William James Glover

MSc Computing

Contents

0.1 Abstract

In the past decade autonomous computing mechanisms such as machine learning have grown in popularity, with a wide spectrum of professions and tasks looking to employ such techniques. One task that has seen the potential for automation is the imaging of prostate via MRI. Researchers have been looking at using these techniques to assist in clarifying the structure of the prostate to aid in the diagnoses of prostate cancers. As a result of this research there have been a myriad of methods considered and tested. This paper reviews the current landscape of these methods, selecting and ideal model in terms of efficacy and specificity in segmenting the prostate, as well as identifying the best data-set to use in future research and testing . Once the "ideal" model was selected we confirmed its efficacy by selecting what we deemed to be a potentially weak aspect, and attempting to refine the model in said aspect. This was accomplished by altering a structurally similar model and comparing its variation in Dice score **(DSC)** when using the selected data-set. Finally we applied these alterations to the "ideal" model and compare the differences seen from its base version to confirm the most ideal configuration. The findings were then placed within the limitations of the project allowing for discussion on potential future application of these findings, as well as the methods employed within this paper.

0.2 Introduction

In recent years Neural networks and Deep learning have seen a dramatic increase in interest and application, mainly thanks to increased GPU power and availability. This has allowed for model frameworks to be increasingly developed upon and documented, leading to a plethora of highly specialised models being available for certain tasks.

In the context of medical imaging, there has been significant increase not only in the efficiency of automated classification and segmentation task but also their efficacy, with some of the best models achieving results near to those of trained professionals (Isensee et al. 2020).

As this efficacy creeps up to fully replicating human ability these models can potentially be employed in aiding professionals in a clinical setting, providing faster and more accurate diagnosis for pathology's such as prostate cancer.

This paper attempts to look at the current state of this automation when applied to segmenting prostate MRI's, a organ with an internal structure that is notoriously hard to discern. This is vital in the diagnosis of cancers as significantly different prognosis's are heavily dependent on where pathology's reside. In this paper there will be an overview of multiple automated and semi-automated techniques that can produce segmented prostate images to be used to aid in cancer diagnosis's.

The literature around this will be reviewed and the models seen within will have their strong and weak points evaluated and compared with one another as well as their overall efficacy. From here the aim is to select a final model based on score, availability and considerations from previous literature, then seek to evaluate its weaknesses and potentially improve upon them. The result of this evaluation and testing is to provide the most exemplary model possible for future use and research.

As well as the model, research and testing requires a robust data-set that allows for sufficient comparison to real world scenarios and aids to validate the findings of the research utilising the model. This paper also takes this into account, selecting the most applicable data-set from the literature and validating it internally.

The selected model will be trailed with different configurations using the selected data-set, with the metrics obtained from these configurations being compared to one another to eventually isolate the most effective configuration.

A familiarity with machine learning advised but not a pre-requisite , python code knowledge would be a benefit although in terms of the model selection and application the paper should provide a a sufficiently encompassing overview.

0.3 Aims and objectives

Aims

- Evaluate the current automatic and semi-automatic segmentation methods, and their components.
- identify an ideal data-set to use for testing the efficacy of segmentation models, and validate the selected model with internal testing.
- Select the most ideal model from the literature.
- Assess and evaluate differing model configurations, selecting the ideal one
- Showcase selected models efficacy.

Objectives

- Provide the most applicable data-set for model use.
- Identify the most ideal model, and its strongest configuration.

0.4 Background

This section aims to provide an overview of the knowledge needed to understand the model application seen in this paper, as well as provide context for why a certain type of model was selected. Papers were sourced using Cardiff Universities internal library as well as Google Scholar.

0.4.1 MRI Overview

The prostate is an organ that plays a pivotal role in the male reproductive tract, producing and excreting the seminal plasma that combines with sperm to form semen (Barrett et al. 2019). Although the prostate plays an instrumental role in male anatomy and function it also presents a significantly high prevalence of cancerous pathology's, being responsible for 1 in 5 new cancer cases in men (Siegel et al. 2019) with particular prevalence shown in men over the age of 65 and those of african descent(Rawla 2019). Although typically it is slow to progress treatment can be costly and intrusive, with certain late stage variants essentially being incurable (Sumanasuriya and De Bono 2017). Therefore an early diagnosis plays a vital role in the management and treatment of such pathology's, MRI imaging has shown to be a favorable tool in achieving this early diagnosis (Bloch et al. 2008).

Due to MRI's ability to better distinguish between soft tissues when compared with other imaging techniques, they have become a mainstay in prostate imaging since their introduction the early 80s (Giganti et al. 2019). Most notably to aid in the detection of cancerous legions within the prostate, by allowing radiologists to categorise the lesions more accurately by employing the images along with classification techniques such as PIRADS (Weinreb et al. 2016).

In terms of structure the prostate overlaps with the Urethra and Seminal Gland and is composed of 4 main segments (Kumar and Majumder 1995). The Peripheral zone is the largest section and encompasses about 4/5ths of the entire prostate, with the remainig 5th being made up of the transition and Central zones (Kumar and Majumder 1995), as well as a structure that lies anterior to the Central/Transitional zone is the anterior fibromuscular stroma. (Bouyé et al. 2008).

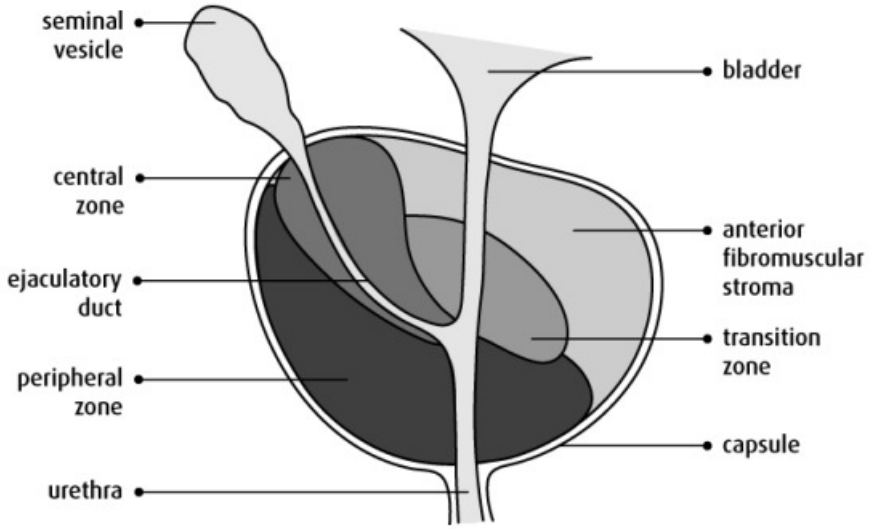

Figure 1: A sagittal plane image of the prostate indicating its position relative to the bladder, urethra and seminal vesicles.**(The prostate - Canadian Cancer Society.)**

When diagnosing prostate cancers MRI images are reviewed under the guidance of the PIRADS v2 grading system. This is a grading system allows clinicians to assess the danger of any tissue abnormalities present in the image, however there is a large discrepancy in how the abnormalities are diagnosed on either side of the transitional/central zone border (Barentsz et al. 2016). As a result one of the most prevalent issues seen in MRI interpretation is the ability to differentiate between the border of the Peripheral zone (PZ) and transitional/central zone (TZ/CZ) of the prostate. This is a key aspect that can impede the diagnosis accuracy and as a result the efficacy of the supplied treatment (Vargas et al. 2012). This is since tumour location in the TZ typically has a more favoured pathology and a higher recurrence free survival rate (Lee et al. 2014), making a clear distinction between the two vital in providing an accurate diagnosis and cancer grading.

When using MRI's to image the prostate, for segmentation purposes T2 weighted MRI's are employed over other variants of MRI as they have been shown to provide superior delineation of the prostates specific zones (Lovegrove et al. 2018). More recently Multi-parametric MRI's have also been utilised, these combine T2 Images with and Diffusion's weighted images and Dynamic Contrast-Enhanced images, this provides a better all round image including the superior segmentation of the T2 imaging with alongside imagine techniques typically used for tumour classification (Stabile et al. 2019).

Advents such as these in the imaging the prostate have been vital in allowing clinicians to accurately gauge the serverity of abnormalities in prostate tissues.However despite these improvements, the efficacy of these diagnosis is still shown to be quite limited with large heterogeneity in results seen across differing studies. (Oerther et al. 2021). This has resulted in more complex

techniques being developed with the goal of aiding clinicians in this task and reducing these inconsistent diagnosis's.

0.4.2 Automatic Computer Aided Segmentation

To rectify the aforementioned issues that arise from the image heterogeneity and variability when interpreting prostate MRI's, radiologists adopted computer aided vision to aid in the segmentation of lesions and zonal anatomy within the prostate.

A computer can aid in this process in a fully or semi-automatic fashion (SAMPSON 1999). Fully automatic will segment the image fully and present the final output for use without any human intervention. Whilst semi-automatic techniques are normally altered by classic methods before the final image is formed. As the clinicians role already is hindered by manually interpreting the MRI image, semi-automatic is not typically the preferred choice as it requires someone trained in MRI interpretation to provide input to the images resulting from the models.

Fully automatic models can be trained with varying levels of autonomy as well, typically in two distinct ways, supervised or unsupervised learning. In supervised learning the training data consist of input and output examples, in the case of the prostate the input would be a typical MRI, and the output would be the labelled variant of this (Nilsson 1996). This allows the algorithm to learn how the input data should be processed from a pre provided example but does require more pre-processing (Yves Chauvin and Rumelhart 2009).The alternative is unsupervised learning which only takes the input data and generalises characteristics from it that can be supplied to new test data. This will not be as tailored as supervised methods but does, to some degree, prevent the model tailoring too closely to the characteristics of a training set (Hilton et al. 1999).

Early automated models such as Atlas based segmentation trained their algorithms with previously labelled images, with structures in the inferred image to be matched to labelled structures from the training set, essentially allowing for automatic labelling of the new image.This is beneficial as it allows for multiple structures within the image to be identified as apposed to treating the prostate like a discrete area, with a yes or no answer, structures within the prostate can be identified and located thanks to this process. This method has been implemented for prostate segmentation with significant efficacy, achieving a dice score **(DSC)** of 0.89 in (Ayache and Al 2012), however this paper also signified newer methods at the time such as Voxel based classification outperformed Atlas models, although it is worth noting Voxel used manual whole prostate segmentation's while Atlas did not, potentially skewing the results .

Both models had issues segmenting out the peripheral zone as it presented as thin in the images, resulting in a less clear cut boundary. This could cause issues in real world applications of the model, as high image heterogeneity occurs across different clinical spaces. This paper also

utilises ADC MRI's along side the T2's, MP-MRI's are not always performed and a purely T2 image may perform significantly worse on models such as these, meaning their efficacy may be quite limited across differing clinical environments.

Another algorithm, the Random Walker Algorithm, functions off of edges and vertices, edge weights give probability a random walker will traverse the edge based on the image, this probability will be high if traversal is likely. Some nodes have ground truths with user supplied labels(Grady 2006) . On each pixel you compare the probability of reaching each label first, you then label the pixels with the outcome, with the highest probability route being what is followed (Li et al. 2013) . This model Showcased decent efficacy, achieving a DSC of 71.9. Although this came with the caveat of treating the image as purely discrete object, limiting the discerning of finer structures. The paper also showcased a high sensitivity to image noise which disrupted the models effectiveness . Furthermore despite using a significantly small data-set of 30 images, the model heavily taxed the hardware used taking a Long time and consuming a high amount of memory (Gao et al. 2018).

The final model reviewed was Super Pixel based segmentation. This model is more functionally distinct than the last two, it would group pixels together basing them on shared properties, this allowed borders to be easily distinguished, resulting in clear segmentation (Kłoczko 2017). The main draw of this is the reduction in computational power needed as not every pixel needs to be individually assessed (Achanta et al. 2012). However, as a result of its structure, image quality can directly affect its efficacy, but there is significant variation in the algorithms used and a quality one can override these issues to some extent(Ibrahim and El-Kenawy 2020).

The Majority of these models did show efficacy but had significant setbacks in terms of the level of automation, hardware time/power cost, and aspects of the segmentation's themselves. Most notable is the models lacking ability to deal with heterogeneous image sets either requiring pre-processing or pre-selection of the images used for training.

0.4.3 Machine Learning

Just feeding a model data will not necessarily result in a significant level of efficacy. Underfitting is where the input is not sufficient for the model to learn the necessary information, as a result it will not be able to delineate the correct information in the testing data (Cunningham and Delany 2021). Over-fitting is the opposing issue, where the data fed to the model is too extensive and as a result means the model can now delineate from that data set with high efficacy but will perform poorly on other data-sets as it essentially 'tuned' to detect features as they present within that data-set (Nichols et al. 2018).

(Mitchell 2017) describes Machine learning as an algorithm that can optimize itself from experience with new data, the sample 'Training data' that is fed to it. This presented an excellent

opportunity for image segmentation, as image variability could be accounted for by training on a wide selection of data, moving beyond simple pattern matching and helping to remedy the Over-fitting seen in earlier automated models .

Method Name	**Method Description**	**Pros**	**Cons**	**Prostate Application**
Markov Random Fields (Kato 2011)	An un-directed graphical model, the edges have no orientation associated with them, connected edges form cliches. That associated characteristic can be matched to based on probability.	Can represent data without worrying about directionality. Can clique slightly differing data together, so variations in images can be held together. (allows each pixel to belong to more than one class)	Difficult to generate and interpret data. Costly.	Fuzzy MRF used to segment the prostate, (Liu et al. 2009) DSC 0.98, significantly higher than K-clustering
Support vector machine (SVM) . (Cortes and Vapnik 1995) (Habes et al. 2013)	Classified features that differ are segregated. Support vector boundaries are aligned with this along the extremes of each characteristic. The hyperplane formed in between them is the aggregator, classification is based on the side of the characteristic it falls on.	Good for multidimensional characteristic recognition, highly applicable to 3D images in medical imaging. Memory Efficient. Works well with limited data sets.	Requires costly cross validation to convert to a probability. Impacted heavily by noise in images. Larger data sets can impact the efficacy.	(Habes et al. 2013) Didn't measure efficacy with DSC so not comparable to other papers.Performed well on a limited size 16 data-set, when analysed using Hausdorff distance.
Random Forests (Schroff et al. 2008) (Ghose et al. 2012)	Generates a forest of decision trees, containing nodes that represent characteristics of the image, taken from the training data.	Segments the image and its semantic regions. Prevents overfitting to some degree. Handles large data sets well, maintaining high dimensionality.	Better at classification then segmentation, as it cannot identify outside the training data. May over fit noisy data sets, issue with MP-MRI.	In (Ghose et al. 2012) the DSC ranges from 61 to 73, depending on the boundary being observed. Based on the PROMISE12 data-set.

Linear Regression (Wang 2019) (Yuan et al. 2012) (Liu et al. 2011)	Trained using a regression model that will map input values to output values, establishing their relationship, this can then be used to assess output on test data.	Far more simple than some other models and easier to interpret.	Has a more discrete nature than some other methods, as a result fluctuations or issues with the image will severely impact efficacy.	No papers with sole usage of algorithm could be found, usually used more a confirmation of efficacy such as in (Turkbey et al. 2013b).
K Clustering (k-means) (Dhanachandra et al. 2015) (Hamerly and Elkan 2003)	Finds the mean of a grouping of data and refines this process over and over to place the mean in the ideal position allowing a separation between the two sets, which you can classify on either side.	Can implement contrast stretching to improve low contrast images. Easy to implement.	Initial choice of the mean location will impact the efficacy, requiring extra computation power to implement a decent selection algorithm for this. Averaging data means it is less accurate in certain scenarios.	(Gao et al. 2012) Is on a CT scan but used K-mans and rectifies some of the issues by treating the mean as a dictionary that holds more specific values in it. Proved affective with a 91.3 DSC score. But not relatable to MRI.

Table 1: Table summarising early machine learning Algorithms, including and overview of each, along with its pros and cons and its efficacy when applied to prostate medical imaging. **DSC = Dice Score**

Whilst unsupervised learning seems like and ideal fit for medical imaging, especially with the added bonus of reducing over-fitting in application it makes difficult to look for certain features which is a key requirement when considering segmentation. The issue that arises from this is that mentioned above, under-fitting.

As these early machine learning models were more popular for classification, they were very effective at identifying structures based off the training data. However the heterogeneity and noisiness of prostate MRI images meant that the models struggled to perform segmentation task. As a result the vast majority of the methods outlined above used supervised training, with the exception K Clustering.

Though they fair-ed better in relation to to segmentation, the majority either were used

exclusively for classification or were applied to segmentation by altering a classification model. From what is observed in Table 1 dedicated segmentation algorithms are a rather niche when looking at early machine learning algorithms.

Across the models seen in table 1 the issues of over fitting and under-fitting can be seen in the Cons column, early deep learning algorithms where exceedingly dependent on the consistency of the data provided to them and would still over-fit. This combinations presented a key issue as limited access to variable data-sets meant that the models generated and tested would not necessarily translate to real world application.

0.4.4 Artificial Neural Networks (ANN's)

ANN's are named due to their 'mimicking' of a processes that is thought to occur in the human brain. Structured in such a way as to replicate the synaptic connections that lie within the brain, each node acts essentially acts as a synapse and can generate a weighting based on the information it receives. Depending on the weighting it can fire down edges (such as neurons in the brain) to a node specifically adept at processing information of that weighting (Gupta 2013).This allows for weightings and connections relating to individual characteristics to be adjusted based on the training data, meaning specific structures in the images can be be used to reinforce said weights without impacting weights that do not relate them.

These weights are stored in the hidden layer, this passes input information to the correct corresponding node in the output layer. Manually entering the weights of such a large structure would be extremely costly, to prevent this the network is designed in such a way that it can self-adjust the weightings based on data provided to it (Gupta 2013). This typically comes in the form of reinforcement learning, an offshoot of supervised learning where the network is only given critiques of its efficacy to refine said weightings (Sutton and Barto 2018).

Loss Function

This critique is performed using a loss function, this calculates the error seen from the models predictions and the outcome from the training data (Rosasco et al. 2004).

$$DICE = 2(A * B)/(A + B)$$

Dice loss has proven to be the most popular choice for medical segmentation in recent years as evidenced by its proliferation across the models reviewed in this paper and papers such as (Zhao et al. 2020), it works by evaluating the overlap of the automatically segmented image with a ground truth, where an exact match would result in a dice loss score of 1, with incomparable images scoring 0 (Zhao et al. 2020).

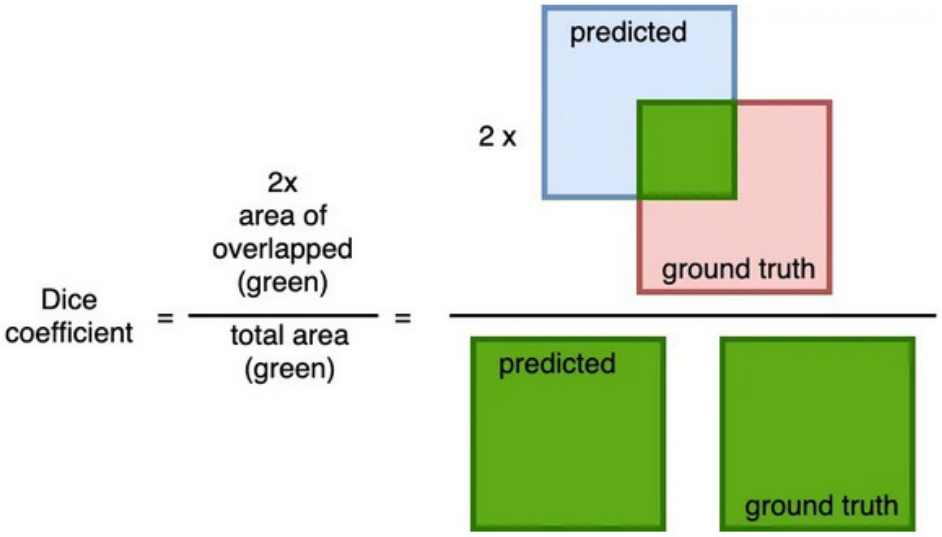

Figure 2: Dice Loss Function Summary (image segmentation - neural network probability output and loss function (example: dice loss). 2020)

This loss score is then typically applied using a method called gradient descent, where the weighting applied would result in the smallest loss (Specifically sum of the squared residuals between expected and observed) which optimises the architecture speeding it up significantly (Lecun et al. 1998).

Gradient Descent, Learning Rate and Momentum

Popular gradient descent algorithms are stochastic gradient descent algorithm where each layer's weighting is adjusted based on the loss experienced from each image as apposed to regular gradient descent that updates after the entire data set, this is costly particularly with larger data sets, but does prevent the weight overreaching the minima. The middle-ground solution to this is to use the Mini Batch version of this which selects a single training example from the batch for pass, this speeds up computation as there is a lot less to compute whilst maintaining the optimisations that occur from the batch method (Li et al. 2014b). This is due to the computational cost of processing the data from the whole data-set not being an issue, making it especially ideal for larger data sets (Mustapha et al. 2020).

The learning rate dictates how much the weights within the model can be updated with each pass, typically ranging from a value of 0 to 1. This means that a smaller learning rate will typically require more epochs or iterations to produce an accurate result, however this will prevent over reaching of the minima, due to the gradual descent (Lee et al. 2022).

In more recent models a feature called Momentum has been used to improve Gradient Descent algorithms by preventing ill fitting from singular poor examples in the data set. Using momentum the step is not only determined by the gradient of the current step but the accumulation of previous steps in the model(Alamri et al. 2022). However this can lead to overshooting of the minima so in recent years acceleration has been employed

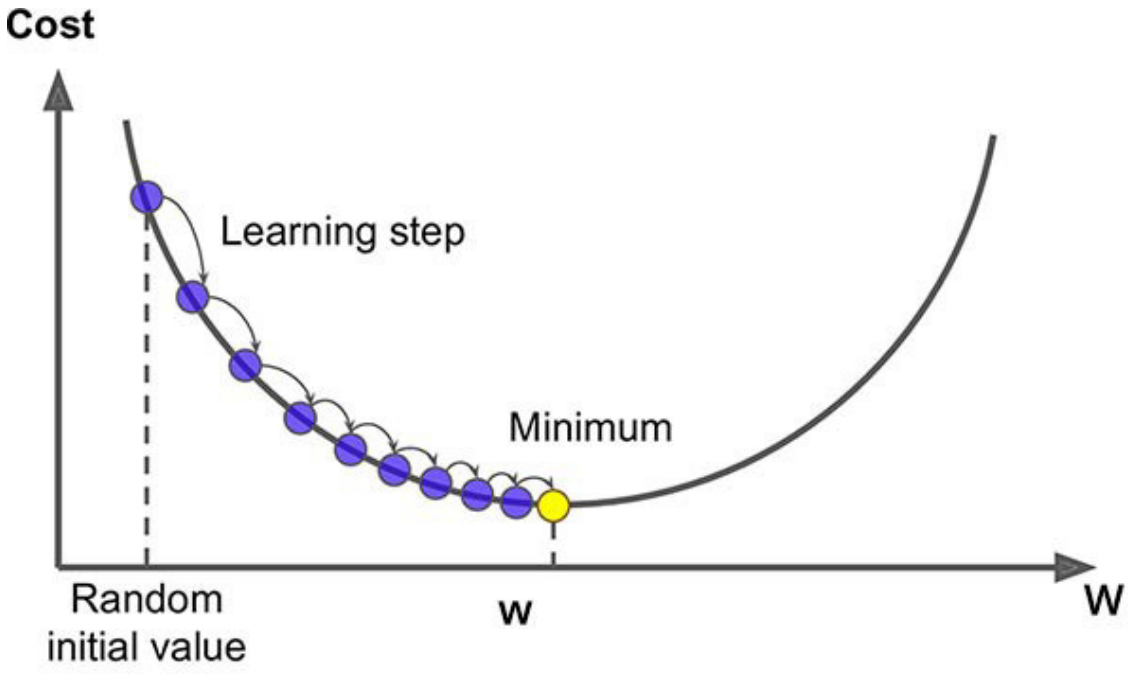

Figure 3: Overview of Gradient Descent (Kansal 2020)

as well. Which automatically adjusts the momentum rate based on the variance of the current step and its assumed step from the previous examples has been used such as Nesterov (Dozat 2016). ADAM is a gradient descent algorithm that has gained favour over the last few years as it allows application of Learning rates and momentum to individual parameters. This means it can more directly follow the ideal path to the minima for each parameter speeding up the model whilst retaining its accuracy(Dozat 2016).

There are typically two structures to ANN's, the first is a more linear approach referred to as feed-forward networks where each neuron can only pass information on to one other neuron. The alternative to this is a recurrent network which sends signals to multiple neurons (Gupta 2013). Due to their binary relationship for input and output, feed-forward networks are best used on distinct values that will output based on their conformity to specific weightings, this relationship also means that feed-forward networks do not integrate 'memory' in their neuron selection.

Recurrent Neural Networks (RNN's)

Recurrent neural networks utilise 'memory' as prior information collated from the neurons is still fed down and can interact with information in the current neuron influencing its output (there is also no weight difference between nodes in a layer), this allows for more variability in the data set without compromising the efficacy of the output (Abiodun et al. 2018). This has been applied to segmentation scenarios as seen in (Wang et al. 2019), but has been limited in its application to prostate segmentation normally playing a less significant role in more recently proposed models such as (Lu et al. 2020).However this results in an issue called the long term dependency problem. As more information builds on the node the RNN will struggle to learn new contextual information (Bengiot et al.

1993).

A specific mechanism to remedy this Is "Long Short-Term Memory" (LSTM), which has dedicated memory blocks that store contextual information from previous inputs. Here it takes in information from the current input, forgets irrelevant information whilst retaining useful data, then providing the current memory blocks context to the weighting (Sak et al. 2014).

Although recurrent networks allowed for context to be applied to specific weightings, and helps to resolve the issue of heterogeneity in images,one big draw back is that they still lacked contextual information between different structures in the images. This limits ANN's in the context of image segmentation as although they perform well at feature extraction, when aiming to image and discern the boundaries in the prostate identifying singular features is not enough . Whilst there have been attempt to rectify this such as feeding in images in sequential order in terms of the structure being segmented i.e. the PZ/CZ boundary, the efficacy seen suffers under image heterogeneity, and is extremely reliant on the appropriateness of the training data (Seo et al. 2020)

0.4.5 Deep Neural Networks (DNN's)

Deep learning is a sub-sect of Neural Networks in which the hidden layers far exceed the number present in ANN's/RNN's (Albawi et al. 2017) and the connections between each node and layer are more complex (Abiodun et al. 2018). Another altered aspect of Deep Neural Networks (DNN's) when compared to NN's is the abstraction that takes place at each level, higher layers pick out lower-level features and the deep layers pick out higher level features that will be significant to human observers.

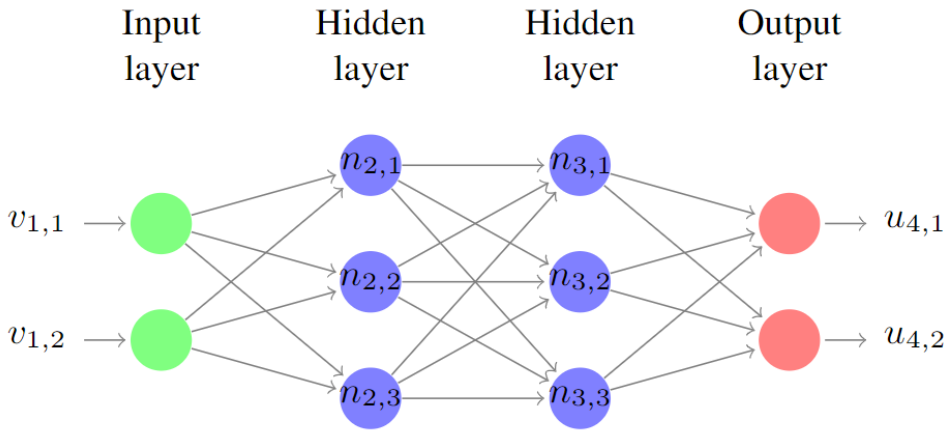

Figure 4: Depiction of a Neural Network with multiple hidden layers (Sun et al. 2019a)

(Emmert-Streib et al. 2020) argues that the adoption of deep learning techniques had been accelerated by the need to process increasingly large data-sets. As the architecture can hold multiple learning algorithms in its 'deep' structure it can correctly store information relating to heterogeneous characteristics seen throughout large Data sets. This has coincided with commercial GPU power seeing a significant increase in the 2010 allowing these architectures to be used by a far broader range of researchers (Sun et al. 2017).

This ability to process large heterogeneous data sets has allowed significant developments in image segmentation and classification in both medical imaging and the other fields. Despite its efficacy in these tasks, there are still issues associated with deep learning. The advent of increased GPU power certainly opened the door for deep learning algorithms but didn't't make them an entirely feasible option in terms of training and processing time. Both these aspects of DL models can be time and resource heavy, albeit significantly faster than when performed on CPU's (BUBER and DIRI 2018).

One issue that is exacerbated by the deep layering of DNN's is that of Internal Covariate Shift (ICS), where a change in the weighting input at each layer of the DNN occurs during training, this can slow the model as it needs to adapt to this new data (Ioffe 2015). This slows down the DNN training significantly as it now has to account for these changes at each layer. To remedy this batch normalisation has been employed on mini batches to standardise the inputs at each layer. This speeds up the training time and can reduce the impact off poor weight initiation (Ioffe 2015). This does have limited applicability on small batch sizes and the images forming the batch, which prevents it's use along side mini batch gradient descent limiting the speed and efficacy of the models (Lian and Liu 2019).

As well as the impact of ICS, the increasing depth of DNN's makes the gradient descent on each layer more reductive, as propagating down the levels (forward-propagation) results in overlapping computations occurring also increasing the computation time. Back-propagation was developed where the loss function on each layer is assessed from the output upwards. This removes the processing of redundant calculations in the chain rule led gradient descent (Li et al. 2014).

0.4.6 CNN's

CNN's are designed specifically for in-taking data in the form of images, they are particularly useful in this regard due to their preservation of the spatial relationships within the images (Selvikvåg Lundervold and Lundervold 2018). This is accomplished by the image going through many filtering layers that can discern spatial features over each section of the images grid like structure (Yamashita et al. 2018).

The composition used by the CNN facilitates this unique decomposition and separates it from more broad deep learning implementations. Generally, the 3 main layers are the Convolutional, pooling and fully connected layers.

When training a CNN, typically you have a forward stage where information if fed into the model, and then proceeding this a backwards stage, where back-propagated weight adjustment occurs.

These two processes both being completed for the entire training set is referred to as an epoch, studies may only use one epoch but typically this may result in under-fitting, instead a higher number will typically be used to optimise the training with iterations will be the number of batches fed to achieve one epoch (Khosa et al. 2020).

The initial Input Layer Is where pixel values of the base image are held. After this the

image enters the Convolutional layer, which is what gives the architecture its name, in this an array of numbers (Kernel) is applied over the grid separated input in the form of numbers referred to as a tensor. This kernel slides over the tensor composed image acting as a specific filter identifying features in each tensor (Selvikvåg Lundervold and Lundervold 2018). The kernel is typically found in a 3x3 format, but larger variants have been used in certain scenarios (Yamashita et al. 2018) and it typically shifts over one pixel at a time (This is referred to as a stride of 1). The kernel undergoes padding to ensure the width of the feature map matches the kernel which then allows the dot product generation of obtained probabilities from each tensor, yielding a single value (Yamashita et al. 2018).

Once this has been accomplished the tensor grid is down sampled in the Pooling Layer. This takes individual tensor layers and down samples them. When using Max Pooling this is accomplished by downsizing by a factor (stride) of 2, reducing the number of learn-able parameters whilst insuring the CNN a degree of image invariance (Selvikvåg Lundervold and Lundervold 2018). This allows for abstraction that after multiple layers will result in the identification of more discrete features (O'shea and Nash 2015). Although Max polling is the most common, average pooling is sometimes used where the average is taken across a kernel of Y*X (for a 2D image).This helps to remedy the issue of over-fitting as the strucutres picked out are now more high level meaning they will apply to more MRI cases .

Fully connected layers take the output from the convolutional and pooling layers and flattens them (converts them into and a 1d array of probabilities as opposed to a matrix structure). The probabilities can the be used for feature classification. (O'shea and Nash 2015).

Activation Functions

Activation functions are used following the convolutional layers and fully connected layers, they essentially ensure the input data for the next node in the hidden layers or for the overall CNN output. This is a non-linear activation function that approximates the input into that node into a standardised form. This needs to be nonlinear as the data can be complex enough that linear activation's functions will not be strong enough to process it, this is essential to allow the back propagation method mentioned before (Chigozie et al. 2018).

A popular choice for the convolutional layers is the rectified linear unit (ReLU), where the value is 0 if below 0 and then ranges from zero to the input number when positive. This has been widely adopted in DL networks in recent years due to its time efficiency (Krizhevsky et al. 2017) and diminishes the impact of the vanishing gradient problem experienced from back propagation as seen in (Talathi and Vartak 2016).

The vanishing gradient problem persists where weight between layers in the network become too similar, as the weight adjustment is based on the gradient between the weights the update to the weights will also be very small. As a result the weight cannot be updated sufficiently to reach the minimam, hindering the networks ability to learn (Rehmer and Kroll 2020).

However, this is typically only used within the hidden layers of the CNN, for the output layer other algorithms such as softmax (Fritscher et al. 2016) or sigmoid (Astono et al.

2020) are preferred, due to their output of 0 to 1 being easier to interpret (Goodfellow et al. 2016).

CNN's as described above can be designed to process input in multiple ways. Patched-wise CNN's take a sub-sect of the image being looked at and process it individually, once this sub-sect had been fully processed the CNN will move onto a neighbouring pixel and preform the same on the next patch . This is computationally costly as there is much repetition in the pixels encompassed in the patches, furthermore, the balance between image quality and the number of pooling layers can be hard to gauge and can impact the efficacy of the CNN itself (Gholamalinezhad and Khosravi 2021).

Early CNN architectures were used solely for classification. However, these models did introduce features that would be implemented into segmentation specific CNN's. They generally follow the patched wise structure outlined above, consisting of only an encoder, and processing selected patches one at a time. Below is an overview of three early notable models and components they introduced. They are not specific for medical segmentation, with the majority of them being used for broad classification.

AlexNet (Krizhevsky et al. 2017)

- Pioneer of using ReLUs as activation functions, early proof of efficacy.
- Overlapping Max pools were used to reduce the error rate of typical max pooling.
- Introduced dropout to prevent over-fitting, by dropping one neuron the back propagation path changes, resulting in more robust learnt weights.

VGGNet (Simonyan and Zisserman 2015)

- Replaced AlexNet large kernals in the first and second convolutional layers with a 3x3 kernal to achieve higher accuracy in the image.
- Achieved a new level of accuracy in classification at the time.
- Reduced the error from AlexNet significantly.

GoogleNet (Szegedy et al. 2015)

- Uses Global average pooling at the end as apposed to fully connected layers.
- Has a 1x1 kernal in the middle of the network to reduce the number of operations and as a result the processing time.
- Combine max pooling sizes 3x3, 5x5, and 1x1 on each layer, extracting different features at each level.
- This further reduced the error seen in VGG.

Despite these improvements in classification, performance degradation of the DNNs due to the "Vanishing Gradient" problem was an issue seen from the deeper structures, new models such as highway networks and the Residual Neural Network (ResNet) were developed to combat this degradation affect, accomplished by introducing features such as skip connections. ResNets connects layers further upstream to the current layer to provide information and context, it is built off of VGG and is not dissimilar from a recurrent network. This reduces the gradient diminishment and increases performance is deeper models (He et al. 2015).

One main caveat is that whilst there is literature to indicate the above models have proven effective when segmenting the prostate such as in (Chen et al. 2021) and (Abbasi et al. 2020) and models such as GoogleNet have open source builds online, these segmentation's relate to the whole prostate and tumours within, a hindrance employed by the fully connected layer. This layer struggles to provide output for real world images, fairing better with constructed images. When using real world images they are limited to a small number of structures with little heterogeneity (which makes it struggle with arease such as the PZ/CZ border)(Long et al. 2015),(Liu et al. 2021).. Another caveat of the fully connected layer is that it has a high computational cost as seen in (Dosovitskiy et al. 2016).

0.4.7 Fully Convoluted Networks (FCN's)

FCN's replace the fully connected layers in classic CNN's with 1x1 convolutional layers, this allows for pixel wise classification whilst retaining spatial information (Long et al. 2015). This is ideal for image segmentation as images different sizes can be fed in and effectively segmented and individual pixel segmentation can be combined with spatial knowledge allowing for specific delineation of boundaries based on their characteristics. This lack of fully connected layers also resulted in an improved computation time (Long et al. 2015).

Another addition the FCN provided was the use of up-sampling within the network, at each convolution layer the resulting heat map of features is up-sampled to produce a higher res image that can be aggregated with the other up-sampled heat maps resulting in the final image (Long et al. 2015).

Model	Key Features	Efficacy
PSNet (Tian et al. 2018)	Fine-tuned the last layers if the network as these were the layers that discerned high level features in the prostate.	Improved efficacy over the classic FCN with a DSC of 0.85. Using 2 data-sets including PROMISE12 , giving a good range of data, sourced form various hospitals, indicating a decent efficacy even under assumed heterogeneity in the data-set. This score was also further validated by the use of 5 fold cross validation to gather the results. However it is worth noting efficacy was not consistently high across all metrics used.
DenseNET (Huang et al. 2018)	Builds on the skip connections design from ResNet and applies connections between every layer, this allows less parameters to be considered at each step. Just adding a subsect of feature maps on each level. This reduces the computational costs of learning redundant feature maps and helps resolve the vanishing gradient problem.	This has been adapted in hybrid designs with other CNN architectures, but no basic DenseNet has been used for prostate segmentation. However, it does show significant efficacy in classification when compared with the base FCN, however this is not directly transferable to segmentation efficacy only implied.

DeepLab (Chen et al. 2017) (Chen et al. 2018) (Khan et al. 2019)	**DeepLab v1** Atrous Convolution – allows maintenance of spatial resolution despite the convolution layers, a form of "devolution" where the kernel is dilated out to a larger size but still only takes in the same number of parameters. This aides as the drop in resolution is never as severe as in typical models. **DeepLab v2** Atrous Spatial Pyramid Pooling (ASPP) – using multiple atrous layers with different sizes to identify features which are then processed in other branches to form an overall result. This allows for identical structures of similar sizes to be processes without altering the scale and distorting the image. **DeepLab v3/v3+** Multi grid method – As each consecutive layer is undertaken the convolution scale increase by a factor of 2 to maintain image size. And removed Random Field – Resulting in higher efficacy.	(Khan et al. 2019) showed increased efficacy over both PSNet and the base FCN when segmenting the peripheral and central zone of the prostate. However this was done against internally tested models of PSNet and FCN, as apposed to a comparison with the papers directly, as a result it is hard to determining the validity of these results, especially when considering the PSNet paper reported a much higher DSC score.
SegNet (Khan et al. 2020) (Badrinarayanan et al. 2017))	Introduces a multilevel decoder that mirrors the encoder that is typically akin to VGG or ResNet. The decoder up-scales its feature maps to the encoder feature map of the same level by taking indices from each encoder level.	In (Khan et al. 2020) segNet showed a significant improvement over base FCN, but was not as efficient as DeepLabV3+, although this was not used in a medical segmentation context.
PSP-Net (Zhao et al. 2017) (Malekijoo and Fadaeieslam 2019) (Yan et al. 2021)	Pyramid Pooling Module – Different pooling layers are used after each convolutional layer, then are up sampled and concatenated providing structural information within a larger context. This also prevents image size differences impacting the efficacy of the model. This paper also used contrast-limited adaptive histogram equalization to increase the contrast seen in the image.	In (Yan et al. 2021) showed a significant efficacy increase when compared with FCN when using the PROMISE12 data-set, although there no mention of the dice coefficient in the paper limiting the comparability to other models.

HD-Net (Jia et al. 2019) (Jia et al., 2019a)	Similar in structure to MSD net but with as 2d boundary decoder as opposed to a 3d one. This also follows a denseNet structure where skip connections are present throughout. The 2d decoder extracts volumes from the encoder. Like MSD the 3d encoder feeds information to the 3d decoder at each layer, but unlike MSD it does not pool information from the boundary decoder before the segmentation decoder portion is undertaken. It also employs convolultional pyramids as well as a residual refinement block which uses 1x1x3 convolutions to capture 2d features within the plane.	Producing a DSC of 0.9135 and scoring 90.34 in the PROMISE 12, it currently sits 3rd place. This structure is complex and although the paper outlines it there are no models present online for external use.
MSD-Net (J. Sathianathen et al. 2020)	Combines a 3D resNet encoder with a specific boundary decoder and segmentation decoder. The boundary decoder output is fused with the encoder output before entering the segmentation decoder.	Currently placing 1st in PROMISE12, but the paper is limited in description of the model due to it being built off HD-Net, and as with HD-Net online models are lacking in availability.

Table 2: Table presenting FCN's that provided notable new aspects to the base FCN model, that have either then become more common in newer models or showed a high efficacy when metrics were applied.

These Fully convectional models allowed for not only discernment of the whole prostate but also differentiation between the PZ and CZ using pixel classification to identify the boundary.Many of the models present here had features in place to combat the vanishing gradient issue, or refine the up-scaling seen in early CNN's, providing clear images that showcased the segmentation of the prostate.

However, despite the efficacy select models show here they are still very computationally costly and the up-scaled images are still not on par with the base MRI image, meaning that although the border would be segmented from the model, the rest of the image suffered. In medical image analysis this makes it far more difficult to effectively use systems such as PIRADS V2 for diagnosis purposes.

The main drawback of these models is their availability, few have open source code available to use such as MSD-Net and HD-Net, this issue is further amplified by their papers being relatively brief, with the aforementioned models being examples of this, with MSD-Net being particularly vague as it is a variant of HD-Net. This makes the models more difficult to replicate internally without compromising efficacy. As well as this other models such as Dense-Net have not been applied to prostate segmentation as noted by the literature and results of web searches.

0.4.8 U-Net

U-net, so called due its structure forming a "U" shape, is uniquely equipped when compared to other CNNs as it provides higher detailed output. It does this by essentially having a de-convolution process whereby the convoluted image is passed through an "de-convolution" tree. In this process it concatenates the input with the corresponding output feature map from the descending (convolution) side of the tree, essentially restoring its resolution at that level before it is passed further up the tree (Ronneberger et al. 2015). Decoders were used previously in models such as SegNet but the U-Net structure differs in that it passes over entire feature maps to the de-convolution tree as apposed to just the output from the pooling layers (Ronneberger et al. 2015). V-Net is shown later in this paper, providing a visual overview of the convolution an de-convolution tree, Figure 7.

This structure removes the need for fully connected levels, as at the end of the process the output size is equal to the input size, this lends itself particularly well to image segmentation as the as the localisation and distinction of borders is far superior when each low level convolution is classified within the higher level image, effectively allowing the low level aspects to improve accuracy whilst the high level can extract complex features (Liu et al., 2021).

Another big advantage of U-net is the ability to use data augmentation methods to artificially increase the data pool. By altering the images and running then through the CNN again, essentially you are providing other imaged for the neural network to learn from (Ronneberger et al. 2015). Although this may not be exclusive to the u-net build but just associated with the period where this became more common place, as data augmentation has been observed throughout this review on ealrier models.

Model	Key Features	Efficacy
USE-Net (Rundo et al. 2019)	Squeeze and Excitation blocks are added after each convolution layer. These blocks contain 1x1 convolutional layers and activation functions to aid in providing clear data in each scenario.	The USE-Net model showed an increase in efficacy over base U-net by a significant margin when they were both trained on the same data set. However this was further improved by training with multiple data-set, although it is worth noting base U-net also improved from this training strategy. A big draw of USE-Net is its reported efficacy under smaller data-sets when compared with U-Net, In terms of real world application this feature could allow the model to proliferate into may settings with limited data-set sizes.
ResU-Net (Xiangxiang et al. 2018)	Introduces skip connections within each layer such as in Res-Nets to prevent gradient degradation.	Achieved a DSC of 0.872, but does indicate traditional segmentation algorithms outperform it in certain metrics. Regardless shows and improvement over base U-Net.

Z-Net (Zhang et al. 2019)	The convolutional layers are arranged in a z shape, this essentially allows for skip connections to be implemented between each layer allowing spatial information to be preserved.	Showed that the architecture worked well with 2d resizing methods used to normalise the input data size.
Deeply supervised U-Net (Zhu et al. 2017)	The addition of supervised blocks deep within the hidden layer of U-Net preserves the gradients throughout so the vanishing gradient issue is reduced. And like google net they introduce smaller kernels in this case 1x1, reducing the number of parameters to consider and as a result reducing computational costs.	Showed very limited improvements over Classic U-net. The paper did mention it allowed for a quicker convergence time potentially speeding up the U-Net model whilst retaining similar efficacy but the paper only eluded to this and doesn't showcase this improvement.
Bridged U-Net (Chen et al. 2018)	The main caveat associated with U-Net is that is difficult to have lots of layer, as at a certain point the image quality starts to degrade not unlike what was seen in early FCN's. Certain papers have attempted to remedy this, an early method was (Chen et al. 2018b), still aimed at 2D images but utilising a double U-Net structure where contextual information at each layer in one of the encoder-decoder segments was also fed into the next level, essentially allowing for more levels to be introduced. This paper also introduced the Exponential Linear Unit, as a replacement for ReLu, in an attempt to resolve the vanishing gradient issue.	This showed an improvement in dice score, with the base U-net achieving 0.8715 and this version achieving 0.8996. However this structure does imply a longer time for a models to run and there is no considertation of this in the paper.
Cascaded U-Net (Zhu et al. 2018)	This also follows a bridge like structure, but differs in the fact one encoder-decoder pair is solely used for segmenting the whole prostate, whilst the second one focuses on segmenting the peripheral zone.	This showed efficacy improvements in both when compared with U-net in both whole gland and zonal segmentation. This paper did utilise an internal data-set for training and testing limiting the comparisons that Can be made to other models.

U-Net++ (Zhou et al. 2019)	This model introduces a decoder tree presenting from each encoder level, this allows for information to be reinforced as it moves from the encoder to the final decoder. This is termed the nested structure and provides skip connections between every node adjacent to one another.	This proved to be highly effective whe regarding dice scoring producing a DSC of 0.8974 again 0.7573. However there was only a smaller gain when IoU was used providing 92.55 against 91.03 but both showed a significant increase regardless. Both these also showed improvement when incorporating a Mask region based CNN. These results are biomedical segmentation but are limited in comparability as they were not undertaken on the prostate.
nnU-Net (Isensee et al. 2020), (Isensee et al. 2019)	nnU-Net is a widely applicable pipeline that can tailor itself for different medical imaging tasks, it does so by identifying variable parameters and altering the training process to pick out these task specific parameters.	The DSC for this was just below 0.8 for the PZ and just above 0.9 for the TZ, beating out specific pipelines to also come first in the PROMISE12 challenge at the time with a score of 89.65.
3D-Unet (Mooij et al. 2018) (Çiçek et al. 2016)	As the name implies performed 3d convolutions on a 3d image. This was the first 3D version of U-Net and allowed for fully or semi-automated segmentation of the images in question.	Whilst this showed and efficacy increase in the DSC compared to the 2D u-net when considering the peripheral zone (0.85 up form 0.82) it did fall at the transitional zone which saw a drop to 0.60 from 0.77. They identified issues with the thinness of the PZ boundary prohibiting it from being coded in on certain layers.
V-Net (Milletari et al. 2016)	At each stage a residual function is learned and applied to the output from the convolutional layers in a similar manner as ResNet. Other than that, it is similar to the base 3D model. Its most notable structural changes from U-Net, excluding the foray into 3D is the use of 2x2x2 convolutions in place of pooling layers which they argue increases efficiency of the model as there is no mapping required between pooling input and output during back propagation. One other difference from U-Net is the use of the slightly altered PReLu as opposed to ReLu.	Comparisons to base 3D U-net are limited from this paper, there is no talk of specific segments within the prostate, however using as dice based loss function it achieved a DSC of 0.869. This model also has multiple entries within the PROMISE12 challenge showcase a consistently decent efficacy

Table 3: Significant CNN variants built upon U-Net are noted above, with the key features they introduced and their efficacy.

The U-Net architecture consistently provided a way to produce high quality fully segmented prostate MRIs. As seen in its many offshoots which all showcased significant efficacy when compared with a manual segmentation with the majority achieving dice scores above 80. The introduction of data augmentation became common place, with extremes such as nnU-Net heavily relying on tailored augmentations to allow for improved efficacy.

CNN's have become the de-facto choice for medical image segmentation due to the reasons outlined in this report such as excellent output image quality. However, although models have now achieved dice scores in the early 90's (Isensee et al. 2020), they can still not rival the levels achieved by a trained individual. Furthermore the issue of heterogeneity has meant that normalisation and optimisation of the data-set has become more commonplace, this renders the models even less computationally efficient as outs of prepossessing must proceed the the training and testing of the model. Methods such as this having to be employed have resulted in more recent models efficiency decreasing as minor improvements in model performance requiring large developments in model design and training.

From the literature above it is clear that Fully Convoluted Networks and U-Net specific Networks present the most viable option for the selected model. Both show high scores across the board in terms of Dice score, and work to resolve the majority of problems seen in early automated model, the Vanishing Gradient Problem, under and over-fitting due to image heterogeneity and output image quality to name a few. They both also represent a turn to dedicated segmentation models as apposed to to altered classification models, particularly in the world of medical imaging. This next section showcases the rationale behind the chosen model and provides further evaluation of the models outlined above.

0.5 Selecting The Dataset

Data Set	Patients /n	Images/n	MRI	Pro's	Con's
Promise 12 (Litgens et al. 2014)	50	100	3T/1.5T	T2 weighted images Includes metrics for comparison. (Including leaderboard) Sourced from 4 places.	Small dataset
ProstateX (Hulsen 2019)	346	N/A	3T	Includes DWI and DCE as well as T2	Images chosen for lesion segmentation
Decathelon	N/A	48	N/A	ADC Included	Very small dataset
UKMMC	11	229	3T	WG Included	Small patient number
NCI-ISBI 2013	40	542	3T	Includes PZ, CA, labelled Masks	No whole gland labelled masks

Table 4: Overview of available data sets, the patient number, the number of images that comprises them and their pro's and con's, (Gillespie et al. 2020), (Khan et al. 2020)

Data-sets for prostate MRI's are quite limited , more so when considering the need for segmented versions of each image to accompany it, resulting in the small pool of the 5 data-sets above . The decathlon data-set and UKMMC both had limitations in the number of patients images present, with decathlon not providing the actual number and only 48 images implying a very small pool of patients, whilst UKMMC only provided 11. Although Deformations can be used to artificially inflate the data set there is potential when using a small data set of carrying over biases (Shorten and Khoshgoftaar 2019). It is worth noting this specific paper is regarding GANS and uses multiple dog breeds as the example for this bias, never the less this data-set is significantly small and was deemed to likely carry over biases even with augmentation.

NCI-ISBI 2013 has a similar sized patient pool to PROMISE12 but lacks the labelling of the whole gland meaning segmentation's of the CZ-TZ boundary would be all the model produced, negating the whole gland. Whilst not inherently bad for this papers direction, it was deemed preferential to have segmentation's of the whole gland. This was done to future-proof the findings of this paper in the scenario the models is compared down the line in a challenge such as PROMISE12 which requires full prostate segmentation.

Both ProstateX and PROMISE12 provide T2 images for segmentation purposes, and both have leader-boards of other models for direct comparison. Although ProstateX has a far larger pool of patients PROMISE12 has become the dominant place to test and compare prostate segmentation models. Furthermore ProstateX focuses more on the legion identification aspect of imaging, which is less applicable to this paper. AS a result the PROMISE12 data set was selected, with considerations of its small size. Luckily this is somewhat offset by the fact it sources the images from 4 differing locations providing some inbuilt heterogeneity amongst the test cases.

Intially the papers interest was surrounding MP-MRI's. However, there were notable issues with the data set being provided due to external factors. As a result Promise12 was selected as a training set due to the whole gland nature of it as well as its data-set size and diversity, even though it only supplies T2 images. However, as the gland segment delineations in the MP-MRI's largely hinge on the T2 images efficacy, improvements seen

in this paper should carry forward to MP-MRI's.

The Datset is provided at the link below along with the leaderboard containing previously attmepted bulds:

PROMISE12 Data-Set

0.6 Selecting a Model

0.6.1 Notable Exclusions

• Base Variants such as classic 2d U-Net and Classic FCN have been excluded as they have both seen significant improvement with new respective models and have acted as more of a baseline.

• PSP-Net (Yan et al. 2021) was excluded as it only used ROC curve metric in paper, and no model present in PROMISE12.

• UNet++ (Zhou et al. 2019) Was excluded as there was no data on performance in prostate segmentation from the papers found, furthermore it does not appear in PROMISE12, as a result there is little room to assess its effectiveness in regard to the prostate.

• Papers only presenting PZ and TZ specific Dice Scores were not considered as there was little comparative data formed from this metric as only a few papers such a (Rundo et al. 2019) used these metrics for the specific boundaries whereas whole gland was widespread, allowing for easy comparison.

0.6.2 Initial removals

PS-Net

Provides a good degree of metrics despite not featuring in the PROMISE12 challenge, which does allow for comparison of when regarding other models. However, these metrics do indicate that this model is lacking when compared with others most notably in the metric of DSC score where it achieved 0.85.

This is not a poor score, however there is also no model build available online as of September 2022. As a result it is difficult to see if this efficacy could be improved with minor tweaks to it's parameters or structure, and in terms of the efficacy shown there are other models more readily available that preform similarly to PS-Net, as a result it has been not been considered for the final model.

DeeplabV3+

Scores for both SegNet and DeepLab can be seen in Table Table 8.

Whilst this model does have code available online for implementation, it is limited in terms of prostate segmentation. With the most information regarding prostate segmentation efficacy coming from (Khan et al. 2020) which is mainly aimed at assessing SegNet it may be hard to tailor the model based off of this . Futhermore, In regards to PROMISE12, no entry of DeepLab has scored above 80 showcasing a significant lack of efficacy when compared with the higher scoring models.

SegNet

SegNet has an impeccable code write-up online, but this is not specific to the task of prostate segmentation. The models do score highly in whole gland when using Dice Scoring and very well in CG boundary, but a more objective comparison to other models proves difficult as it does not feature on PROMISE12, as a result comparisons would have to be made with a self reported metric rending this model a poor choice for this paper.

USE-Net

Even more limited than SegNet, this only reports DSC for the PZ and CG not the whole gland, and does not feature in PROMISE12. Furthermore there is no implementation online to run the model and compare.

Deeply Supervised U-Net

Similar to USE net with lack of reference to PROMISE12 and lack of code preventing further comparisons. Although its paper does provide a DSC for the whole gland where it preforms decently, unfortunately there is still not enough data to effectively compare without recreating the model from scratch and even with that undertaking the score is still not comparable to the highest performing models only showing slight improvement over base U-Net.

3D U-Net

Whilst variants of 3D U-Net do feature in the PROMISE12 challenge, and there are models present online, the literature surrounding its use with the prostate for segmentation purposes is very limited. Not only this but the score achieved in PROMISE12's best implementation of this model is not high ranking, scoring 0.8689, as a result it was removed at this stage in the selection process .

0.6.3 Comparison of Promise12 Data

Scores sourced from individual papers are lacking in a comparative sense, as scores can be based on averages or single cycles, or use differing metrics and or differing data-sets. As mentioned before the PROMISE12 data set and leader board were placed at the forefront of this project due to the plethora of comparative data it provides. As a result, this stage

in the selection process looked to compare only data sourced from the PROMISE12 leader-board. The best ranking version of each model was selected to reflect their full potential, and allow for comparison of each in their best light.

Name	Type	Haussdorf distance/mm	RVI %	ASD /mm	DSC %	PROMISE12
MSD-Net	FCN	3.55	0.83	1.116	92.9	91.9072
HD-Net	FCN	3.93	5.01	1.36	91.35	90.3441
ResU-Net	CNN	4.69	0.74	1.4	91.18	88.47750057
Z-Net	CNN	4.41	5.92	1.43	90.05	87.80677579
Bridged U-Net	CNN	5.58	5.75	1.59	89.96	86.5
Cascaded U-Net	CNN	4.28	2.08	1.4	91.23	89.3853
V-Net	CNN	5.28	4.83	1.75	89.36	86.47432464
nnU-Net	CNN	3.95	1.24	3.3	91.93	89.6507

Figure 5: Internal comparison, showcasing the models and their corresponding PR12 scores and metrics.

0.6.4 Selecting The Final Model

Z-Net and Bridged

All these models have been enhanced with pre-processing and model alterations, however despite their improved efficacy from their base counterparts they still do not score highly in any of the metrics when compared to other models. To add to this, although there are models present online, they are not specific to the prostate, as a result they were deemed unfit for selection and were negated from the chosen models.

Cascaded U-Net

This model performed similarly to ResU-Net but achieved a better total score on the PROMISE12 leader board, however in terms of DSC the two models show no significant difference. Due to the fact ResU-Net has the code available online in an open-source format whilst cascaded does not, ResU-Net appears to be the preferred choice for achieving this metric level.

ResU-Net

Although performing well and having open-source code available the details of the exact model used are not present, this means efficacy may not be achievable without a high degree of alteration that would have to be determined from the literature. Other variants of ResU-net may be sourced within the PROMISE12 challenge, but their efficacy would be lower. This hinders the validity of their inclusion within the highest achieving models. As a result, it was negated from the internal testing stage.

MSD and HD-Net

Both models are very similar with MSD being an extension of HD, although they are the highest achieving models of the PROMISE12 there is a lack of resources available online for them in terms of prostate imaging, both have papers describing the algorithms

used, but their structure in terms of using multiple encoders is rather complex and as a result would prove time consuming to reverse engineer. This complexity is furthered in the MSD model as the literature on it is lacking. As a result they have been removed from the final choice.

V-Net

Whilst not the strongest performer in terms of Dice Score, there is a wide variety of builds for this model available online in an open-source format and specified to imaging the prostate. This matched with its still moderately performing efficacy, with a level similar to ResU-Net, and the fact the tweaks to the model have been noted makes it a valid choice for a model to be built upon. However as an out of the box option it is still lacking due to its low efficacy.

nnU-Net

With a plethora of resources online including code and a broad description on how to implement said model accompanied with the high score on the PR12 challenge makes nnU-Net an ideal choice for the final model, it also works as a good bench mark for the other models as it comes pre-loaded with a PROMISE12 setup so the results can be gained locally with little to no issue if the inferred image pool is utilised.

0.7 Analysis of Chosen Model

0.7.1 nnU-net

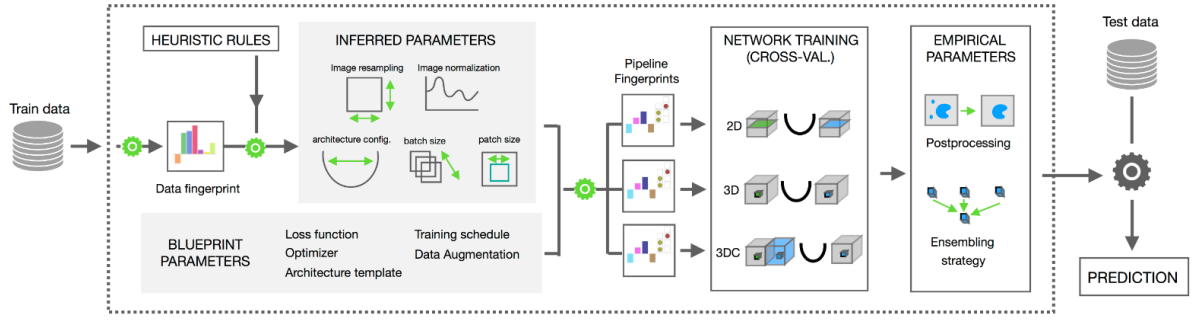

Figure 6: Visual overview of nnU-Net (Isensee et al. 2020)

As previously mentioned nnU-Net is not prostate specific, instead this CNN adapts to fit the data set applied to it. It does so by through a number of methods, the first of which is encompassed in the Data Fingerprinting stage.

Here the data is cropped to remove the majority of zero values, the paper notes that this has no effect on the ability to segment most data-set's, and they also note it reduces

computational costs by reducing the image size. Then based on these cropped images the model creates a fingerprint by capturing the modalities, intensity, spacial features and number of images, amongst other characteristics.

Once this fingerprint is achieved, Rule-based parameters are applied to the data-set allowing the configuration of the U-Net models, as well as choosing the "ideal" one. Model configuration also includes the levels of down-sampling to be applied. The ideal batch and patch size is also determined and more augmentations may be applied such as re-sampling or intensity normalisation(using z scoring).

The data is then fed into each model along with the fixed parameters mentioned below being applies to the data-set.

- Learning Rate = Initial, 0.0003, decayed throughout the training following the 'poly' learning rate.
- Loss Function = Dice and Cross Entropy
- Architecture Template = U-Net structure with skip connections instance normalization, leaky ReLU, deep super-vision (topology-adapted in inferred parameters).
- Data Augmentation = Rotations, scaling, Gaussian noise, Gaussian blur, brightness, contrast, simulation of low resolution, gamma correction and mirroring
- Optimizer = stochastic gradient descent with Nesterov momentum = 0.99
- Training Procedure = 1,000 epochs/250 mini-batches, foreground oversampling
- Inference Procedure = Sliding window with half-patch size overlap, Gaussian patch center weighting.

Once the U-Net architectures have processed the data, cross validation is used to assess the ideal choice for inference on a test set along side post processing to remove any false positives occurring in the image.

As visible in it's paper nnU-Net is heavily on augmentation and pre-processing techniques for its levels of efficacy, they propose in the paper that this pre-processing may be a prerequisite for surpassing trained human performance.

However, where the clear impact their pre-processing has on the efficacy is evidenced, the varying of hyper parameters is not,as a result this efficacy may be further heightened by implementing hyper parameter tweaks and structural changes that differ from the base nnU-Net model, this paper aims to investigate the effects of such hyper-parameter tweaks on the models efficacy in a bid to provide a more refined overview of nnU-Net and its efficacy, as well as ensure the model selected in this paper provides the best performance possible in any future use. At current nnU-Net uses AutoML to set its hyper parameters based on performance, but this time and resource heavy. It also only alters certain hyper-parameters such as batch size and normalization, here we are hoping to alter the direct structure as well to provide a more ideal configuration in terms of the static components such as activation functions.

In reference to the above, throughout the rest of the paper "Hyper parameters" is used to refer to structural changes such as the activation function that may not be typically classified as such.

nnU-Net supplies a Github repo containing its out of the box build along with extensive documentation on its usage at:

nnU-Net Implementation

0.8 Overview of Model Development

As mentioned above this project was undertaken by altering the structure of the neural network in terms of hyper-parameters as apposed to adding in more pre-processing aspects which nnU-Net already relies strongly on.

With nnU-Net being such a high scoring model with a build ready to test present online, it seems logical to build upon its structure for a further performance increase. However it's structure is far more complex that most variants of U-Net. Having 3 different U-Nets within, a 2D,3D and 3D cascaded form and a large focus on prepossessing means that changes to the the model could interact in an odd manner with said prepossessing/data augmentation. This may impede the efficacy of the results or reduce the validity of any conclusions drawn from them.

The nnU-Net models complexity also results in a longer training and testing periods, and with multiple model configurations this could lead to a overly long period of testing. This complex structure also results in configuration difficulties when attempting to run multiple models concurrently, due to consideration such as environmental parameters.

To negate these issues this paper proposes using a model of V-Net to quantify the effect of internal changers or "tweaks" on the U-Net based architectures efficacy when training and running inference on the PROMISE12 data-set and then attempt to apply them to nnU-Net, this offers a final display of nnU-Net's efficacy as well as confirming the most ideal configuration.

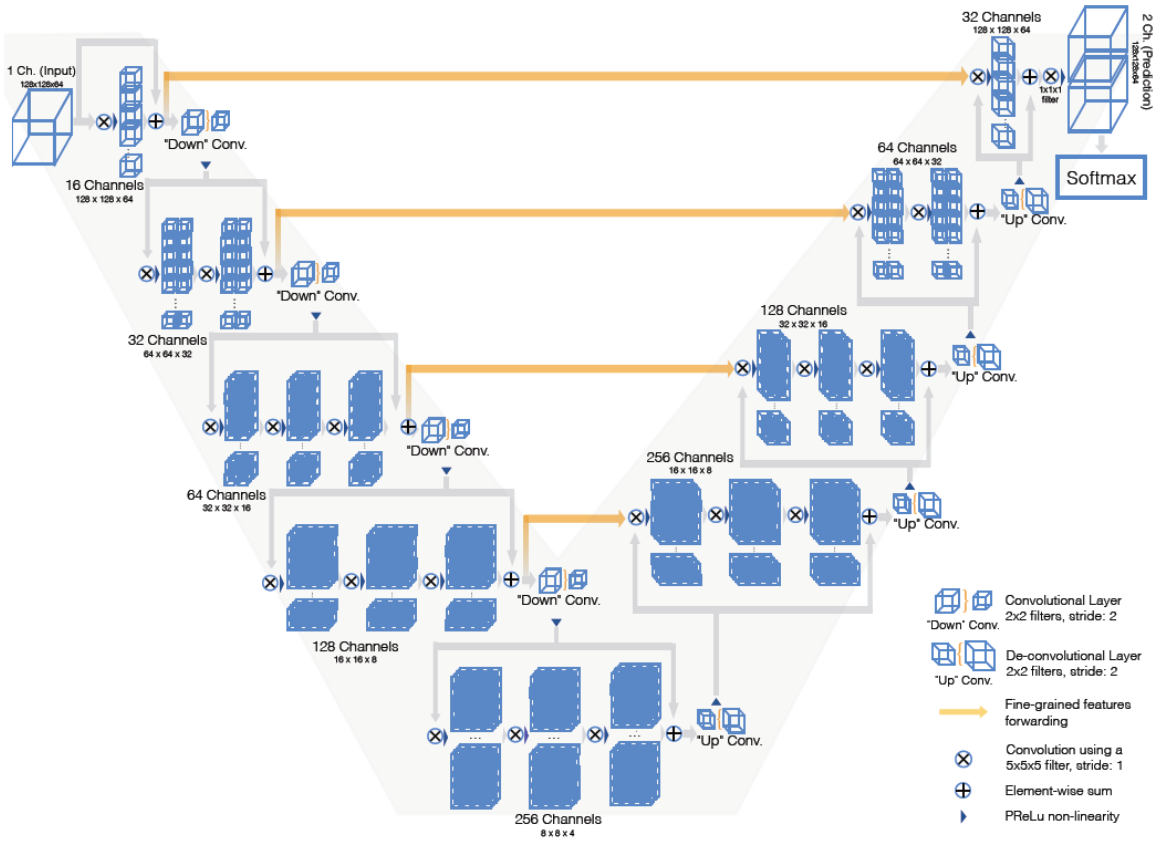

Figure 7: Visual overview of V-Net (Milletari et al. 2016)

V-Net is a much simpler model when compared with nnU-Net, essentially being an early 3D variant of U-Net. However for its simplicity it still scores decently, and its variants are still notable in the PROMISE12 challenge, furthermore the models comprising nnU-net are all offshoots of the U-Net/V-net architecture. This would suggest a maximal carryover of the results seen in V-Net when the configuration is applied to nnU-net,however this may not be the case due to the extended pre-processing that nnU-Net performs.

0.9 Initial Promise12 Results

0.9.1 Sourcing the Model and Computer

The base v-net model was to be used for the initial testing phase, due to its age many implementations that have been made available online, building a new model seemed redundant.

The model used in this paper was sourced from:

V-Net PyTorch Implementation

The base model here is relatively unchanged throughout the models being tested with only a few hyper-parameters altered and slight changes to the code added for tasks such as cross validation.

This implementation was based off of the initial V-Net paper but has been translated into PyTorch. All models in this paper were run on the Super-computing Wales **(SCW)** System utilising a Tesla V100-PCIE-16GB GPU on the Redhat 7.9 OS. The GPU was vital to allow for CUDA enabled modelling which greatly sped up the model, specifically this paper ran the models on CUDA 11.5.

Whilst there is still an element of pre-processing, there is far less than in the nnU-Net configuration. This allows the results drawn from this provide the as close to a raw comparison of the hyper-parameters as possible, whilst still relating to the structure of nnU-Net. Any variance seen in the nnU-Net implementation will be indicative of interaction from non discrete parameters within the model.

The build matches the V-Net architecture described in the paper with the exception of ELU being used in place of PReLU, however this was changed to fit the paper before any tests were performed.

A few parameters did have to be assumed as they were not explicit within the, most notably was the Gradient Descent algorithm, SGD-Mini-Batch was chosen as the batch size was mentioned indicating it was not pure SGD, the other popular choice in models is ADAM but with this becoming popular more recently the assumption was made base V-Net used SGD.

There was also no mention of epochs in the paper, only 30k iterations the mentioned batch size of 2, as a result the model we used was reworked to follow this structure, this was done to ensure comparisons could be made between ours and theirs to indicate the variance in the output formed from their augmentation within the paper.

Gamma was also present within the model, and was negated for the initial test to fall in line with the paper as a result it is not mentioned throughout this paper.

0.9.2 Cross validation the train set

Another method utilised in this paper that differs from the original V-net paper is the inclusion of cross validation. Here 5 fold cross validation was used to assess the efficacy of the data set.

Typically cross fold cross validation is a method used to ensure the efficacy shown from a model is not due to a synergistic effect between the training and testing set. In this method the data-set is split into smaller batches which each take a turn being the testing set whilst the rest of the data-set trains said model.

5 fold cross validation is already present in nnU-Net but is used during the model training to allow for optimal configurations when running inference. Here it is identifying anomalies in the data set, this was not undertaken to exclude images in future but to better identify how the data-set should be configured and utilised when training and testing further models and comparing to externally tested models, as well as being a potential indicator of any outliers seen in later tests performed in this paper.

For instance although this paper is internally comparing models, the best data-set configuration would allow for a superior comparison to other models present in the literature, as a result ensuring that weakly performing training images are not clustered together not only allows for a better performing model but one that displays efficacy similar to that seen with a larger training set where edge cases produce a lesser performance impact, which would better reflect an ideal real world scenario.

Similarly the same can be applied to over-performing data-set distributions, which in a small data-set such as this can produce equally poor results in terms of generalist performance of the model.

As the base V-Net model did not feature any form of cross validation it was implemented using the **SKlearn** Python package more specifically the kfold component. This was used to form the data set distribution and the training code was altered to loop over these distributions and populate the data-set queue based on the fold the model was currently training on.

```
def trainTestSplit(self,imgs, labs, img_keys, kfold):
    print("Saving Cross Validation Distribution")

    imagesTr ={}
    labelsTr ={}
    imagesTs ={}
    labelsTs ={}


    #splitting the keys based on the kfold passed in
    for fold, values in enumerate(kfold.split(img_keys)):


        if fold !=0:
            continue

        print(values)

        training_keys =[]
        testing_keys =[]

        training_indexs =values[0]
        testing_indexs =values[1]

        for index in training_indexs:
            training_keys.append(img_keys[index])


        for index in testing_indexs:
            testing_keys.append(img_keys[index])



        train_images ={i: imgs[i] for i in img_keys if i in training_keys}
        train_labels ={i+'_segmentation': labs[i+'_segmentation'] for i in img_keys if
                                                i in training_keys}
        test_images ={i: imgs[i] for i in img_keys if i not in training_keys}
        test_labels ={i+'_segmentation': labs[i+'_segmentation'] for i in img_keys if
                                                i in testing_keys}


        imagesTr[fold] =train_images
        labelsTr[fold] =train_labels
        imagesTs[fold] =test_images
        labelsTs[fold] =test_labels

    all_folds =[imagesTr, labelsTr, imagesTs, labelsTs]

    with open('/nfshome/store03/users/c.c1631387/V-Net/KFoldDistribution.json', 'wb')
                                            as fp:
        pickle.dump(all_folds, fp)
```

Figure 8: Code Snippet showcasing how the cross validation data-sets were produced

0.9.3 Initial Results

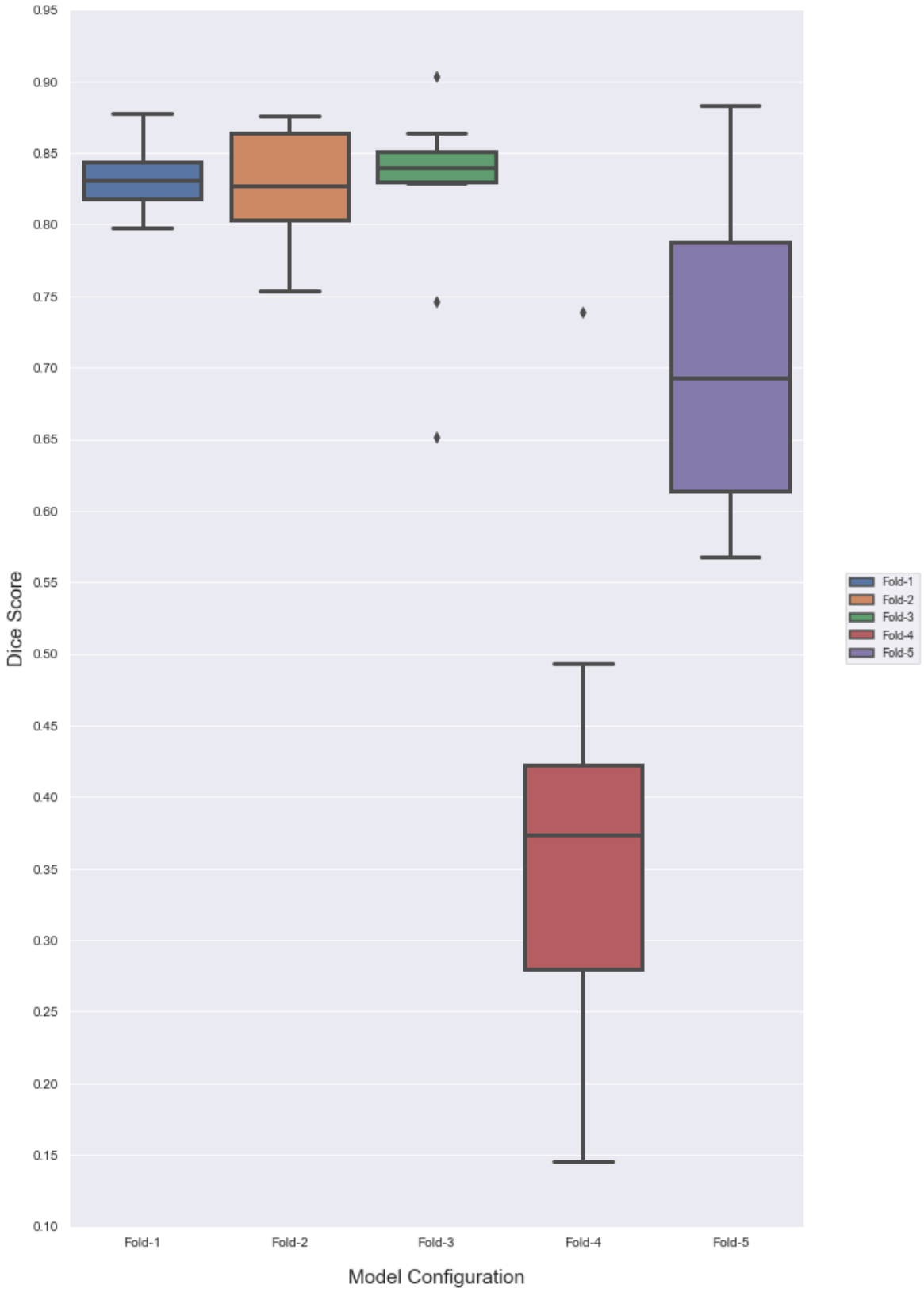

Figure 9: Dice Score Distribution across folds in cross-fold validation of PROMISE12 data-set

FOLD	Test Img 1	Test Img 2	Test Img 3	Test Img 4	Test Img 5	Test Img 6	Test Img 7	Test Img 8	Test Img 9	Test Img 10
1	2	10	13	22	24	28	29	33	34	35
2	5	8	27	32	37	41	42	44	45	47
3	3	4	6	9	14	16	20	21	25	31
4	12	15	17	18	19	26	38	43	46	49
5	0	1	7	11	23	30	36	39	40	48

Table 5: Testing images used in each fold of the 5 fold cross validation, all images omitted from each row were used to train the model for the specific training run.

As presented above the yield from the cross validation showed relatively uniform results with the exception of fold 4. In this fold excluding Case18 which scored a 0.739 DSC, however the majority of cases showcased very low scores below 0.5 DSC. This may indicate a poor performing data configuration for either the training or testing, however this seems unlikely as the fold where randomly selected in terms of configuration. The model parameters remained static during this training which implies an error occurred on the machine, however this is unlikely. Although this is somewhat re-enforced as the fold saw continuously low dice scores throughout, with the relatively high scoring case18 occurred at the end of the evaluation run. This may possibly indicate a bottleneck on the machine running the model especially as it is not seen throughout the rest of the paper.The most probable cause would be that a section of the model was altered accidentally before resuming a checkpoint in its run, however res-testing would be required to validate the claims.

The large range of Dice scores present in this fold may also be reinforce the claims of weak image configuration or an unwanted change to a hyper-parameter. In an attempt to gain insight into this issue case 49 from this set was included in the main model testing phase, to validate it being a weak performer.

Re-testing was not undertaken due to time constraints and as internal evaluation was not dependent on the values seen here, this was undertaken for future evaluation against external models. This anomaly did not continue throughout the next stage of testing, further removing the need for extended retesting.

The remaining folds fared significantly better as evidenced in Table 9, with fold 1 and 2 both having ranges well below 0.15, and showing consistently high efficacy across the evaluation cases. Fold 3 and 5 both had high scoring cases producing the 2 highest scores of 0.903 and 0.882 respectively. However the Dice Score ranges seen over cases in both these Folds was far greater than those of fold 1 and 2 indicating the cases here were not as much an ideal match to this models configuration as those in the other folds and that those data-sets are possibly more heterogeneous than 1 and 2, full fold scores can be found in the appendix at Table 16.

This had little impact on fold 3's average value, as it scored on par with fold 1 and 2 in this regard. Fold 5 fared worse from this variability with the average DSC coming in significantly lower, although it is worth noting fold 5 showcased less outliers when compared with fold 3.

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
0.8318	0.82824	0.82034	0.37601	0.70801

Table 6: Average Dice Score from each fold in the cross-fold validation of the PROMISE12 data-set

0.10 Structural Alterations/Hyper-parameter tuning

As outlined earlier in this paper there are multiple hyper-parameters that can be altered within a neural network. In this case the hyper-parameters that underwent these alteration's were selected based those found in the high performing models outlined in the literary review section of this paper.

Whilst there are many hyper-parameters and many options within each parameter, the literature can only provide so much information as to the efficacy of certain combinations and parameters.For this paper time constraints did result in a limited number of times models could be run. As a result hyper-parameter choices were based off of the existing literature and were limited in terms of internal testing, resulting in only a select number of models available to be run. To deal with this caveat the paper only considers a limited number of hyper-parameters and their variants to be selected as testing candidates.

Model	Loss Function	Gradient Descent	Learning Rate	Batch Size	Weight Decay	Activation function
U-Net	Cross Entropy	SGD	-	-	-	ReLU
3D U-Net	Weighted Softmax	SGD/ADAM	0.00001	1	-	ReLU
V-Net	Dice	SGD	0.0001 decreases by power of 1 each 25k iterations	2	-	PReLU
ResU-Net	Dice	ADAM	-	32	-	ReLU
Z-Net	Dice	ADAM	0.001	8	-	ReLU
Bridged U-Net	Cos-Dice	ADAM	0.001	24	-	Elu/ReLU
Cascaded U-Net	Dice	-	0.0001	-	-	-
U-Net++	Dice	-	0.0003	-	-	-
U-Net3+	-	-	-	-	-	ReLU
nnU-Net	Multiple	ADAM	0.01 with poly learning rate Applied	Variable	0.00003	ReLU
3D Unet^2	Hybrid	ADAM	0.0003	Variable	$10^{-}5$	-
USE	L-loss	SGD	0.01 X 0.2 every 20 epochs	4	5x$10^{-}4$	-

Table 7: Hyper Parameters present in evaluated models

Batch Size

Batch Size in the initial V-Net paper was set to 2, in more recent years smaller batch sizes such as this have been less favoured as although it converges quickly can result in a noisy image, so is typically paired with a low learning rate to remedy this (Kandel and Castelli 2020). However, a benefit it does have is having a smaller computing footprint due to the faster conversion. nnU-Net also uses a batch size of 2 when training a model to abide by GPU memory constraints (MIC-DKFZ/nnUNet. 2021). Due to the fact that a change in the batch size would need to coincide with a suitable change to the learning rate in V-Net and would require many internal models to perfect due to great variability within the literature, batch size remained at 2 to fit time constraints. Furthermore nnU-Nets batch size is altered when running inferred data as it pulls the most ideal model from the 5 fold validation performed so manually setting it here has very little carry over.

Momentum

Momentum in both nnU-Net and the base V-Net model was 0.99, on-top of this other papers that were analysed in the literary portion of this paper lacked information on the momentum or didn't include it. As a result momentum has been omitted from the final pool of tweak-able hyper-parameters, as without examples supporting other ideal values changing the momentum risks disrupting the finding in the model, and as it is the same in both models momentum instead remains a fixed parameter. As there will be may other factors affecting the model keeping this the same anchors both models in a similar base setup, which should increase the degree to which the V-net tweaks efficacy carries over to nnU-Net.

Weight Decay

Weight Decay information was significantly lacking when reviewing other high scoring models either due to it not being included or not mentioned in their papers. V-Net lacked this information whilst the Github build of nnU-Net showcased a value of 3E-1 (MIC-DKFZ/nnUNet. 2021), as a result this was selected as the weight decay in this papers V-Net build to maintain consistent values for the parameters not being altered in both models, and an in depth explanation of this parameter was not included in the paper.

Loss Function

As mentioned previously Dice Loss has proven to be the most popular choice across the reviewed literature and this is echoed in the vast majority of CNN's investigated, this trend has become more extreme in recent years,and whilst nnU-Net has applied multiple loss functions, to meet the time constraints imposed on this model Dice was selected as a constant parameter for the V-Net builds.

Gradient Descent and Learning rate

Of the papers and models reviewed every single one utilised a gradient descent algorithm of SGD or ADAM. In the majority of ADAM cases this Descent algorithm coincided with a Learning rate of 0.001, as a result for use in this paper ADAM was partnered with this learning rate to provide the most ideal setup for producing effective results.

SGD was more variable across the papers ranging from a set 0.00001 value to larger values which also decayed by various factors. However more modern papers such as nnU-Net use a seemingly retain larger starting rates, with nnU-Net starting a 0.01 when utilising SGD, and reducing this further with the ply learning rate functione .This paper chose to follow the aforementioned 0.00001 value, to differentiate itself from the base nnU-Net model allowing for comparison down the line if an SGD model were to be selected. Furthermore as the V-Net build used in this paper is more simple and doesn't hold complex parameters such as acceleration it was argued that utilising Learning rates seen similar to older models using SGD would produce effective results even without the inclusion of newer feature and pre-processing.

Activation Function

The most variety seen amongst the hyper-Parameters was in the activation functions, with 4 variants of the ReLu function. One of these Leaky ReLu is used in nnU-Net, whereas the base V-Net model utilises PReLu. PReLu itself appears more dated as newer models tend towards Leaky ReLu/eLu/ReLu, with ReLu in particular appearing an a few of the more recent models. Elu only appears in one paper as an optional choice alongside ReLu but is also present in the base code of the V-Net model used throughout this paper implying efficacy over PReLu.

All 4 of these functions were selected to be modelled along side differing Gradient descents and Learning rates .

0.10.1 Generating models to Test

Initially grid search was to be employed to allows for ideal model selection, it is available via the **sklearn** package used earlier in the paper for cross validation.

However, due to the lack of variety in the hyper parameter setup a simple for loop was implemented to alter the parameters once each model had run.

Activation Function	Gradient Descent	Learning rate
ELU	ADAM	0.001
ELU	SGD	0.0001
ReLU	ADAM	0.001
ReLU	SGD	0.0001
PReLU	ADAM	0.001
PReLU	SGD	0.0001
LReLU	ADAM	0.001
LReLU	SGD	0.0001

Table 8: Final selection of hyper parameter setups in tested models

0.10.2 Model Assessment

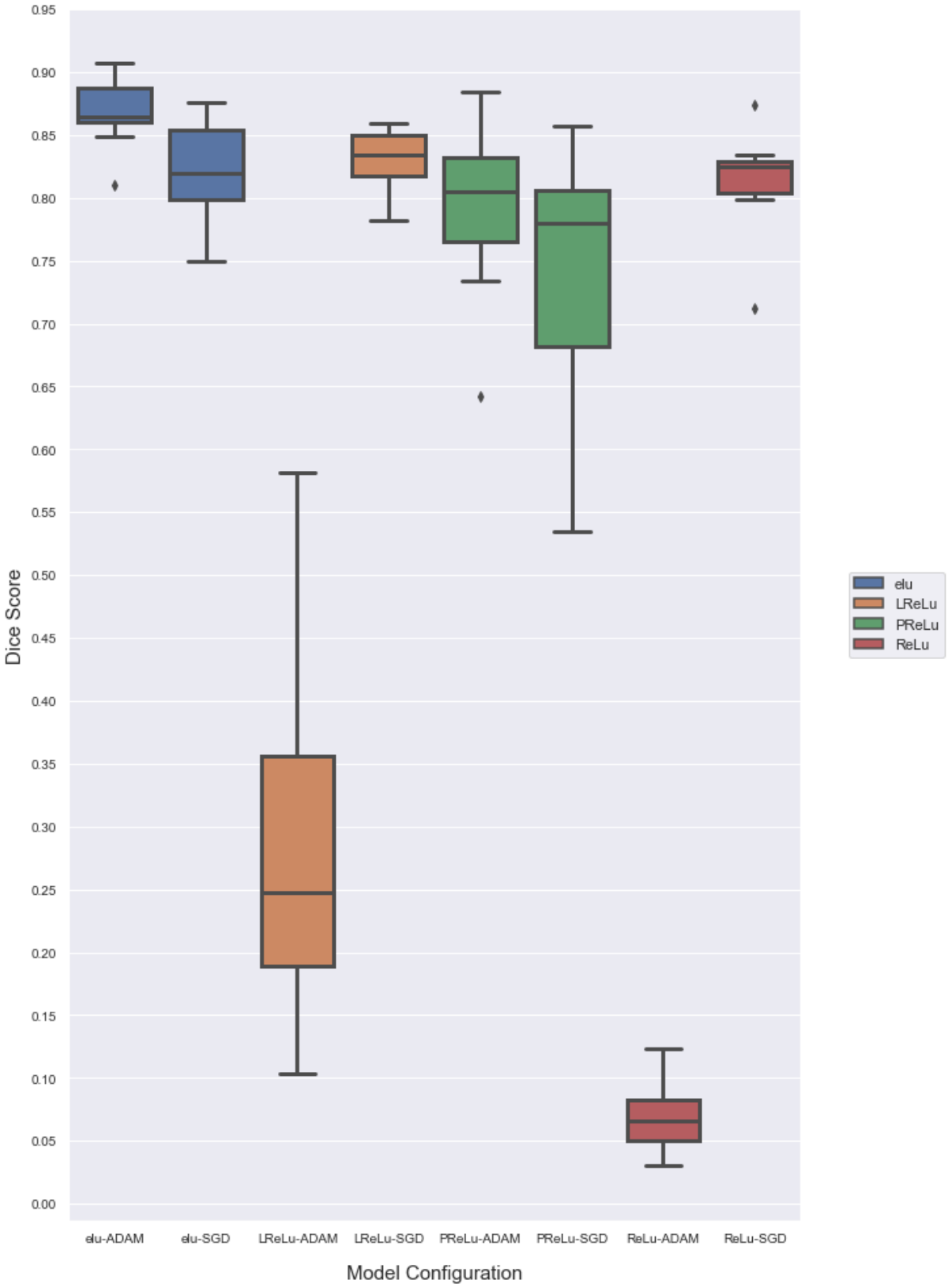

Figure 10: Range of Dice Scores achieved with each model presented as box-plots

The variants of the V-net model produced significantly differing results in terms of efficacy. The 3 highest scoring models in terms of average score were Elu-ADAM, LReLu-SGD and Elu-SGD all achieving an average Dice score over 0.82. ReLu-SGD fell just short of the top 3 producing a marginally worse result than Elu-SGD achieving an average score of 0.81313 awith the latter producing a result of 0.82084, as well as this it produced more outliers making it a less consistent choice than Elu-SGD, as a result it was not placed into the second round of testing.

Certain models that fared well under the use of the SGD optimiser,saw signifcant drops in efficacy when utilising ADAM in its place. This is most evident in the results of LReLu and ReLu, both scored well when placed with SGD achieving Dice scores in the low 80's but their ADAM counterparts did not follow this trend. LReLu saw a large range of values, indicating it may not generalise structures well when partnered with ADAM, suffering in more unique cases. ReLu however saw the inverse issue, where it was unable to identify structures all together, gaining low scores across the board with no DSC above 0.15 being achieved. To ensure these low scoring models where not caused by a faulty model, the model parameters where reset and ran on the same target set again.

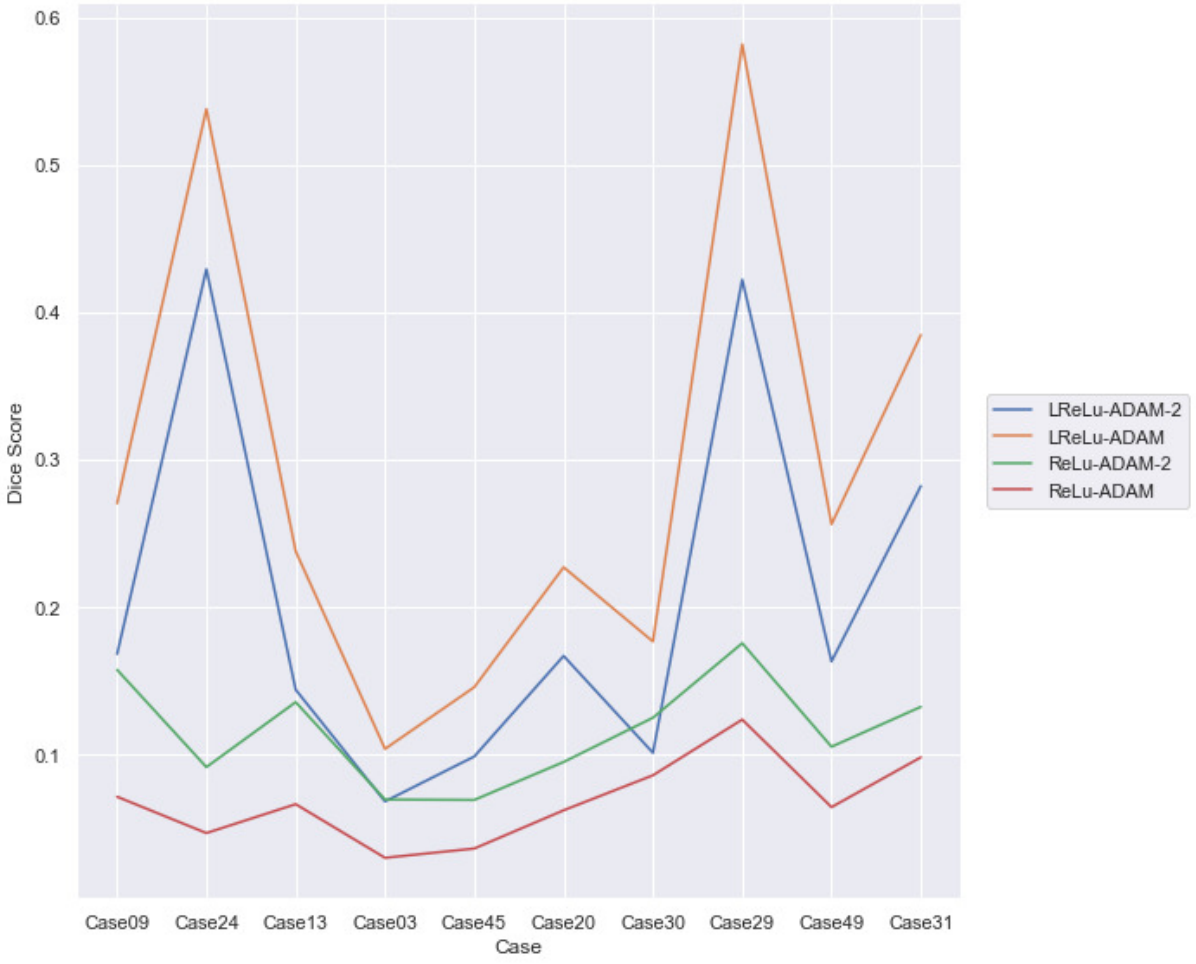

Figure 11: Scatter plot showcasing the Dice Score of the low scoring models attempts on the initial data-set configuration

LRelu-ADAM	Relu-ADAM	LRelu-ADAM-2	Relu-ADAM-2
0.292222798	0.068663962	0.20445402	0.115774922

Table 9: Mean Dice Scores Achieved from the poor performing models.

The second test of these models reinforced the findings with similar trends seen in both LRelU and ReLu, with only slight deviations seen within the score of each test case. LReLu fared slightly worse on the second attempt whereas ReLu perfomed slightly better, however this improvement did not place it on par with the rest of the examined models.

For the most part the initial high scoring models showed differing efficacy across the cases tested, however case 03 showed a significantly lower dice scores on all 3 models regardless of the optimiser, well case 31 performed poorly on the SGD builds . This could be indicate of more heterogeneous samples in the data set or potentially a structure present in certain samples that the V-Net model struggles with.

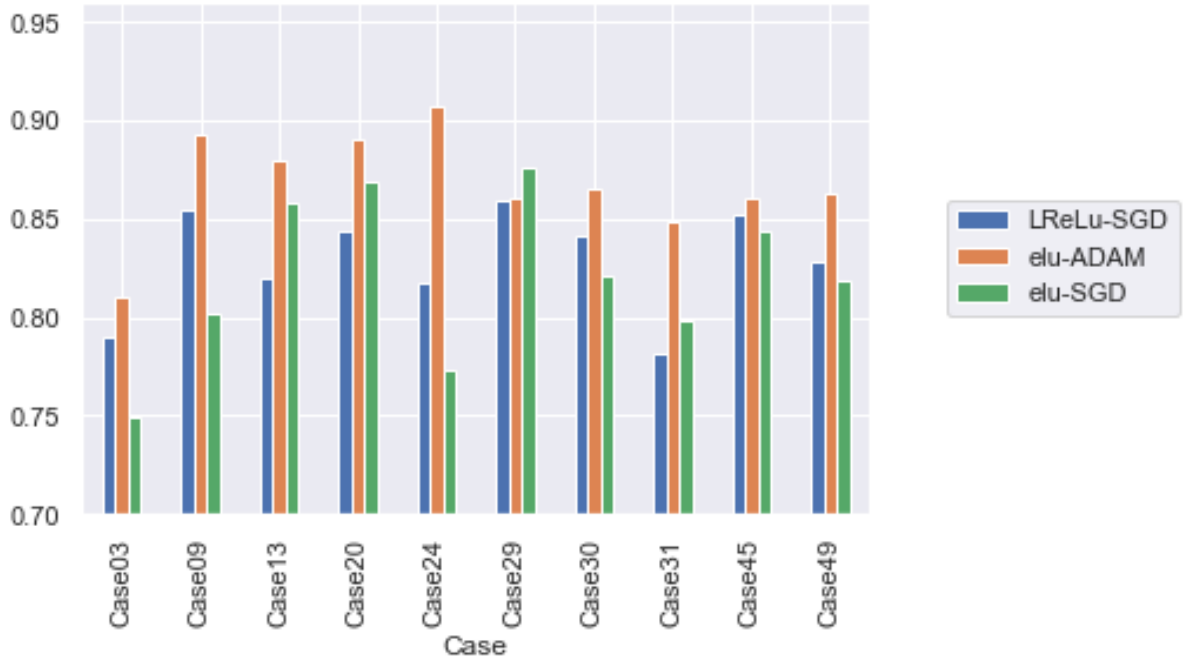

Figure 12: Grouped Bar Graph showcasing the Dice Scores achieved for each test case in the initial data-set configuration

The high scoring values were already established on this data arrangement so the second round of testing was undertaken with a differing configuration.

Within the 2nd rounds results Elu-Adam showed significantly more consitent results than the other models only showing a large efficacy drop on case 23. The differences in efficacy of the specific optimisers was more apparent here than in the first test, both SGD builds handled case 23 far better than the Elu build, but the inverse was also true case 19 where both SGD builds saw a massive drop, with LRelu's been the most extreme.

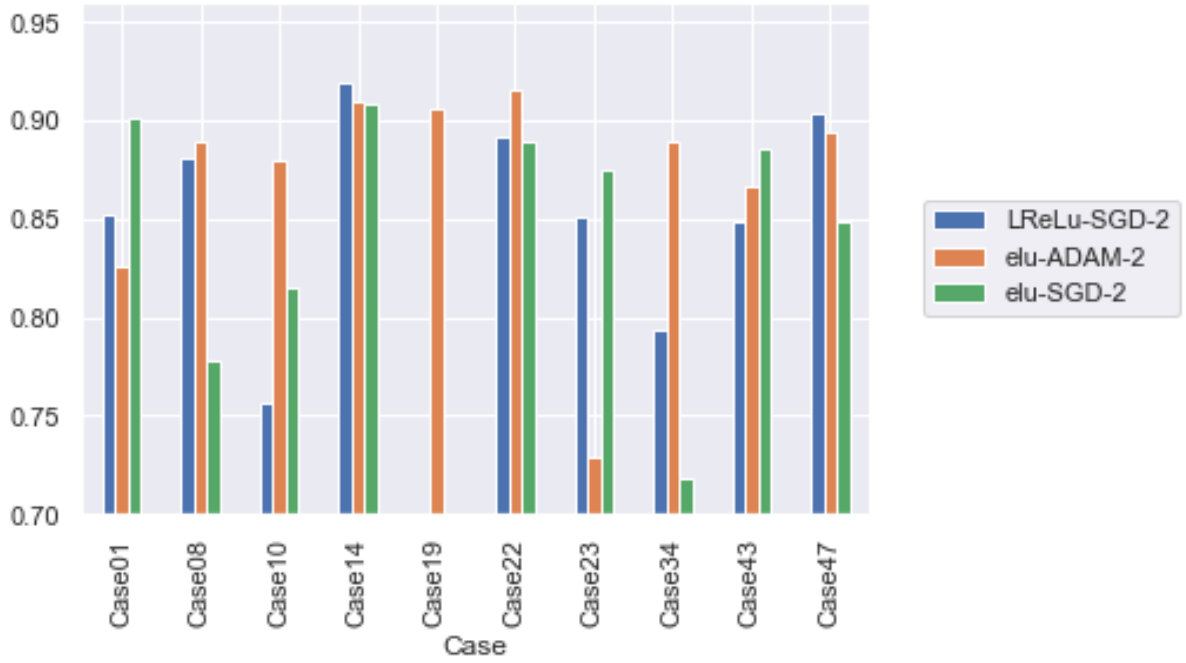

Figure 13: Grouped Bar Graph showcasing the Dice scores achieved on a new data-set configuration

Round	ELU-ADAM	ELU-SGD	LReLU-SGD
1	0.867540777	0.820839405	0.828844726
2	0.870441735	0.830230534	0.818103313

Table 10: Mean Dice Scores Achieved from the well performing models in the 2nd round of testing.

The two rounds confirmed the efficacy of all 3 models, and showcased the Elu-ADAM build as being the most effective when considering stability amongst the different cases and overall average Dice score.

0.11 Application to nnU-Net

The data was data-set was converted to nii.gz files from mhd. The newly selected ELU-ADAM model was then ran against the LReLU- SGD model, whilst keeping other parameters the same and using the 3D model present in nnU-Net. It should be noted time constraints only allowed for a single run-through on each model to confirm efficacy. The scores graphically represented throughout this section mostly present the score achieved without post processing, this was done to view the raw efficacy of the configurations

present, furthermore the post-processed scores showed very slight improvements with the average Dice score being 0.0001 higher in the ELU build and 0.00003 in the LRelu Build.

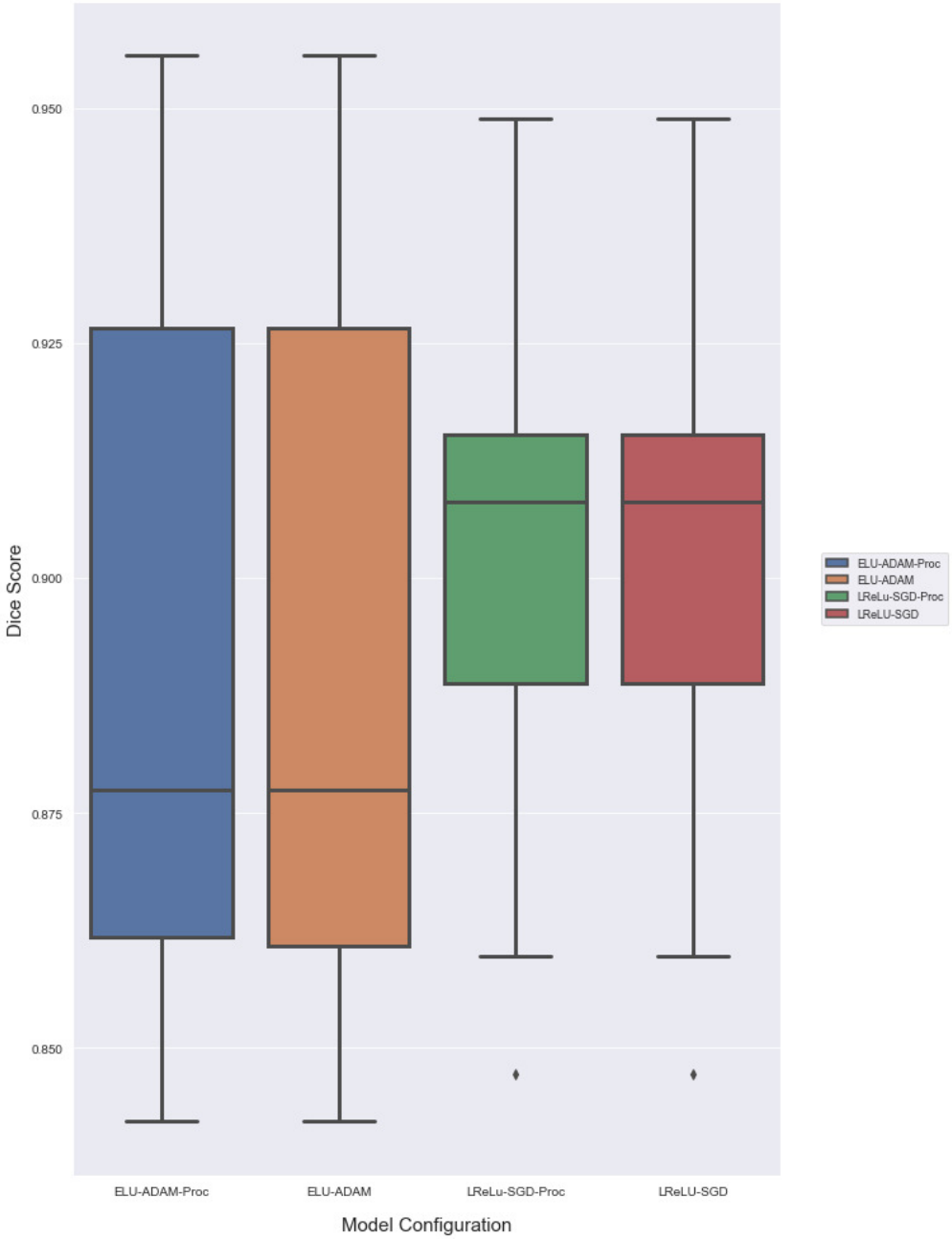

Figure 14: Box plot showing the range of scores achieved using both nnU-Net models, the proc models indicate DSC scores achieved after post-processing was applied

As shown in Table 14 both models performed with high efficacy resulting in median scores similar to one another with individual scores for both processed and processed being shown in Table 19.

Case	ELU	LReLU	ELU Post-Processed	LReLU Post-Processed
02	0.8731462479	0.90154446	0.873146247	0.901544459
03	0.86601437	0.884541664	0.86601437	0.859785050
05	0.853171278	0.847216313	0.853171278	0.847216313
19	0.859106782	0.910201129	0.860391603	0.906112180
20	0.881750338	0.859785051	0.881750338	0.884541664
25	0.935610612	0.948959507	0.936103531	0.948959507
28	0.842139468	0.914785994	0.842139468	0.914785994
31	0.932568326	0.915432723	0.932568326	0.915432723
39	0.908705882	0.947092261	0.908705882	0.947402952
43	0.95562825	0.90611218	0.95562825	0.910201129
Mean	0.890784155	0.903567128	0.890961929	0.903598197

Table 11: Testing DSC scores from both nnU-NEt models and their post processed variants.

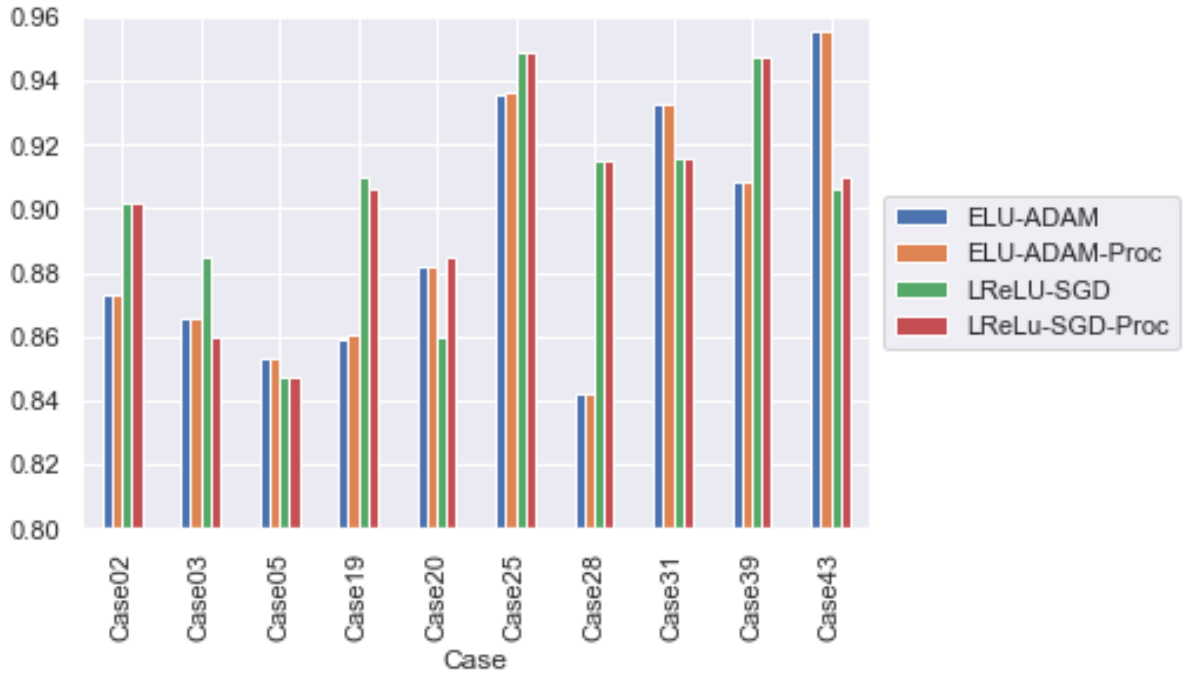

Figure 15: Grouped Bar plot showcasing the Dice scores achieved for both nnU-Net models across the testing set

0.12 Discussions

Both models showed significant efficacy when tested on this data-set configuration, with no DSC score coming in below 0.84. Both models produced raw results from testing as well as results achieved with a post processing component. This added post-processing a marginal increase in efficacy that was not deemed significant. When considering the raw results, the ELU model achieved a mean DSC of 0.890784155, fairing worse than LReLU which produced an mean of 0.903567128, this was only marginal though with the overall difference being under 0.1 DSC .The inverse scenario was true when testing out the V-Net builds, with ELU outperforming LReLU both highest score achieved and highest average score. This could be down to the pre-processing element or the differing initial learning rate , as well as other factors, applying the same to ELU would be the next logical step in confirming the learning rates impact.

The efficacy did not indicate that either configuration was significantly superior, with fluctuations in results likely varying along with the data-set cases used for testing, however more models would need to be run on the data-set to confirm this theory. Furthermore the differing hyper-parameter configurations seemingly had little effect when compared to the dice scores seen in the PROMISE12 data, including the samller initial learning rate of 0.001, although this was still impacted upon by the poly learning rate function so likely dropped relatively quickly and took a slightly shorter time to converge, though testing with the base nnU-Net model and applying this higher learning rate would allow confirmation of this point .

In terms distribution of scores across the test set, more variance was seen than when just considering the highest mean DSC, with LReLU's median score being significantly higher than ELU'. As well as this, the range of values produced by the LReLu builds is notably smaller, indicating more consistent results across the data set with the exception of the singular outlier present. This result is echoing that seen in the V-Net builds, in which LReLU's range was also significantly more compact, possibly suggesting LReLU is overall less impeded by image heterogeneity. In terms of the values of DSC achieved from both models, the range was shown to be largely similar, although with ELU producing both the lowest scoring and highest scoring test and having a much larger inter-quartile range. This suggests the LReLu is a more ideal choice as it appears less impacted by heterogeneity in the data-set, or is possibly working in a more synergistic fashion with the pre-processing than the ELU build is.

This potential impact of heterogeneity on efficacy was reinforced from the V-Net builds, specifically when the LReLu-SGD build that performed well in the first round producing the second highest median score with a small range, saw certain images such as Case19 in the second round resulting in a significantly lower DSC than any other image. At the same time these points are somewhat refuted from the consistency of the ELU builds and the poor performing builds when retested. However this point may be valid and this could potentially indicate ELU as being the more consistent choice across the data-set.This does indicate that the carry over from V-Net to nnU-Net may be limited, and that nnU-Nets complex strucutre negates findings from the V-Net builds being echoed within it's results.

Overall the findings produced here showcase the efficacy of nnU-Net reinforcing the published data relating to it as well as the PROMISE12 score it has achieved. However

altering the activation function and gradient descent algorithm shows little impact on nnU-Nets efficacy, although the majority of findings between models don't line up, this echoes the results from the V-Net builds where both these models where high scorers with little variance between them.

However, as mentioned before, validating the points above would require further testing to confirm and as a result these points lie out of the scope of the paper . This underlies a core issue with this paper, as due to GPU sourcing issues and time constraints only a small number of models could be ran, and was limited to 2 days run-time before having to be restarted. To fully justify the points mentioned above, further models would need to be run using these nnU-Net builds,ideally with differing data-set configurations.

In terms of efficacy across the models modalities, the time constraints allowed for the comparative nnU-Net models to be run only in the 2D modality, so it is highly possible 3D and cascaded variants display vastly different efficacy. In terms of direct translation from V-Net efficacy carried over well when moving from 3D to 2D, indicating these hyper parameter choices may present a strong model across modalities. When looking at certain publications, the 2D variant has proven to be a higher scoring model when compared to nnU-Nets 3D build however this is not significantly higher and was validated on the inferred data they provide where-as this paper has only used the training data for it testing, never the less both internal and external literature does elude to there being only a minor difference in the efficacy between modalities (Isensee et al. 2019).

In regards to the data-set validation performed, the weak FOLD 4 results appeared to be an anomaly, as further results within the paper did not replicate this poor performance. Most notably this was not echoes in the testing of differing V-Net builds, where the ReLU-SGD build fared signifcantly better, with the worst DSC score being 0.7124. Furthermore test case 49 was used in the V-Net training and presented a DSC of 0.821 in the ReLU-SGD build. It is difficult to decipher what caused this anomaly without re-testing, it is likely the model was setup incorrectly, although other low scores exhibited in ReLU-ADAM and LReLU-ADAM presented after a re-test, at which point their setup was confirmed to be correct and the low scores were confirmed to be present on those models. Never the less the nature of internally testing with the same test cases negated the need to remove outliers form the data-set as models efficacy was being leveraged against one another.

0.13 Conclusions and Further Directions

From the limited application in this paper caused by external constraints, it is hard to validate any claims regarding ideal setup of nnU-net or the role of image heterogeneity and pre-processing. What it does confirm is the efficacy of the nnU-Net setup in the literature and available online, making it an excellent out of the box model for use in segmenting the prostate. This shown in terms of efficacy when evaluating metrics, ease of use in terms of setup and navigating its online documentation, and its consistency across modalities and data configurations.

These last points are need of validation, it is recommended that for confirmation of the

models robustness in these factors the models are ran on a variety of data-set configurations and indeed data-sets themselves, we were limited in selecting the most ideal publicly available data-set in this paper, initially there was to be an internally sourced data-set, but issues arose in collating this data. Ideally once fully utilising the PROMISE12 set and potentially submitting it for official scoring (with the blessing of both models developers), the internally sourced set as well as other public-ally available sets should be used to further test the effects of data heterogeneity on the models observed in this paper to provide more insight into this issue.

To further add to this the efficacy of other nnU-Net internal models is a lacking section in this paper with only the 2D model being used and even then only for one run. nnU-Net provides internal model selection once the training data has been fed in allowing for it to setup an ideal build based off the data it collates. The findings from V-net may be more applicable to the 3D model and ELU could present a more ideal choice in that scenario.

Another to point to consider is that although nnU-Nets efficacy remained high in internal testing, there was a very small range of Dice scores achieved throughout both tested models. This limited range was not seen as strongly in the V-net testing phase, most likely due to the higher variety of activation functions tested as well . This likely indicates that at this high level hyper-parameter tuning whilst still providing a slight boost in efficacy does not play as vital a role as pre-processing now does. However only a limited number of hyper-parameters where tuned in these models and more extreme differences achieved through methods such as grid search may discredit this point. It is suggested that to fully understand the role hyper parameters play alongside nnU-Nets pre-possessing, that grid search be employed and more variable builds run with V-Net and nnU-Net.

This would also allow for more comparison between perceived efficacy in both models and how a good V-Net build translates to nnU-Net, a more short term method of gaining insight into this would be running a poor performing V-net build such as ReLu-ADAM and seeing if its performance was echoed in nnU-Net. At current it appears the carryover is rather limited, as a result if future research is time limited, its advised modelling only be performed with nnU-Net.

Once thees variables are check, if further efficacy or confirmation of efficacy was required, pre-processing would next logical step, depending on the importance gauged from these other tests. This paper has remained based around hyper parameters and model structure and at offers very little insight into the role of pre-processing so future research would compliment it greatly.

More broadly speaking it would ideal to use nnU-Nets automatic configuration which actually serves to negate the need for hyper-parameter selection. Testing this against a more varied selection of hyper-parameter configurations selected by researchers or obtained using Grid search algorithms that differ from nnU-Nets AutoML method. The confirmation of nnU-Nets hyper parameter selections effect on efficacy could remove the need for future papers surrounding hyper parameter tuning and allow for a more robust conclusion surrounding the use of pre-processing's importance, and how it may now be the more vital component for building an effective model. As well as this the fact the parameters not normally tweaked were altered in this paper may have a combined effect with the AutoML selection, which could either be positive or negative. To fully see the extent of the efficacy that may be achieved inference with these parameters is a must.

Overall a well performing model and robust data-set has selected and their idealistic features displayed. For internal future research, PROMISE12 provides a varied seemingly robust data-set whilst nnU-Net makes for a fine model to employ not only for its efficacy but for the amount of resources available for its setup and application.The combination of both should allow for out of the box efficacy in research or an ideal starting point for future model development.

0.14 Reflective Learning

This project has been a bit of a roller coaster for me.

Initially this project was set on selecting a model based on published literature and internal testing, in hopes of selecting a model to be used within a PHD project. However with GPU availability at an all time low I was forced to turn to utilising a remote computer provided by Cardiff university, their Linux labs. Whilst they are undeniably an excellent resource to the university I believe my lack of understanding on how they function hindered by ability to use their machines to my benefit. With multiple attempts at running models on there resulting in my data being removed or the model continuously being interrupted by machine faults and shutdowns.Leading to data collection being continuously being pushed back closer to the deadline.

This swayed my approach on the project and after discussion with my supervisor lead to me altering my approach, instead of running many models and slowly whittling them down to a final one the brunt of the model selection was now to be done using the literature. With the core of the project still being the selection of an ideal model, this was more of a lateral move than a compete re-write. As a result the modelling now revolved around confirming the efficacy of the selected model by attempting to compare its current "ideal version" to a version developed from my internal testing.This also coincided with the fact the internal data-set we were attempting to source had been held-up so the focus moved to also identifying an ideal data-set that models could use if the internal data-set was not available.

This reduced the number of models to be run but still left me short on time, and with persisting issues using the Linux lab my project was extended to meet this, now with the caveat of working a full time role around it. With said role requiring a lot of self learning outside of its core hours initially I struggled greatly to balance the two.

Switching to super-computing wales took some time as I had to learn how to navigate its Linux build, schedule models to run, set up environmental variables on the shared computer and deal with time limitations imposed on the system. However, once I got to grips with this I was able to run the models I needed with far less interruptions although certain weeks I would setup models to run whilst I was preoccupied and return to find they had not initialised properly, never the less SCW allowed me to complete this project, and I must admit I could of ran more models if I had headed the advice early on form my supervisor to switch to SCW.

A core lesson to take away form this is the not be intimidated by altering core aspects of the project even if you had per-sued a different direction until that point, I was reluctant to do so as I thought it would essentially cost me time to learn an entirely new system as I had already been doing so for the Linux lab. This was not the case though as the knowledge I had gained on Linux terminal carried over greatly and although I did have to learn the nuances of SCW it was a much less steep learning curve than anticipated ,especially in relation to the Linux labs computers. Furthermore I believe the model I have selected fully earns its merit without the need for internal testing of other models, and I am happy I was able to achieve what I did in this paper.

This indicated to me that I was potentially perceiving time management in the wrong way

and that being too dogmatic in the path I was taking resulting in said time management acting as more of a hindrance than a benefit. My take away form this was that time management doesn't need to be a linear mechanism and that investing time in resources that where not expected in the original itinerary for a task can greatly benefit it.

As mentioned above i feel like my newfound abilities to use the Linux terminal stand out as a high-point in displaying the knowledge I gained from this project, it was an aspect I was hesitant about at the beginning of this project, as I had only ever used windows and the majority of my computing knowledge had only been gained in the last year during my degree. However i found it quite intuitive and have ended up preferring it to the windows terminal, now utilising Linux commands by choice in window due to this preference. This also more generally extended my comfort and knowledge when it came to using any form of command line interface, and Im not sure I would of ever delved into this naturally as I would have always chosen the UI if it was down to my own volition.

I think most would consider my degree to lean more towards the software engineering route than that of more broad computer science, coding was much more of a focus than the principles behind it, so coming into the project with little to no knowledge of AI was initially quite a shock. This arose as understanding the principles behind the models is such a core part of AI, in a certain way coding is a very small part of this field. As a result I had to do a lot of reading to gain even the initial understanding of how image segmentation is achieved, this pulled on mathematical knowledge as well. I was genuinely surprised at my grasp of the core concepts, as it they were not as much of a challenge as anticipated, particularly the maths side of things, although I didn't delve excessively deep into the mathematics aspects and my knowledge of that aspect is more high level than other aspects such as model structure. Overall I do believe I have gained a well rounded understanding of automatic imaging techniques, in terms of this paper it may have been a bit unfocused as techniques such as GANS were delved into at one time, and aspects such as these may have bled into the literary portion of the paper, however I do feel like it provides excellent focus on the task at hand. In terms of my more general development this extra reading definitely aided in my understanding of the autonomous imagine field, possibly to a degree not expected of the writer of this paper, a good example of this would be the papers I read on lesion imaging of the prostate and the deeper understanding I gained of the MRI's.

My skill coding in Python was my strong point entering this project, but it was still in its infancy at that time, as mentioned before the coding aspect was not the largest part of this project but the code written and deciphered significantly increased my comfort with the language as well as expanding my knowledge on its potential applications. The code implemented within the model provided the most severer challenge as understanding of the model was required to ensure changes did not impact the efficacy or stability of said model. I was definitely lacking understanding of the model the first time I ran it, having done reading around it and having a brief overview of the code and listening to the ideology of the code being "just boiler plate". Instead, my main gain in understanding was galvanised by altering the code and implanting new structures such as the cross validation. This indicated to me that at least in terms of coding and me understanding complex structures within, I should be treating the literature associated with it as a supplement as apposed to the main source of knowledge and that interacting with the code was the way I learnt it best. This technique has already proved invaluable to me in my working life, and is one that I will likely be employing for the rest of my time interacting with code and computers as a whole.

The more comfortable coding came in the form of utilising python to form the data analytic such as the scatter and box plots seen in this paper, as well as using packages such as SKLearn to convert images into compatible data formats. Although this didn't prove as difficult to me as other aspects of this project it was key for my personal developments as it showed me the wide range of applications code has, before this I would of looked for pre-made software that would perform these. Now I am much more keen to investigate if a task I need done can-be implemented in code, and this has generated a positive feedback loop of me attempting more tasks and performing them in code, which has led to me being a much more well rounded coder.

Another technical skill I gained was the ability to construct documents in Latex, initially I found this quite a schock as although the learning curve is not very steep specific tasks can be tricky to navigate especially when compared to using more mainstream documentation methods. However this has proven to be another vital skill I have carried over to my work-life, now writing the majority of my documents in Latex and seeing noticeable improvements of my understanding in object based languages such as HTML.

All in all I think this project has greatly benefited me as a person. I have a much greater appreciation of the struggles researchers go through and that the actual research can end up being one of the least challenging factors, and that the resources available for certain Fields may not be as abundant as I first thought.

Personally I have developed from someone who only knew AI in a very high level manner, to someone who can now understand the underlying mechanisms in place as well as the overall model structure, how its tailored to the role it performs and how it resolves persisting issues seen across the field. I now have a decent understanding of the hierarchy of models seen in this field as well as being able to assess their efficacy in relation to this hierarchy. Furthermore, I was able to successfully add aspects to models, tweak their hyper parameters, alter their input data and run them on an operating system I had never used, with only the command line interface. Complementing this has been my increased use of code to interpret the data from both internal and external models.

As well as this I have seen mass gains in my soft skills, my initial reluctance to change aspects of project has shown me that change isn't always negative and that revaluation of a project doesn't need to stop early on in its development as changes past this time can be of great benefit. This has conceited with my improvements in time management, propagated by the poor state this skill was in a the beginning of the project but ending up being a string suite with an itinerary being drawn out for the second half of the project as well as daily progress recording . Furthermore I learnt how I best interprets complex structures such as the models code, an invaluable skill that stretches well beyond coding. Finally I learnt to not be afraid to ask for help on certain things, I tended to think of dissertation as a very isolated piece of work but interactions with computing staff across this projects span showed me that even such an isolated piece of work can greatly benefit from the input of others.

Finally I just want acknowledge that things don't always go to plan, and that is core part of research, although I did find aspects of this project very stressful I credit it as a vital part of my development as a person, not just academically and I am happy I persevered through the low points.

Bibliography

(2020). image segmentation - neural network probability output and loss function (example: dice loss).

(2021). Mic-dkfz/nnunet.

Abbasi, A. A., Hussain, L., Awan, I. A., Abbasi, I., Majid, A., Nadeem, M. S. A., and Chaudhary, Q.-A. (2020). Detecting prostate cancer using deep learning convolution neural network with transfer learning approach. *Cognitive Neurodynamics*, 14:523–533.

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4:e00938.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:2274–2282.

Alamri, N. M. H., Packianather, M., and Bigot, S. (2022). Deep learning: Parameter optimization using proposed novel hybrid bees bayesian convolutional neural network. *Applied Artificial Intelligence*, 36.

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*.

Astono, I. P., Welsh, J. S., Chalup, S., and Greer, P. (2020). Optimisation of 2d u-net model components for automatic prostate segmentation on mri. *Applied Sciences*, 10:2601.

Ayache, N. and Al, E. (2012). *Medical image computing and computer-assisted intervention–MICCAI 2012. Part II 15th International Conference, Nice, France, October 1-5, 2012, Proceedings*. Springer.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495.

Barentsz, J. O., Weinreb, J. C., Verma, S., Thoeny, H. C., Tempany, C. M., Shtern, F., Padhani, A. R., Margolis, D., Macura, K. J., Haider, M. A., Cornud, F., and Choyke, P. L. (2016). Synopsis of the pi-rads v2 guidelines for multiparametric prostate magnetic resonance imaging and recommendations for use. *European Urology*, 69:41–49.

Barrett, K. E., Barman, S. M., Brooks, H. L., X, J., and Ganong, W. F. (2019). *Ganong's review of medical physiology*. Mcgraw-Hill Education ; London.

Bengiot, Y., Frasconit, P., and Simardt, P. (1993). The problem of learning long-term dependencies in recurrent networks.

Berger, A. (2002). Magnetic resonance imaging. *BMJ (Clinical research ed.)*, 324:35.

Berrar, D. (2019). Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology*, pages 542–545.

BUBER, E. and DIRI, B. (2018). Performance analysis and cpu vs gpu comparison for deep learning.

Chauvin, Y. and Rumelhart, D. E. (2009). *Back propagation : theory, architectures, and applications*. Psychology Press.

Chen, J., Wan, Z., Zhang, J., Li, W., Chen, Y., Li, Y., and Duan, Y. (2021). Medical image segmentation and reconstruction of prostate tumor based on 3d alexnet. *Computer Methods and Programs in Biomedicine*, 200:105878.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation.

Chen, W., Zhang, Y., He, J., Qiao, Y., Chen, Y., Shi, H., and Tang, X. (2018). Prostate segmentation using 2d bridged u-net.

Chigozie, E., Nwankpa, W., Ijomah, A., Gachagan, S., and Marshall (2018). Activation functions: Comparison of trends in practice and research for deep learning.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Cunningham, P. and Delany, S. (2021). Underestimation bias and underfitting in machine learning.

Dosovitskiy, A., Springenberg, J., Tatarchenko, M., and Brox, T. (2016). Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Dozat, T. (2016). Workshop track -iclr 2016 incorporating nesterov momentum into adam.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., and Dehmer, M. (2020). An introductory review of deep learning for prediction models with big data. *Frontiers in Artificial Intelligence*, 3.

Fritscher, K., Raudaschl, P., Zaffino, P., Spadea, M. F., Sharp, G. C., and Schubert, R. (2016). Deep neural networks for fast segmentation of 3d medical images. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 158–165.

Gao, J., Yang, Y., Lin, P., and Park, D. S. (2018). Computer vision in healthcare applications.

Gao, Y., Liao, S., and Shen, D. (2012). Prostate segmentation by sparse representation based classification. *Medical Physics*, 39:6372–6387.

Gholamalinezhad, H. and Khosravi, H. (2021). Pooling methods in deep neural networks, a review.

Ghose, S., Mitra, J., Oliver, A., Martí, R., Lladó, X., Freixenet, J., Vilanova, J., Sidibé, D., and Meriaudeau, F. (2012). A random forest based classification approach to prostate segmentation in mri.

Giganti, F., Rosenkrantz, A. B., Villeirs, G., Panebianco, V., Stabile, A., Emberton, M., and Moore, C. M. (2019). The evolution of mri of the prostate: The past, the present, and the future. *American Journal of Roentgenology*, 213:384–396.

Gillespie, D., Kendrick, C., Boon, I., Boon, C., Rattay, T., and Hoon Yap, M. (2020). Deep learning in magnetic resonance prostate segmentation: A review and a new perspective.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. The Mit Press.

Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1768–1783.

Gupta, S. (2013). Concrete mix design using artificial neural network. *Journal on Today's Ideas-Tomorrow's Technologies*, 1:29–43.

Habes, M., Schiller, T., Rosenberg, C., Burchardt, M., and Hoffmann, W. (2013). Automated prostate segmentation in whole-body mri scans for epidemiological studies. *Physics in Medicine and Biology*, 58:5899–5915.

Hamerly, G. and Elkan, C. (2003). Learning the k in k-means.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Hilton, G., Sejnowski, T. J., and Technology, M. I. O. (1999). *Unsupervised learning : foundations of neural computation*. Massachusetts Institute Of Technology.

Ibrahim, A. and El-Kenawy, E.-S. (2020). Image segmentation methods based on superpixel techniques: A survey.

Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., and Maier-Hein, K. H. (2020). nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18:203–211.

J. Sathianathen, N., Omer, A., and Harris, E. (2020). Negative predictive value of multiparametric magnetic resonance imaging in the detection of clinically significant prostate cancer in the prostate imaging reporting and data system era: A systematic review and meta-analysis. *European Urology*, 78:402–414.

Jereczek-Fossa, B. (2014). The utility of diffusion weighted imaging (dwi) using apparent diffusion coefficient (adc) values in discriminating between prostate cancer and normal tissue. *Polish Journal of Radiology*, 79:450–455.

Jia, H. (2020a). Method description.

Jia, H. (2020b). Method$_d$*escription.pdf*.

Jia, H., Song, Y., Huang, H., Cai, W., and Xia, Y. (2019). Hd-net: Hybrid discriminative network for prostate segmentation in mr images. *Lecture Notes in Computer Science*, pages 110–118.

Kandel, I. and Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6:312–315.

Kansal, S. (2020). Quick guide to gradient descent and it's variants.

Kato, Z. (2011). Markov random fields in image segmentation. *Foundations and Trends® in Signal Processing*, 5:1–155.

Khan, Z., Yahya, N., Alsaih, K., Ali, S. S. A., and Meriaudeau, F. (2020). Evaluation of deep neural networks for semantic segmentation of prostate in t2w mri. *Sensors*, 20:3183.

Khan, Z., Yahya, N., Alsaih, K., and Meriaudeau, F. (2019). Zonal segmentation of prostate t2w-mri using atrous convolutional neural network.

Khosa, C. K., Mars, L., Richards, J., and Sanz, V. (2020). Convolutional neural networks for direct detection of dark matter. *Journal of Physics G: Nuclear and Particle Physics*, 47:095201.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90.

Kłoczko, M. (2017). Superpixels-based image segmentation superpixels-based image segmentation view project.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324.

Lee, J. J., Thomas, I.-C., Nolley, R., Ferrari, M., Brooks, J. D., and Leppert, J. T. (2014). Biologic differences between peripheral and transition zone prostate cancer. *The Prostate*, 75:183–190.

Lee, S., Kang, Q., Al-Bahrani, R., Agrawal, A., Choudhary, A., and Liao, W.-k. (2022). Improving scalability of parallel cnn training by adaptively adjusting parameter update frequency. *Journal of Parallel and Distributed Computing*, 159:10–23.

Li, A., Li, C., Wang, X., Eberl, S., Feng, D. D., and Fulham, M. (2013). Automated segmentation of prostate mr images using prior knowledge enhanced random walker.

Li, H., Zhao, R., and Wang, X. (2014a). Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification.

Li, M., Zhang, T., Chen, Y., and Smola, A. J. (2014b). Efficient mini-batch training for stochastic optimization. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Lian, X. and Liu, J. (2019). Revisit batch normalization: New understanding and refinement via composition optimization.

Liu, D., Xiong, Y., Pulli, K., and Shapiro, L. (2011). Estimating image segmentation difficulty.

Liu, X., Langer, D. L., Haider, M. A., Yang, Y., Wernick, M. N., and Yetik, I. S. (2009). Prostate cancer segmentation with simultaneous estimation of markov random field parameters and class. *IEEE Transactions on Medical Imaging*, 28:906–915.

Liu, X., Song, L., Liu, S., and Zhang, Y. (2021). A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13:1224.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation.

Lovegrove, C. E., Matanhelia, M., Randeva, J., Eldred-Evans, D., Tam, H., Miah, S., Winkler, M., Ahmed, H. U., and Shah, T. T. (2018). Prostate imaging features that indicate benign or malignant pathology on biopsy. *Translational Andrology and Urology*, 7:S420–S435.

Lu, Z., Zhao, M., and Pang, Y. (2020). Cda-net for automatic prostate segmentation in mr images. *Applied Sciences*, 10:6678.

Malekijoo, A. and Fadaeieslam, M. J. (2019). Convolution-deconvolution architecture with the pyramid pooling module for semantic segmentation. *Multimedia Tools and Applications*, 78:32379–32392.

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation.

Mitchell, T. M. (2017). *Machine learning*. Mcgraw Hill.

Mooij, G., Bagulho, I., and Huisman, H. (2018). Automatic segmentation of prostate zones.

Mustapha, A., Mohamed, L., and Ali, K. (2020). An overview of gradient descent algorithm optimization in machine learning: Application in the ophthalmology field. *Communications in Computer and Information Science*, pages 349–359.

Nichols, J. A., Herbert Chan, H. W., and Baker, M. A. B. (2018). Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical Reviews*, 11:111–118.

Nilsson, N. J. (1996). Artificial intelligence: A modern approach. *Artificial Intelligence*, 82:369–380.

Oapos;shea, K. and Nash, R. (2015). An introduction to convolutional neural networks.

Oerther, B., Engel, H., Bamberg, F., Sigle, A., Gratzke, C., and Benndorf, M. (2021). Cancer detection rates of the pi-radsv2.1 assessment categories: systematic review and meta-analysis on lesion level and patient level. *Prostate Cancer and Prostatic Diseases*.

Rawla, P. (2019). Epidemiology of prostate cancer. *World Journal of Oncology*, 10:63–89.

Rehmer, A. and Kroll, A. (2020). On the vanishing and exploding gradient problem in gated recurrent units. *IFAC-PapersOnLine*, 53:1243–1248.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

Rosasco, L., Vito, E. D., Caponnetto, A., Piana, M., and Verri, A. (2004). Are loss functions all the same? *Neural Computation*, 16:1063–1076.

Rundo, L., Han, C., Nagano, Y., Zhang, J., Hataya, R., Militello, C., Tangherloni, A., Nobile, M. S., Ferretti, C., Besozzi, D., Gilardi, M. C., Vitabile, S., Mauri, G., Nakayama, H., and Cazzaniga, P. (2019). Use-net: Incorporating squeeze-and-excitation blocks into u-net for prostate zonal segmentation of multi-institutional mri datasets. *Neurocomputing*, 365:31–43.

Sak, H., Senior, A., and Google, B. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling.

SAMPSON, G. (1999). Dafydd gibbon, roger moore, and richard winski (eds). handbook of standards and resources for spoken language systems. mouton de gruyter. 1997. isbn 3-11-015366-1. dm 298. xxx+886 pages. *Natural Language Engineering*, 5:301–307.

Schroff, F., Criminisi, A., and Zisserman, A. (2008). Object class segmentation using random forests.

Selvikvåg Lundervold, A. and Lundervold, A. (2018). An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29.

Seo, H., Badiei Khuzani, M., Vasudevan, V., Huang, C., Ren, H., Xiao, R., Jia, X., and Xing, L. (2020). Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications. *Medical Physics*, 47.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6.

Siegel, R. L., Miller, K. D., and Jemal, A. (2019). Cancer statistics, 2019. *CA: A Cancer Journal for Clinicians*, 69:7–34.

Simonyan, K. and Zisserman, A. (2015). Published as a conference paper at iclr 2015 very deep convolutional networks for large-scale image recognition.

Stabile, A., Giganti, F., Rosenkrantz, A. B., Taneja, S. S., Villeirs, G., Gill, I. S., Allen, C., Emberton, M., Moore, C. M., and Kasivisvanathan, V. (2019). Multiparametric mri for prostate cancer diagnosis: current status and future directions. *Nature Reviews Urology*.

Sumanasuriya, S. and De Bono, J. (2017). Treatment of advanced prostate cancer—a review of current therapies and future promise. *Cold Spring Harbor Perspectives in Medicine*, 8:a030635.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era.

Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., and Ashmore, R. (2019). Testing deep neural networks.

Sutton, R. S. and Barto, A. (2018). *Reinforcement learning : an introduction*. The Mit Press.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions.

Talathi, S. and Vartak, A. (2016). Improving performance of recurrent neural network with relu nonlinearity.

Tian, Z., Liu, L., Zhang, Z., and Fei, B. (2018). Psnet: prostate segmentation on mri based on a convolutional neural network. *Journal of Medical Imaging*, 5.

Vargas, H. A., Akin, O., Franiel, T., Goldman, D. A., Udo, K., Touijer, K. A., Reuter, V. E., and Hricak, H. (2012). Normal central zone of the prostate and central zone involvement by prostate cancer: Clinical and mr imaging implications. *Radiology*, 262:894–902.

Vincent, G., Guillard, G., and Bowes, M. (2012). Fully automatic segmentation of the prostate using active appearance models.

Wang, W., Yu, K., Hugonot, J., Fua, P., and Salzmann, M. (2019). Recurrent u-net for resource-constrained segmentation.

Wang, Z. (2019). Deep learning for image segmentation: veritable or overhyped?

Weinreb, J. C., Barentsz, J. O., Choyke, P. L., Cornud, F., Haider, M. A., Macura, K. J., Margolis, D., Schnall, M. D., Shtern, F., Tempany, C. M., Thoeny, H. C., and Verma, S. (2016). Pi-rads prostate imaging - reporting and data system: 2015, version 2. *European Urology*, 69:16–40.

Xiangxiang, Q., Yu, Z., and Bingbing, Z. (2018). Automated segmentation based on residual u-net model for mr prostate images.

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9:611–629.

Yan, L., Liu, D., Xiang, Q., Luo, Y., Wang, T., Wu, D., Chen, H., Zhang, Y., and Li, Q. (2021). Psp net-based automatic segmentation network model for prostate magnetic resonance imaging. *Computer Methods and Programs in Biomedicine*, 207:106211.

Yuan, J., Wang, D., and Li, R. (2012). Image segmentation using local spectral histograms and linear regression. *Pattern Recognition Letters*, 33:615–622.

Zhang, Y., Wu, J., Chen, W., Chen, Y., and Tang, X. (2019). Prostate segmentation using z-net.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network.

Zhao, R., Qian, B., Zhang, X., Li, Y., Wei, R., Liu, Y., and Pan, Y. (2020). Rethinking dice loss for medical image segmentation.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2019). Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, pages 1–1.

Zhu, Q., Du, B., Turkbey, B., Choyke, P. L., and Yan, P. (2017). Deeply-supervised cnn for prostate segmentation.

Zhu, Y., Wei, R., Gao, G., Ding, L., Zhang, X., Wang, X., and Zhang, J. (2018). Fully automatic segmentation on prostate mr images based on cascaded fully convolution network. *Journal of Magnetic Resonance Imaging*, 49:1149–1156.

Çiçek, , Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432.

APPENDIX

FOLD	CASE	DSC
1	Case28	0.843544185
1	Case10	0.797212183
1	Case35	0.849505603
1	Case02	0.830221117
1	Case34	0.827350557
1	Case29	0.815288424
1	Case24	0.842849374
1	Case13	0.831863105
1	Case22	0.877457201
1	Case33	0.802664459
1	Average	0.831795633
2	Case41	0.828293264
2	Case08	0.753793478
2	Case44	0.863909602
2	Case05	0.822674811
2	Case42	0.79643631
2	Case27	0.863778472
2	Case47	0.867089033
2	Case32	0.876167059
2	Case37	0.783816993
2	Case45	0.826455653
2	Average	0.828241467
3	Case06	0.651115417
3	Case04	0.852223754
3	Case20	0.843352735
3	Case25	0.903787017
3	Case09	0.845862269
3	Case21	0.831848025
3	Case31	0.828802407
3	Case03	0.746114492
3	Case16	0.863697171
3	Case14	0.836634994
3	Average	0.820343792
4	Case43	0.353404254
4	Case49	0.301529646
4	Case15	0.4274441
4	Case19	0.394715518
4	Case46	0.272013068
4	Case17	0.145750374
4	Case12	0.226359576
4	Case38	0.493457437
4	Case26	0.406985402
4	Case18	0.739034832
4	Average	0.376069427
5	Case36	0.610463679
5	Case07	0.624872088
5	Case48	0.631584525
5	Case11	0.882853448
5	Case00	0.851850629
5	Case39	0.795151055
5	Case30	0.766530931
5	Case01	0.568112493
5	Case40	0.753161788
5	Case23	0.595533907
5	Average	0.708011508

Figure 16: Full DSC's achieved during 5 fold cross validation of training set

CASE	ELU-SGD	ELU-ADAM	PReLU-SGD	PReLU-ADAM	LReLU-SGD	LReLU-ADAM	ReLU-SGD	ReLU-ADAM
Case09	0.801279068	0.892371774	0.857186258	0.854653895	0.8550331	0.270442426	0.829091311	0.071559869
Case24	0.773004532	0.907026529	0.53468591	0.834220588	0.817172408	0.537648797	0.712439358	0.046929866
Case13	0.857659757	0.879510283	0.80676949	0.823678553	0.820285678	0.237944826	0.808180571	0.066511579
Case03	0.749732256	0.809801996	0.658943355	0.642373025	0.790341437	0.103973992	0.873468101	0.030151434
Case45	0.843116641	0.859973609	0.795779347	0.733997464	0.852141619	0.145844012	0.834049165	0.036444537
Case20	0.869334698	0.889773726	0.646380603	0.788071752	0.843522668	0.227107465	0.798274934	0.062352363
Case30	0.821317911	0.865436196	0.748200595	0.822010577	0.840809762	0.176796556	0.829010844	0.086080194
Case29	0.876253724	0.860500395	0.850521505	0.884497941	0.858853877	0.581573784	0.82858479	0.123849846
Case49	0.818406522	0.862469018	0.802677929	0.783684969	0.828538835	0.256299168	0.821092427	0.064505823
Case31	0.798289239	0.848543346	0.763902009	0.759601414	0.781747401	0.384596914	0.803027332	0.098254062
Average	0.820839405	0.867540777	0.746504724	0.792679012	0.828844726	0.292222798	0.813721955	0.068663962

Figure 17: DSC's achieved in initial running of V-Net models

CASE	LReLU	ReLU
Case09	0.168305457	0.15760693
Case24	0.42911458	0.091643758
Case13	0.1440911	0.135687456
Case03	0.068364017	0.069671743
Case45	0.098856464	0.069386542
Case20	0.166996434	0.09510348
Case30	0.101349354	0.125112355
Case29	0.422036886	0.17561014
Case49	0.163297296	0.105452187
Case31	0.282128662	0.132474571
Average	0.20445402	0.115774922

Figure 18: DSC's of the poor perfoming models second test

CASE	LReLu-SGD	ELU-SGD	ELU-ADAM
Case34	0.794183493	0.718686521	0.889110267
Case14	0.9187181	0.90852046	0.909287512
Case22	0.891145229	0.889333189	0.915589213
Case19	0.485634744	0.683053851	0.905706763
Case10	0.755961955	0.814648628	0.8796556
Case43	0.84811765	0.885275185	0.866998971
Case23	0.850724518	0.875144362	0.729635537
Case01	0.851860523	0.901549339	0.825799048
Case08	0.881186306	0.777904153	0.888725579
Case47	0.903500438	0.848189831	0.893907726
Average	0.818103313	0.830230534	0.870441735

Figure 19: DSC's of the well performing models second run on new data configuration