CONVOLUTIONAL NEURAL NETWORKS FOR ULTRASOUND CLASSIFICATION

INVESTIGATING THE USE OF CONVOLUTIONAL NEURAL NETWORKS FOR PRENATAL HYDRONEPHROSIS ULTRASOUND IMAGE CLASSIFICATION

By LAUREN SMAIL, BSc

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree of Master of Science

DESCRIPTIVE NOTE

MASTER OF SCIENCE (2018) Psychology, Neuroscience & Behaviour	McMaster University Hamilton, ON, CANADA
TITLE:	Investigating the Use of Convolutional Neural Networks for Prenatal Hydronephrosis Ultrasound Image Classification
AUTHOR:	Lauren Smail, BSc (McMaster University)
SUPERVISORS:	Dr. Ranil Sonnadara, PhD Dr. Suzanna Becker, PhD
SUPERVISORY COMMITTEE:	Dr. Luis Braga, MD, PhD Dr. Jim Reilly, PhD
NUMBER OF PAGES:	xiv, 79

LAY ABSTRACT

Prenatal hydronephrosis is a serious condition that affects the kidneys of fetal infants and is graded using renal ultrasound. The severity of hydronephrosis impacts treatment and follow-up times. However, all grading systems suffer from reliability issues. Improving diagnostic reliability is important for patient well-being. We believe that developing a computer-based diagnostic aid is a promising option to do so.

We conducted two studies to investigate how ultrasound images should be processed, and how the algorithm that produces the functionality of the aid should be designed. We found that two common recommendations for ultrasound processing did not improve model performance and therefore need not be applied. Our best performing algorithm had a classification accuracy of 49%. However, we found that several images in our database were mislabelled, which impacted accuracy metrics. Once our images and their labels have been verified, we can further optimize our algorithm's design to improve its accuracy.

ABSTRACT

Prenatal hydronephrosis is a common condition that involves accumulation of urine with consequent dilatation of the collecting system in fetal infants. There are several hydronephrosis classifications, however all grading systems suffer from reliability issues as they contain subjective criteria. The severity of hydronephrosis impacts treatment and follow up times and can therefore directly influence a patient's well-being and quality of care. Considering the importance of accurate diagnosis, it is concerning that no accurate, reliable or objective grading system exists. We believe that developing a convolutional neural network (CNN) based diagnostic aid for hydronephrosis will improve physicians' objectivity, inter-rater reliability and accuracy.

Developing CNN based diagnostic aid for ultrasound images has not been done before. Therefore, the current thesis conducted two studies using a database of 4670 renal ultrasound images to investigate two important methodological considerations: ultrasound image preprocessing and model architecture. We first investigated whether image segmentation and textural extraction are beneficial and improve performance when they are applied to CNN input images. Our results showed that neither preprocessing technique improved performance, and therefore might not be required when using CNN for ultrasound image classification. Our search for an optimal architecture resulted in a model with 49% 5-way classification accuracy. Further investigation revealed that images in our database had been mislabelled, and thus impacted model training and testing. Although our current best model is not ready for use as diagnostic aid, it can be used to verify the accuracy of our labels.

Overall, these studies have provided insight into developing a diagnostic aid for hydronephrosis. Once our images and their respective labels have been verified, we can further optimize our model architecture by conducting an exhaustive search. We hypothesize that these two changes will significantly improve model performance and bring our diagnostic aid closer to clinical application.

ACKNOWLEDGEMENTS

I would first and foremost like to thank my co-supervisors, Dr. Ranil Sonnadara and Dr. Sue Becker, for their guidance and support this past year. Without you two none of this would have been possible. I also want to thank Dr. Kiret Dhindsa for always taking my questions. You've taught me a lot this past year, for which I am very appreciative.

I would also like to thank my parents, Sandra and Morgan, for their continued support and enthusiasm for my research, and my pursuit of higher education. It means the world to me when you ask about my research so that you can share it with your friends. I love you both. I am also very grateful for my two boys, Loki and Marcel. Thank you for always keeping me smiling.

Last, but definitely not least, I would like to thank my boyfriend Brayden. Thank you for all your love, support, and understanding this past year. Your recommendations for breaks and relaxation throughout this past year kept me sane. You always know how to make me laugh, for which I am forever grateful. Thank you for believing in me, I love you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

HN:	Prenatal Hydronephrosis
US:	Ultrasound
SFU:	Society for Fetal Urology
UTD:	Urinary Tract Dilation
APRPD:	Anterior-Posterior Renal Pelvic Diameter
ML:	Machine Learning
AI:	Artificial Intelligence
CADe:	Computer Aided Detection
CADx:	Computer Aided Diagnosis
CNN:	Convolutional Neural Network
ASM:	Active Shape Models
ReLU:	Rectified Linear Activation Function

DECLARATION OF ACADEMIC ACHIEVEMENT

The work in this thesis was conducted by Lauren Smail (hereafter referred to as 'student investigator'), including data cleaning, model design, and data analysis. All work was done under the supervision of Dr. Sonnadara and Dr. Becker with guidance from Dr. Braga and Dr. Reilly. Data collection was conducted by Dr. Braga and his medical team at McMaster Children's Hospital.

Chapter 1: Introduction to the Clinical Problem

Prenatal Hydronephrosis

Prenatal hydronephrosis (HN) is a condition that involves the accumulation of urine with consequent dilatation of the collecting system in fetal infants. It is the most frequent neonatal urinary tract abnormality – occurring in 1-5% of all newborn babies (Woodward & Frank, 2002). HN is detected by prenatal ultrasound (US) and can be caused by several underlying conditions. Although many cases eventually resolve on their own, in severe forms, afflicted infants can require surgery to address the obstruction, and failure to alleviate this blockage can result in permanent kidney damage or scarring.

There are currently several HN classifications, however all of these grading systems have suffered from reliability issues since they contain somewhat subjective criteria (Rickard et al., 2017). The severity of HN impacts treatment and follow up times and can therefore directly influence a patient's well-being and quality of care. Misclassification of any patient into the inappropriate HN category can be detrimental to their health, as well as to the healthcare system itself by incurring costs for procedures that would otherwise be unnecessary. It is apparent that a more accurate and reliable classification system for HN is still needed, and therefore the purpose of this thesis was to investigate and possibly create of a new objective diagnostic aid to help improve HN grading accuracy.

Current Methods of Diagnosis

All patients are normally evaluated after birth by postnatal renal ultrasonography to determine HN severity and the best course of treatment. The two most widely adopted classification systems for HN are the Society for Fetal Urology (SFU) and the Urinary Tract Dilation (UTD) grading systems (Nguyen et al., 2014, 2010).

Society for Fetal Urology classification system. The SFU system was created to standardize the grading of HN (Fernbach, Maizels, & Conway, 1993). Previously, the categorization of HN, as well as its management, was poorly defined, and many physicians would subjectively classify HN cases into mild, moderate and severe forms. The subjective nature of adopting such a grading system resulted in poor inter-rater reliability (Nguyen et al., 2010). Conversely, the SFU system was developed to allow for more objectivity in segregating the different HN grades and, at the same time, create quantitative measures to facilitate an appropriate diagnostic decision.

The SFU proposed a 5-point classification system that grades the upper urinary tract dilation by ultrasound, and focuses on the appearances of the calices, renal pelvis, and renal parenchyma. The severity of HN increases with each SFU grade with SFU grade 0 meaning the patient has no HN, while the highest grade, SFU grade IV, means the patient has severe HN, with parenchyma thinning and some degree of renal function deterioration. See Table 1 for a complete summary of the SFU grades, and their respective diagnostic features.

Table 1: Summary of the Society for Fetal Urology (SFU) grading system.

SFU Grade	Diagnostic Features
SFU Grade 0	Normal kidney
SFU Grade I	Splitting of the renal pelvis
SFU Grade II	Few but not all calices are dilated, in addition to the splitting of the renal pelvis
SFU Grade III	Wide splitting of the renal pelvis, plus all calices dilated; normal parenchymal thickness
SFU Grade IV	Further splitting of the renal pelvis, plus all calices dilated; parenchymal thinning

Evaluation of the SFU grading system. Although the purpose of the SFU system was to standardize and unify the way physicians graded HN, the system fell short of these goals, as lack of agreement regarding the definition of physiologic HN and its clinical management has persisted since the SFU creation (Nguyen et al., 2010). Many different grading systems (e.g. Anterior-Posterior Renal Pelvic Diameter (APRPD), European Society of Pediatric Urology, Uroradiology Task Force, and Onen Grading System) were still being utilized in place of the SFU system well after its release (Nguyen et al., 2014). Furthermore, the times that these different grading systems were being used varied, with some being used preferentially in prenatal evaluation, and others for postnatal evaluation of HN. The preferential split also depended on the physician's specialty, with pediatric radiologists preferring more descriptive grading systems, and urologists preferring more quantitative systems (Zanetta et al., 2012). Finally, there are no correlations between the various HN grading systems (Swenson, Darge, Ziniel, & Chow, 2015). Considering that the SFU classification was developed to remedy these types of issues, the persistence of these problems suggests that an improved HN grading system is still required.

While the SFU classification in general has been shown to have good intra-rater reliability, it has poor inter-rater reliability when distinguishing between "moderate" HN grades (SFU II/III); however, it is fairly successful in differentiating between "mild" (SFU I/II) and "severe" (SFU IV) HN cases (Keays et al., 2008; Nguyen et al., 2010; Rickard et al., 2017). It is interesting to note that in the original SFU classification study, Fernbach et al. (1993) reported similar findings. They showed that physicians could reliably differentiate between mild and severe cases of HN, however, for intermediate cases their ability to correctly identify the HN grade on US was low. This is rather curious considering the goal at inception was to produce a more objective grading system for HN that would result in consistent diagnoses across physicians. These findings can likely be explained by the fact that subjective interpretation is still required for physicians to assign specific SFU grades to HN renal US images (Keays et al., 2008; Rickard et al., 2017).

Urinary Tract Dilation classification system. The UTD grading system was developed to provide a unified classification with an accepted terminology for the management of prenatal and postnatal HN, and is based on detailed assessment of the current literature and expert opinion considering common clinical practice (Nguyen et al., 2014). It is stratified based on gestational age and is based on: 1) APRPD; 2) calyceal dilation; 3) renal parenchymal thickness; 4) renal parenchymal appearance; 5) bladder abnormalities; and 6) ureteral abnormalities (see Figure 1). The stratification is based on the most concerning ultrasound finding. For example, if a patient was found to have abnormal parenchymal thickness with peripheral calyceal dilation, they would be

considered UTD P3 (high risk) regardless of the severity of any of the other sonographic parameters; however, if the patient only has peripheral calyceal dilations they would only be considered UTD P2.

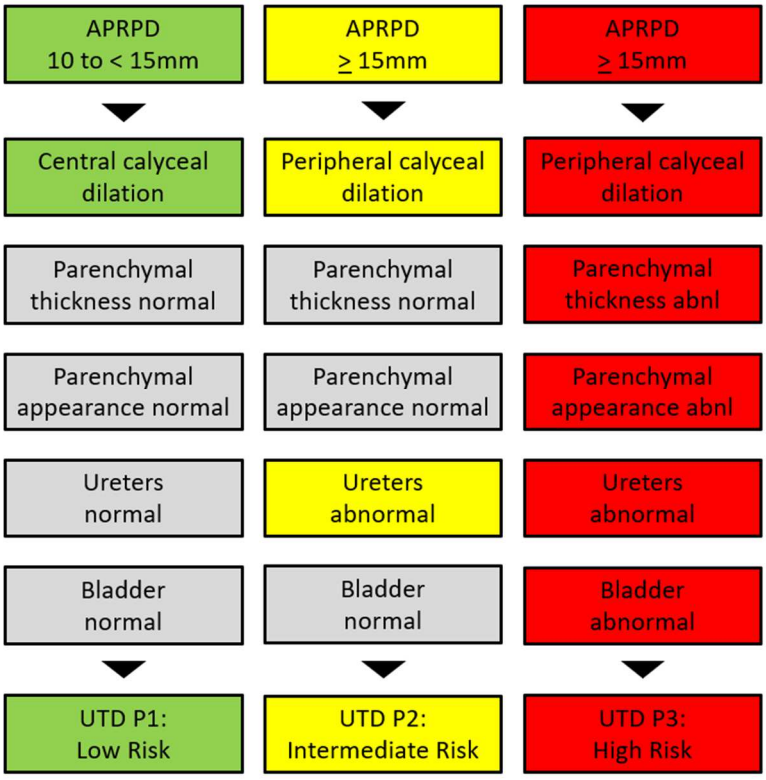

Figure 1. Summary of the Urinary Tract Dilation Risk Stratification system. Coloured boxes represent the abnormal feature(s) characteristic of each UTD grade. The patient is diagnosed based on the most concerning ultrasound feature(s) that result in each respective UTD grade. For example, a patient with only abnormal ureters would be classified as UTD P2, however if the parenchymal appearance was also abnormal the patient's diagnosis would change to UTD P3. Adapted from Nguyen et al. (2014) Figure 3.

Evaluation of the UTD classification system. The UTD system appears to have lower inter-rater reliability than the SFU classification, and just as poor reliability for intermediate grades as the SFU system (Rickard et al., 2017). Therefore, the nuances of

the intermediate grades are still difficult to objectively quantify, and the inability to appropriately classify intermediate grades is still concerning.

Purpose Statement

Considering the importance of accurate diagnosis in the treatment of HN, a very common congenital renal condition, it is concerning that no highly accurate, reliable or objective grading system currently exists. We believe that developing a diagnostic aid for HN so that all residents/physicians are trained to search for the same features will improve inter-rater reliability. Furthermore, we believe that developing a machine learning (ML) based diagnostic aid will improve the objectivity of physicians' diagnoses. Chapter 2 will introduce the concept of ML, some algorithms/models of interest, and how these can be applied to HN to begin to develop a diagnostic aid. It is important to note that poor diagnostic accuracy is an issue in many other disease states and image modalities as well. Therefore, although we will be using HN as our model in this thesis, the long-term goal is to generalize findings and concepts from these studies to other types diseases and image modalities.

Chapter 2: Overview of Machine Learning and Diagnostic Aids

A Brief History and Introduction to Machine Learning

ML, in the broadest sense, is a field of computer science and statistics that is focused on developing computer programs that can learn without explicitly being programmed. To do this, computers employ statistical techniques that allow them to find meaningful patterns in large datasets (Michalski, Carbonell, & Mitchell, 1983). Often ML algorithms hone in on combinations of variables that interact in complex ways – sometimes uncovering relationships that we didn't realize existed before.

ML algorithms date back to the mid 1900's, however, early discoveries of various learning principles such as Bayes or Least Squares error minimization, which are both still used in many ML algorithms today, date back to the mid 1700's (Legendre, 1805; Bayes & Price, 1763; Laplace, 1814). Some of the very first ML algorithms, such as the Stochastic Neural Analog Reinforcement Calculator (SNARC) developed by Minsky (1954), and the perceptron began to lay the groundwork for ML and built excitement among researchers and the general population.

The "perceptron" was developed by Rosenblatt (1958), and is a supervised learning algorithm, meaning it requires labelled data to learn to classify data. The perceptron takes a vector of inputs (x), along with the bias which is always 1 (b), and connects them to a computational unit called a "neuron". The neuron takes the inputs and outputs the sum of the weighted activations. The perceptron learns to find a set of weights to linearly separate the data in question into two classes. If the sum of the weighted activations is

greater than the threshold, the perceptron will predict class 1, but if not, it will predict class 0 (see Figure 2). After making an incorrect prediction, the perceptron will update its weights using Equation 1, where t is the current iteration, α is the learning rate, z is the actual output, y is the desired output, x is the mislabelled training example

$$w(t+1) = \frac{\partial f}{\partial u}\ w(t) + \alpha\big(y(t) - z(t)\big)\, x(t) \qquad (1)$$

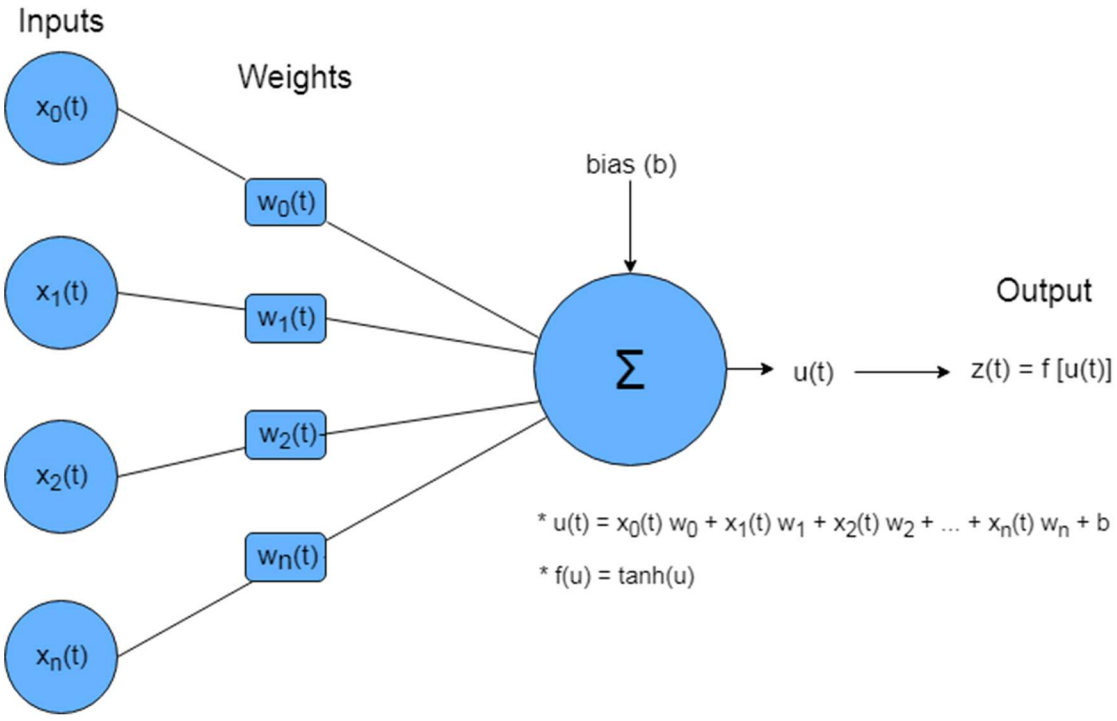

Figure 2. An example of a single layer perceptron with three inputs and a threshold of 0.

After the development of the perceptron, a very simple form of a neural network, neural network research began to increase and once again the excitement surrounding ML and artificial intelligence did as well. However, other researchers questioned whether continued progress could be made with neural networks, given that the perceptron

learning procedure was only capable of learning linearly separable boundaries, and believed that the approach should be abandoned (Minsky & Papert, 1969). Following the theoretical proof of Minsky and Papert (1969), neural network research almost completely stopped except for work conducted by some psychologists. Most researchers returned to symbolic (or classical) artificial intelligence (AI), a form of AI which explicitly represents human knowledge in a declarative form (Olazaran, 1996). During this time other algorithms like k nearest neighbours, a pattern recognition algorithm that classifies data points based on the group membership of its neighbors, was developed (Cover & Hart, 1967).

Neural networks experienced a resurgence in the 1980's primarily due to the development of the backpropagation algorithm (Le Cun, 1986; Rumelhart, Hinton, & Williams, 1986). Much of the initial concern surrounding perceptrons was that they could not be extended to multilayered architectures, and therefore only simple, linearly separable problems could be addressed (Minsky & Papert, 1969). However, backpropagation allows for hidden units (called "hidden" because they lie between input and output layers, and their outputs are not typically observed during training) present in multilayered architectures to learn under what circumstances they should be active to achieve the correct output. To update the parameters of a network, the gradients of the cost function with respect to both the derivative of the weights and biases needs to be calculated for each neuron. The goal is to minimize the error/cost function in weight space using gradient decent. See Appendix A for a detailed description of the backpropagation algorithm.

Following the development of backpropagation, neural network research steadily increased to approximately the same level as 'classic' ML methods. However, neural network researchers were still hindered by the amount of computational power and data required to train larger networks, as well as backpropagation getting stuck in local minima during gradient descent (Tesi & Gori, 1992). Other "classic" methods like random decision forests, which are based on stochastic modelling, and support vector machines (SVM) were developed during this time (Cortes & Vapnik, 1995; Ho, 1995). SVM divide data into categories by attempting to find an optimal hyperplane that creates a large margin between the classes.

Machine Learning and Medical Imaging

ML has been applied to medical images since 1966 for computer aided detection (CADe), but using ML for computer aided diagnosis (CADx) only began to rise in popularity in the early 2000s (Giger, 2018). Some CADx systems, such as one for detecting exudates in colored retinal images for diagnosing diabetic retinopathy, are experiencing great success with accuracies well above 90% (Akram, Tariq, Anjum, & Javed, 2012). Although some groups have found that ML algorithms can achieve higher diagnostic accuracy than physicians (e.g. to detect melanoma (Haenssle et al., 2018; Mar & Soyer, 2018)), most ML algorithms are nowhere near the point where they can be relied upon to independently and accurately diagnose a patient, despite some claims in the popular press (Ng, 2016). These high performing ML algorithms have been able to achieve these results mainly because they have access to hundreds of thousands of quality images to train their models on, whereas most studies/groups do not have access to this

much data. Therefore, although it is possible to achieve high results with ML algorithms, most algorithms do not have enough training data to reach these levels of accuracy and therefore cannot yet be relied upon to make diagnostic decisions in clinical practice. Perhaps in the future if the sharing of medical data is adopted to increase the size of medical databases we could expect more groups to achieve these levels of results.

CADx systems usually involve assessing the structure of interest, and then providing its estimate of disease probability. The physician is then free to use this estimate how they want. To our knowledge, patient management is always left up to the physician, and the CADx systems act more as second opinions. Studies have also shown that the combined synergistic effects of the diagnostic aid and physician knowledge (experience) greatly improved the diagnostic accuracy (Chan et al., 1990; Doi, 2007; Li et al., 2004).

A few groups have successfully implemented ML algorithms as CADx systems, with many of them employing "classic" ML algorithms such as SVM, naïve Bayes classifiers and k nearest neighbours (Akram et al., 2012; Irem Turkmen, Elif Karsligil, & Kocak, 2015; Mudali, Teune, Renken, Leenders, & Roerdink, 2015). Some CADx studies have begun to use deep learning-based ML methods which have shown great promise (Cicero et al., 2017; Song, Zhao, Luo, & Dou, 2017; D. Wang, Khosla, Gargeya, Irshad, & Beck, 2016).

Deep Learning

Deep learning is a general term for an algorithm that trains a many layered (usually greater than 4) network. Deep neural networks are composed of many layers of

artificial 'neurons' which are the elementary units of a neural network. Artificial neurons mimic biological neurons, however, not all aspects of neural firing/propagation are preserved. The dendrite can be thought of as acting as input to a neuron (input value x weight), the soma as the summation function/computation performed by the artificial neuron, and the axon as receiving the output from the soma/summation to propagate along to other layers of artificial neurons.

Deep neural networks learn hierarchical feature representations from raw-data – a type of learning rightfully called 'representation learning' – due to their layered structure. In comparison, conventional or 'classic' ML algorithms, like SVM, often require feature engineering and extraction to transform data into a suitable representation for the algorithm to use. This requirement limits the types of problems that classic ML can be applied to, whereas deep learning models are able to extract useful patterns from raw data on their own (Le Cun, Bengio, & Hinton, 2015). Due to the hierarchical nature of deep learning models, very complex functions can be learned to solve difficult classification problems that were previously unable to be solved by classic machine learning algorithms (e.g. Hinton et al., 2012; Krizhevsky, Sutskever, & Hinton, 2012; Sainath, Mohamed, Kingsbury, & Ramabhadran, 2013).

One example of a deep learning algorithm is the backpropagation algorithm, described above, to update the weights of hidden nodes (see Appendix A). Due to the many layers and nodes in deep networks, high computational power is required to update weights during training, and backpropagation was prone to getting stuck in local minima (Sontag & Hector, 1989; Tesi & Gori, 1992). Therefore, although deep learning models

have been present for quite awhile, training them has only recently become feasible and widespread in part due to the advent of graphics processing units and the availability of larger datasets, but also due to the addition of constraints during training to prevent gradient descent from converging to a local minimum (Krizhevsky et al., 2012; Ruder, 2016).

Deep neural network is a broad term as there are many different 'deep' architectures. Some examples of deep architectures include recurrent neural networks, autoencoders and convolutional neural networks (CNN). CNN are currently the leading architecture for image recognition, classification and detection, and are therefore a very promising option for medical image classification (Le Cun et al., 2015; Le Cun, Bottou, Bengio, & Haffner, 1998).

Convolutional neural networks. CNN are loosely inspired by the visual cortex, specifically the fact that the visual system is a hierarchy where early layers have smaller receptive fields, and the size of receptive fields grows as we introduce more layers (Hubel & Wiesel, 1962, 1968). The general architecture of a CNN is comprised of an input, convolutional layers, pooling, activations and fully connected layers. These architectures can be repeated to create very deep networks (see Figure 3).

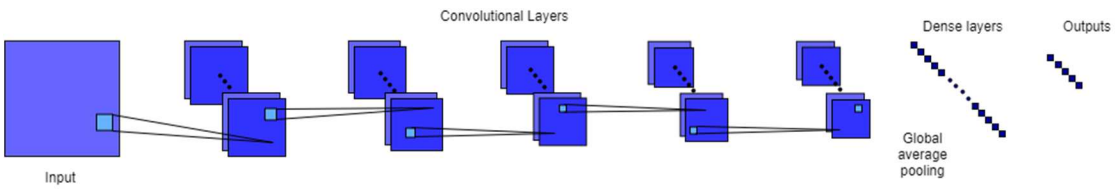

Figure 3. An example of a deep convolutional neural network and its different architectural components.

In convolutional layers filters move across the input image and perform element-wise multiplication. Each layer in a CNN can have many filters, and each separate filter learns and searches for a different image feature. As the filter moves across the image it multiplies the values in the filter with the pixel values of the image. These multiplications are summed producing one number which represents a single location in the image. This process is repeated for every unique location in the input. The resulting array of values that each represent a single location in the image is called a *feature map*, and each separate filter produces its own feature map. Large values in the feature map represent regions in the image where the feature that the filter was searching for was detected. Typically filters in earlier convolutional layers search for simple features (e.g. edges or curves), whereas filters in later layers search for more complex features in the image.

CNN typically apply activations and pooling to the output of convolutional layers. Some of the most common activation and pooling functions are Rectified Linear Units (ReLU), and max pooling respectively. ReLU changes all negative activations (output from a neuron) to 0 which introduces nonlinearity into the system (see Figure 4). This makes the gradients large and consistent. The second derivative of the rectifying operation is 0 almost everywhere, and the derivative of the rectifying operation is 1 everywhere the unit is active making gradient direction very useful for learning (Goodfellow, Bengio, & Courville, 2016). ReLU has been shown to speed up training without negatively impacting accuracy (Glorot, Bordes, & Bengio, 2011). Max pooling reduces the size of the spatial representation and the number of parameters in the network by taking the maximum value of a set sized window and replacing the whole patch by that

value. This helps to reduce overfitting, reduce the size of the representation, and provides a form of translation invariance (Nagi et al., 2011). Max pooling is impacted by the size of the window, and by the stride (i.e. how many pixels the window shifts each time).

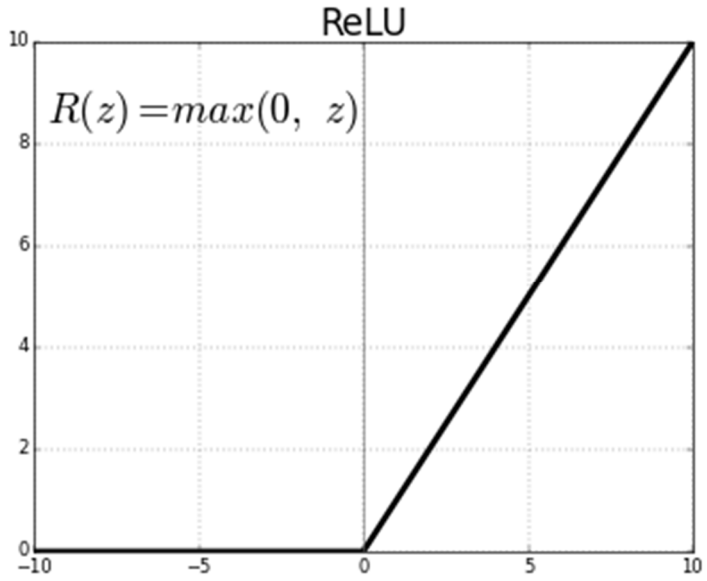

Figure 4. The rectified linear unit (ReLU). Image taken from Eckroth (2017).

After convolutional layers, activation and pooling, the feature maps get flattened to produce a single feature vector to be used by dense layers for classification. Global average pooling can be used in place of flattening to reduce the number of parameters in the network to reduce overfitting. With global average pooling, each feature map is reduced to one number by taking the average of all the values in the feature map which makes it more robust to spatial translations (Lin, Chen, & Yan, 2013). The output of the global average pooling layer can feed directly into the output layer for classification, or into more dense layers. Dense layers are often referred to as fully connected layers because each node in the feature vector connects to each node in the dense layer. Dense

layers perform the actual classification on the features that were extracted by the convolutional layers.

Current Studies

As reviewed in the previous chapter, studies have shown that classifying HN US images is highly subjective, and therefore diagnosis is not consistent across physicians (Rickard et al., 2017). This unfortunate reality means that the standard of care varies according to who interprets a patient's renal US image. The various HN grading schemes do not correlate, and unfortunately there is still no objective grading system.

Diagnostic aids are becoming increasingly popular in medical imaging, and the second opinion that these systems provide has been shown to improve physicians' diagnostic accuracy (Chan et al., 1990; Doi, 2007; Li et al., 2004). Although CNN are very well suited to image classification tasks and tend to perform well, many diagnostic aids employ 'classic' ML models (Le Cun et al., 1998). We hypothesize that HN diagnosis could benefit from a diagnostic aid to provide a reliable second opinion to physicians to help improve diagnostic accuracy, and we specifically believe that a CNN based diagnostic aid is the most promising option.

To our knowledge, developing a CNN based diagnostic aid that can be applied to US images has not been done before, and therefore the current thesis conducted two exploratory studies to assess methodological considerations, namely US image preprocessing and model architecture. Most of the recommendations for US preprocessing are for classic ML algorithms that require feature extraction (e.g. SVM).

Therefore, we investigated whether image segmentation (partitioning and finding regions of interest in the image) and textural extraction, two commonly recommended preprocessing techniques, are beneficial for CNN performance. The second study investigated CNN architectures to understand what components benefited model performance and attempted to find the most optimal CNN architecture for HN classification.

Chapter 3: Database

All data used in this current thesis were from a large database of US images and clinical variables collected by researchers at McMaster Children's Hospital. All data were housed on REDCap™ and were exported and stored on a secure server at McMaster University. The collection of US images and clinical variables, and our subsequent usage of them, was cleared by the Hamilton Integrated Research Ethics Board before the study commenced.

Images

We received a total of 2484 sagittal, 2186 transverse renal US images, 208 bladder US images, and 126 ureter US images from 773 different HN patients from the McMaster University Children's Hospital. Each patient had their US images taken across a variable number of regularly scheduled follow up visits to monitor their HN. The patients ranged in age from 0 to 116.29 months old (M_{age} = 16.53, SD = 17.80). Each image was graded using the SFU and UTD classification system by at least 3 pediatric urologists to maximize the accuracy of the image labels.

Clinical Variables

Along with the images themselves, a total of 22 independent clinical variables were recorded during the patients' visits. Variables marked with (*) were collected during each follow up visit, and included: age at baseline, gender, gestational age, birth weight, circumcision status, laterality, etiology, anteroposterior diameter, SFU grade*, UTD grade*, age*, voiding cystourethrogram (VCUG), age VCUG, vesicoureteral reflux

(VUR) VCUG, age VUR VCUG, surgical status, age at surgery, urinary tract infection status*, continuous antibiotic prophylaxis (CAP)*, CAP type*, breastfeeding status*, percentage of diet breastfed*.

Data Usage

The database is rich with information. However, for our study we only utilized the sagittal and transverse renal US images, and the SFU grades. We decided not to use the UTD grading system, and therefore the US images of the bladders and ureters, because the UTD classification system required images of the kidney, ureters and bladder for each patient. Only a small proportion of patients had all three image types compared to the number of patients who had both types of renal US images. Therefore, by using the SFU grading system we maximized the amount of usable data, which is beneficial for CNN performance.

Chapter 4: Assessing Model Input

Overview

The following study investigated CNN input, with a specific focus on image preprocessing techniques. The purpose was to evaluate whether segmentation and textural extraction, two common preprocessing techniques used for ML and medical imaging, are of any benefit to overall CNN performance.

Methodology

Data

The following study utilized all 4670 renal US images (both sagittal and transverse) and their corresponding SFU grades from the REDCap™ database.

General Preprocessing

The following preprocessing steps were applied sequentially unless otherwise specified.

Cropping. No original renal US images without annotations were available, therefore all images were manually cropped to remove the annotations that were overlaid by radiologists and made grayscale to standardize the colouring of the US images using MATLAB. All general markings (e.g. measurement markers, or the US logo) were always removed, however, some images had annotations overtop of the kidneys themselves, and thus not all annotations could be removed by cropping. See Figures 5 and 6 for an example of the cropping.

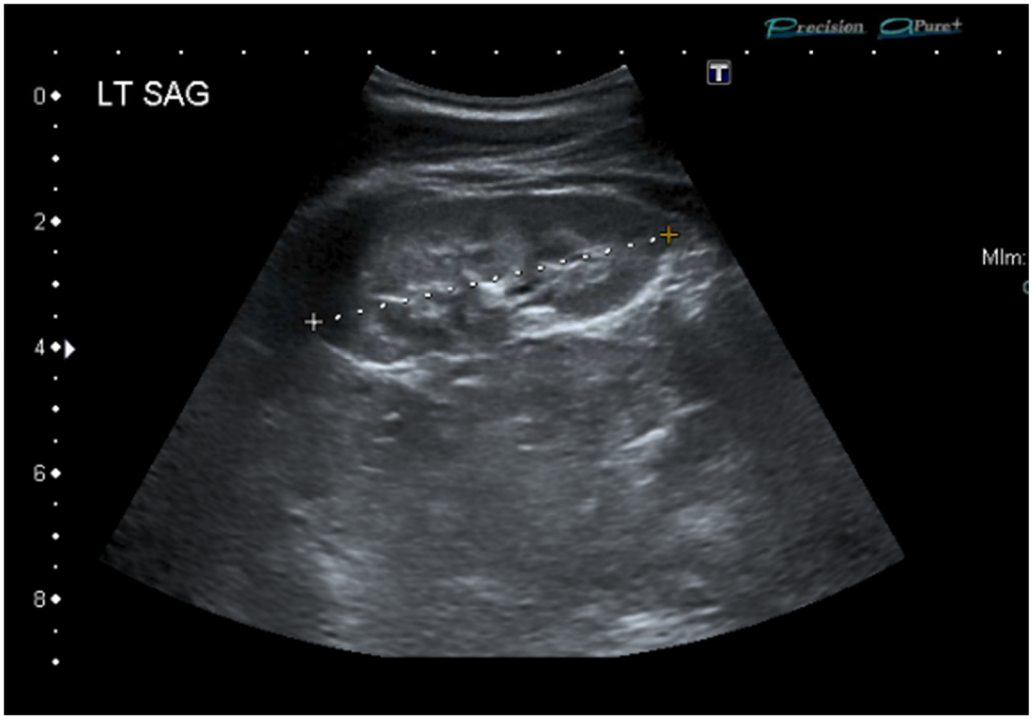

Figure 5. Example of an original renal US with annotations overlaid on top of the kidney itself.

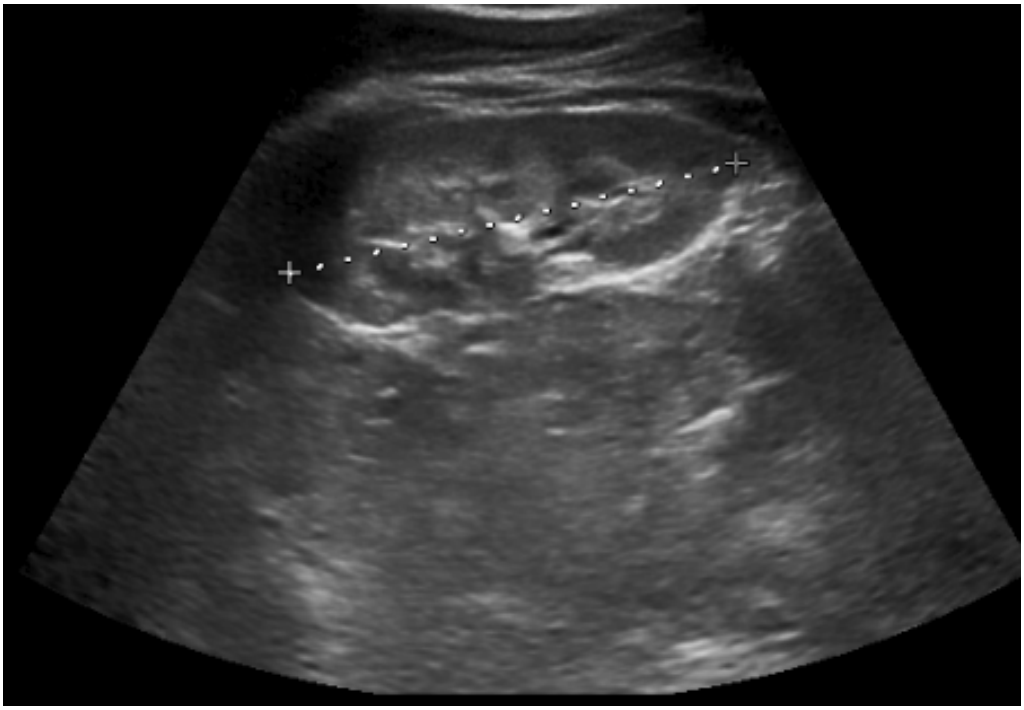

Figure 6. The same renal ultrasound as Figure 5 after the image had been cropped and changed to grayscale. The annotation over top of the kidney itself remains.

Despeckling. Despeckling is a process that removes 'speckle noise', which occurs due to the interference of the returning ultrasound waves at the ultrasound probe. Speckle noise results in white and black 'specks' in regions of the image where they would not be expected, which gives images a granular texture. Despeckling removes the speckle noise which smooths the appearance of the images, however, if too much smoothing occurs, edges are not preserved.

In order to preserve the edges of the kidneys in the US images, a bi-directional FIR-median hybrid despeckling filter first proposed by Nieminen, Heinonen, and Neuvo (1987) was implemented using the "Image Despeckle Filtering Software Toolbox" developed by Loizou et al. (2014). This filter uses a 5 x 5 pixel moving window, and two different sub-filters. The first sub-filter finds the median pixel value along the x-shape of the window, and the second sub-filter finds the median of the pixel values along a cross shape of the window (see Figure 7). The algorithm then takes the median of the first sub-filter value, the second sub-filter value and the centre pixel in the sub-window and finds the median of those three values. It then replaces the value of the centre pixel of the sub-window with the final median value. Combining both of these sub-filters helps to preserve both vertical and horizontal edges (Nieminen et al., 1987). All sagittal US images were despeckled using this method, with two iterations applied over each image. See Figures 8 and 9 for an example of this despeckling method.

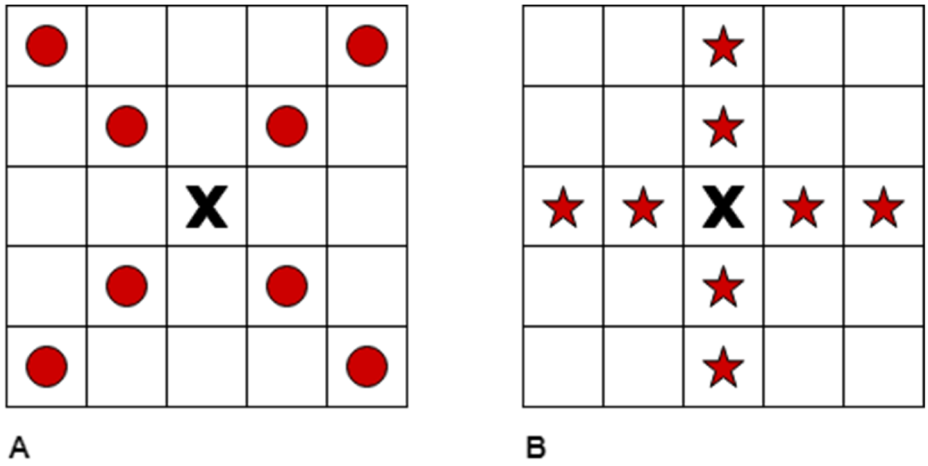

Figure 7. Visualization of the two sub-filters used in the bi-directional FIR-median hybrid despeckling filter. A) The 'x' shaped 5 x 5 sub-filter. The red circles represent the pixels that are included in the calculation of the first sub-filter median. B) The cross shaped 5 x 5 sub-filter. The red stars represent the pixels that are included in the calculation of the second sub-filter median. The black X's represent the centre pixel value in the 5 x 5 window. This centre pixel value will either remain or be replaced.

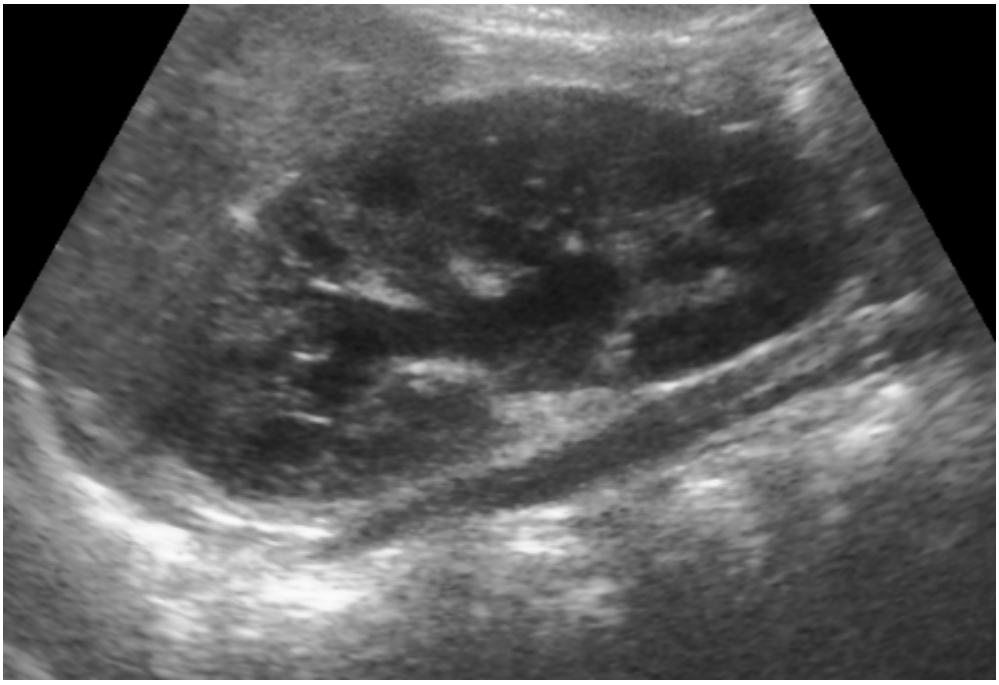

Figure 8. An example of a raw renal US image prior to despeckling.

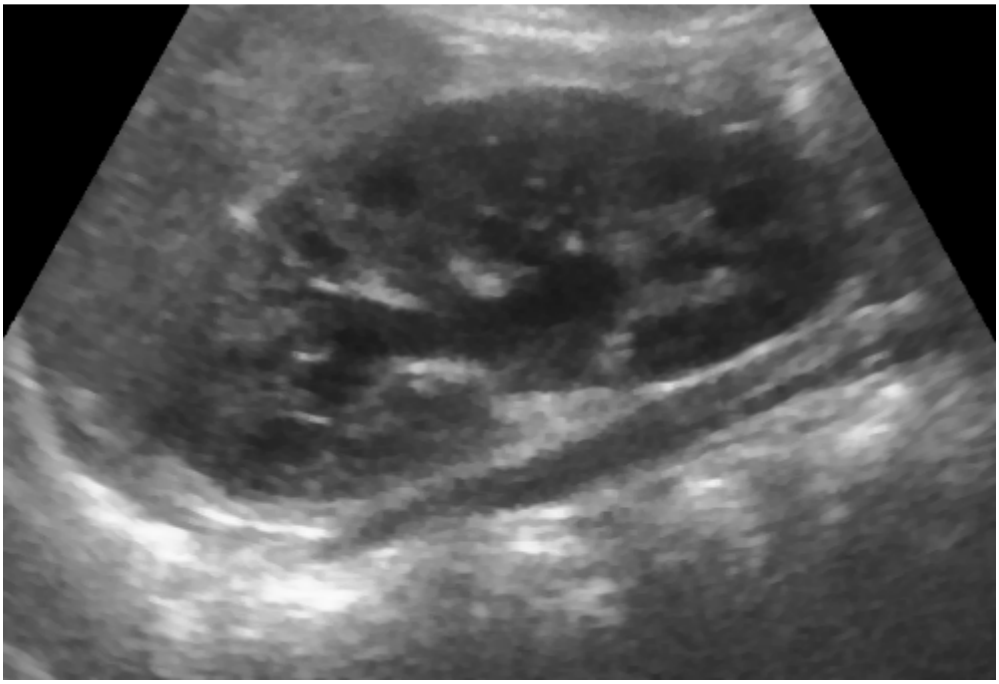

Figure 9. The same renal ultrasound image as Figure 8 after it has been despeckled using the bi-directional FIR-median hybrid filter.

Resizing the images. Both before and after cropping, all renal US images were variable in size. Although it is possible to have varying image sizes as input to a neural network, for simplicity, we chose to resize all images to 256 x 256 pixels to standardize the size without losing too much resolution.

Image segmentation. We explored semi-automatic and manual image segmentation methods. We initially attempted to apply active shape models (ASM) to segment the kidneys from their backgrounds. ASM is typically used for facial recognition or detection, however, there have been some successful applications of ASM in medical imaging (Edwards, Cootes, & Taylor, 1998; Spiegel, Hahn, Daum, Wasza, & Hornegger, 2009; Sun, Bauer, & Beichel, 2012; Wan, Lam, & Ng, 2005; W. Wang, Shan, Gao, Cao,

& Yin, 2002). ASM learns patterns of variability from a subset of annotated images, and deforms from the average shape in ways that are characteristic of the class of objects it represents (Cootes, Taylor, Cooper, & Graham, 1995). Therefore, the model would not deform in ways that are not plausible for a kidney, which we believed would improve the model's ability to correctly segment our kidney US images See Appendix B for a detailed description of the ASM algorithm.

We randomly selected 50 sagittal images (10 from each SFU grade) to test whether ASM could successfully segment the kidney US from our database. After adapting code from Miller (2018) to suit our sample of kidney US, we found that the ASM was not capable of accurately segmenting the kidneys. Due to the high level of variability in the placement, size, and sections of the kidneys from our database, the average shape did not resemble a kidney and therefore was not able to deform in a way that suited all the images in our sample. Since this mode of segmentation was unsuccessful, we instead moved onto manually segmenting the kidneys to assess whether segmentation improved model performance.

We randomly selected 500 sagittal images (100 from each SFU grade) to be manually segmented by a urology resident from McMaster University. Nine of the selected images were poor quality (e.g. the kidney was not visible, or the image was improperly labelled) and were removed, leaving a total of 491 segmented images. All segmented images had an outline drawn overtop indicating where the edges of the kidney were located. We chose 16 points along the outlines and used them to blackout the area surrounding the kidney to remove excess noise. We chose to use 16 points because after

some trial and error, we found that 16 points adequately captured the contour of the kidneys.

Wavelet transformation. Wavelet transformations are widely used in image processing to extract textural information (Baaziz, Abahmane, & Missaoui, 2010; Livens, Scheunders, Wouwer, & Dyck, 1997). They combine frequency filtering with a windowing function and respond to oriented edges at a region in the image. A set of wavelets for different orientations and scales forms a set of basis functions for describing an image. All 491 images that were manually segmented and had their backgrounds blacked out, along with the same set 491 "original" images without any segmentation, underwent a discrete 2-dimensional Daubechies wavelet transform of order 2 using the PyWavelets Toolbox (Lee et al., 2006). The Daubechies wavelets are a family of orthogonal wavelets which defines a discrete wavelet transform, and are characterized by a maximal number of vanishing moments for a given support (Daubechies, 1992). A discrete 2-dimentinal Daubechies wavelet transform is computed by iteratively convolving high and low pass filters with our image (f), followed by down sampling. The low pass filter corresponds to coefficient h_j from the Daubechies scaling function, and the high pass filter corresponds to the coefficient g_h from the Daubechies wavelet function (Natarajan, Casida, Genovese, & Deutsch, 2011):

$$\phi(x) = \sqrt{2} \sum_{j=1-m}^{m} h_j \phi(2x - j) \qquad (2)$$

$$\varphi(x) = \sqrt{2} \sum_{j=1-m}^{m} g_j \phi(2x - j) \qquad (3)$$

The first step of the algorithm performs filtering and down sampling in the horizontal direction:

$$\tilde{a} = (f *^{H} h) \downarrow^{2,H} \quad (4)$$

$$\tilde{d} = (f *^{H} g) \downarrow^{2,H} \quad (5)$$

The second step computes the filtering and down sampling in the vertical direction which produces four separate coefficients:

$$diagonal\ detail\ = (\tilde{a} *^{V} h) \downarrow^{2,V} \quad (6)$$

$$horizontal\ detail = (\tilde{a} *^{V} g) \downarrow^{2,V} \quad (7)$$

$$vertical\ detail = \left(\tilde{d} *^{V} h\right) \downarrow^{2,V} \quad (8)$$

$$approximation = \left(\tilde{d} *^{V} g\right) \downarrow^{2,V} \quad (9)$$

The transformation resulted in four 129 x 129 matrices of the diagonal detail, horizontal detail, vertical detail and approximation coefficients respectively. The approximation coefficients are typically noisy, and therefore they were discarded and not used any further (Mistry, 2013). The horizontal, vertical and diagonal detail coefficients were normalized between -1 and 1 using least absolute deviations method due to its resistance to outliers (Thanoon, 2015). The three coefficient matrices were concatenated for input into the CNN, which resulted in each of the 491 US images being represented by a single 3 x 129 x 129 wavelet coefficient tensor.

Convolutional Neural Networks

We developed a CNN using the Keras neural network API with TensorFlow which contained 4 convolutional layers, a fully connected layer of 512 units, and a final

output layer of 5 units. The architecture was chosen based on models from previous research where CNN were applied to HN US images (Dhindsa et al., 2018). Each layer used rectified linear activation functions (ReLU), except for the last layer which used the softmax activation function for classification. The model was trained to minimize the categorical cross-entropy across classes using stochastic gradient descent with a learning rate of 0.01 and a momentum rate of 0.9 which is applied to speed up learning.

The first convolutional layer contained 16 filters with an input patch of 3 x 3 pixels, and the following three convolutional layers contained 32 filters each with an input patch of 3 x 3 pixels. A stride length of 1 pixel was used in all convolutional layers. Each convolutional layer was followed by max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels.

Batch normalization, a technique whereby the output of the previous layer is normalized by subtracting the batch mean and dividing by the batch standard deviation, was performed after each convolutional layer to make training more robust and efficient. Finally, dropout, a technique where randomly selected neurons are ignored (set to zero) during training, was used for the fully connected layer to reduce overfitting and promote the learning of independent features (Dahl, Sainath, & Hinton, 2013; G. E. Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012; Ioffe & Szegedy, 2015; Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

We inputted 4 different image types into the CNN, resulting in a total of 4 different models (see Table 2). Images that were not wavelet transformed had their pixel values resized between 0 and 1.

Table 2.
The image input types for the four different models.

Input Types
Wavelet transformed (WT) and segmented (S)
Wavelet transformed (WT) and not segmented (NS)
Not wavelet transformed (NWT) and segmented (S)
Not wavelet transformed (NWT) and not segmented (NS)

Results

Each model was evaluated using a 5-fold cross validation loop, which means that we shuffled and split our data into 5 different sections each containing 1/5 of the data. During each of the 5 loops, a different section was used as a test set, while the rest of the data is used to train the network. We used the testing portion of the dataset to test the accuracy of our models by comparing the number of correct responses to the number of incorrect responses. This process was repeated 5 times. All images belonging to a patient always remained in the same set to ensure that within-patient similarities would not cause the model to overfit. We evaluated model performance using the average accuracy across the 5 folds and the average F1 score across the five different test sets. F1 is the weighted average of precision and sensitivity (see Formulas 8, 9 and 10), which provides a better

assessment of the model's performance than a receiver operator characteristic curve when there is an imbalance in the number of samples in each class (Davis & Goadrich, 2006).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (10)$$

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (11)$$

$$F_1 = 2 \cdot \frac{precision \cdot sensitivity}{precision + sensitivity} \quad (12)$$

All input types were evaluated on their ability to classify all SFU grades, since the goal is to eventually develop a diagnostic aid for HN that can help guide physicians towards the correct diagnosis. Therefore the, ML algorithm must be evaluated on its ability to correctly classify all 5 SFU grades. The CNN performance is summarized for all input types below (See Table 3). All four models performed similarly in terms of average accuracy and F1 score, and all models significantly overfit. See Figures 10, 11, and 12 respectively for an example confusion matrix, accuracy per epoch and loss per epoch for a typical cross validation run.

Table 3.
5-way SFU classification results. Average accuracy is reported as a percentage with the standard deviation of the accuracy scores across the 5 folds. S represents segmented, NS represents non-segmented, WT represents wavelet transformed and NWT represent non-wavelet transformed.

Input Type	Accuracy	F1
WT & S	39 $\pm$ 6	0.39
WT & NS	46 $\pm$ 4	0.45
NWT & S	44 $\pm$ 4	0.44
NWT & NS	43 $\pm$ 2	0.42

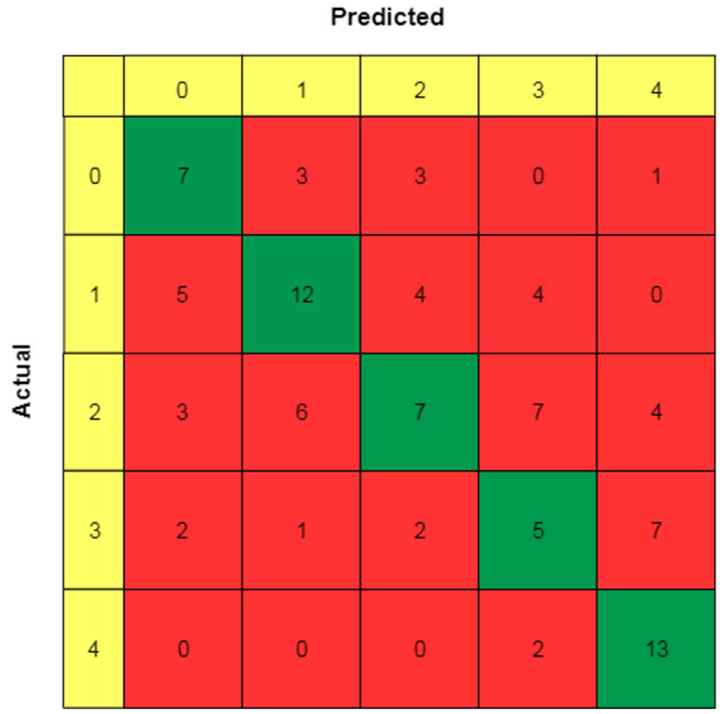

Figure 10. Example confusion matrix from a typical cross-validation run when performing 5-way classification using the non-segmented and non-wavelet transformed images.

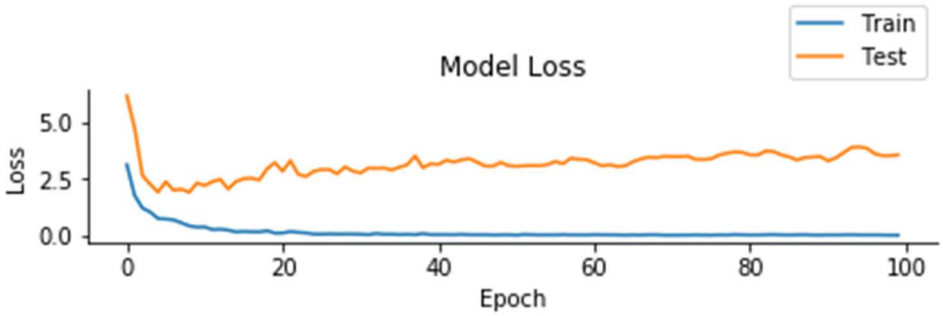

Figure 11. The categorical cross entropy loss from a typical cross-validation run when performing 5-way classification using the non-segmented, non-wavelet transformed images.

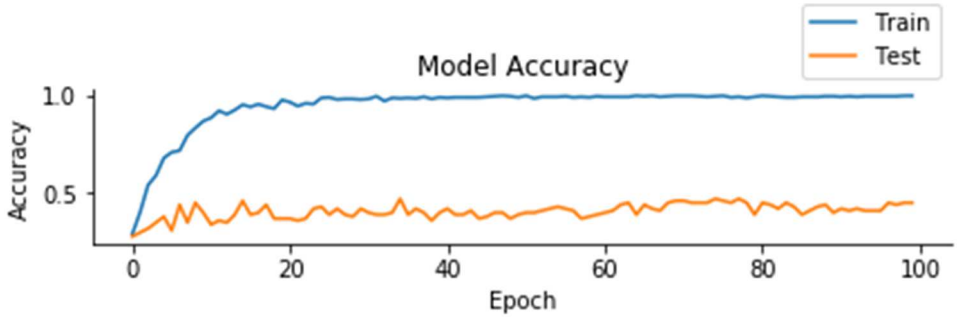

Figure 12. Accuracy from a typical cross-validation run when performing 5-way classification using the non-segmented, non-wavelet transformed images

Discussion

The purpose of the current study was to assess whether segmentation and the extraction of textural information would improve the performance of a CNN. Our results show that these extra preprocessing steps do not improve model performance in terms of average accuracy, or F1 scores. Instead, we found that all four different input types performed similarly. It had been previously thought that segmentation, along with extracting textural information, are vital steps towards developing any sort of diagnostic

aid since the shape of the tissue of interest often provides important clinical information to the physician (Jalalian et al., 2013). However, our results suggest that we may be able to bypass segmentation when using a CNN to classify medical images.

Within the literature, the recommendations for medical image preprocessing (e.g. segmentation and texture extraction) have been dominantly geared towards 'classic' machine learning methods such as SVM, or imaging techniques other than ultrasound (Huynh, Li, & Giger, 2016; Jalalian et al., 2013). The application of deep learning in medicine is on the rise, however, to our knowledge, many of the recommendations for preprocessing have not changed or been updated in a similar fashion. CNN are translationally invariant, extract low level features in early layers of the model, and allow an entire image to be used as input. In comparison, other ML models require features from regions of interest to be extracted. Therefore, segmentation and textural extraction in general in combination with CNN seems rather redundant. Based on this, from a pure deep learning perspective, it is not surprising that we were able to move forward and get comparable model behavior without these extra preprocessing steps.

However, when we take the clinical perspective and consider the SFU classification system which only describes the features directly associated with the kidney itself (e.g. parenchymal thickness, calices), it may seem unclear why segmenting the images would not improve model performance (Fernbach et al., 1993). Segmentation removes background noise/information, which based on the SFU classification system, should be irrelevant. However, perhaps this background information within the US images is important and helps physicians with classification. We found these results surprising

considering the numerous studies that support segmentation and textural extraction and therefore decided to use a smaller sample of images to rule out whether background noise is present but just gets averaged out with more images. We found that when we only used 10 randomly selected images of each SFU grade as input to our models (total of 50 images), each of the different input types produced similar model performance. If the background of the US images was in fact noise, we would have expected decreased performance from the models using non-segmented images since the background noise would be less likely to be averaged out and would therefore have more of an influence on what the CNN is learning. However, these smaller models produced the same pattern of results as the larger models, providing further support that the background information does not hinder model performance.

Anecdotal evidence from one of the urologists at McMaster University who graded the renal US images revealed that some physicians, including himself, do in fact use the background information when interpreting renal US images. Although not directly related to the features of the SFU grading system, the urologist stated that they use the density (i.e. how light or dark) of the liver and spleen are in comparison to the kidney to determine how echogenic (i.e. ability to bounce an ultrasound wave) the kidney is. The echogenicity of the kidney indicates whether the kidney has scar tissue. Although this account comes from a single physician and does not directly relate to the SFU classification system, based on our segmentation findings, it may be beneficial to assess the usage and possible benefit of including background information into classification/grading guidelines. It has been shown that expert physicians are poor at

articulating which individual diagnostic features they are attending to, and instead come to a diagnosis more holistically (Eva, Norman, Neville, Wood, & Brooks, 2002). Therefore, it is possible that using background information when interpreting HN US images is common, but physicians do not articulate all components that were considered when rendering a diagnosis, and thus these components are not well documented. If this is the case, the usage of background HN US image information is not standardized within any of the current HN classification systems, which could be attributing to the poor inter-rater reliability of HN.

Limitations and Future Directions

The current study only used 491 images in total, and only 393 of those images were used to train the model during each cross-validation loop. We did find that all models significantly overfit which might suggest that it is simply memorizing the data. However, based on the confusion matrices from each input type we can see that the model was not randomly guessing a single class and was instead predicting all 5 classes with a similar frequency, and consistently achieved better than chance level accuracy for all input types. Therefore, although the model had less data than what might typically be used for this kind of approach, it does seem like it was able to learn. Considering the depth of our network, it would be beneficial to train the model with a larger number of images to ensure that the model behaves similarly and that there are still no differences between the four input types.

Although the size of the dataset is an important factor to be aware of, being able to achieve similar or better model results without segmentation or preprocessing is a very useful, informative and promising finding. Segmentation can be a very cumbersome task, especially when there is high variability between the images. Therefore, bypassing this step can save a substantial amount of time that can otherwise be used to fine tune model architectures. Finally, these findings have suggested that an investigation of the importance of the background information in the renal US images might be beneficial. Along with re-running these models with a larger subset of data, further studies should be conducted to determine whether physicians and/or CNN utilize background US information when interpreting and classifying HN US images. If physicians do regularly use background information, it would help to explain why inter-rater reliability is so poor. Furthermore, if CNN find relevant information from the background of HN US images, it suggests that these diagnostic features should be standardized for physicians to use.

Chapter 5: Investigating Convolutional Neural Network Architectures

Overview

The following study investigated CNN architectures with the goal of maximizing model performance in terms of accuracy and F1 score (average of precision and recall) for classifying kidney ultrasound images into the 5 SFU grades. This was an exploratory study and all models were developed sequentially, meaning that changes were made to iteratively improve upon the results of previous models. We have included the original model architectures, along with all noteworthy models that were developed as we progressed towards finding the best model architecture.

Methodology

Database

We used the same database of renal US images as we did in Chapter 3 for a total of 2484 sagittal, and 2185 transverse US images from 773 different HN patients. Each image was graded using the SFU classification system by at least 3 pediatric urologists to maximize the accuracy of the image labels. We compared models that were either non-fused single input models, or fused models that used both sagittal and transverse US as input. All 4670 images were used as input into the single input CNN models. For our fused CNN, 2016 sagittal and 2016 transverse images were used as paired input to our models. Each pair of images was from the same patient and the same visit. 638 sagittal and transverse US were not used for these models because they did not have a corresponding partner to be a used as input with.

Preprocessing

All images were cropped, despeckled, and resized in the same manner as in Chapter 3, however based on the results of study 1, neither wavelet transformation nor segmentation improved model performance, so the images were not segmented, or wavelet transformed. All images had their pixel values rescaled between 0 and 1, and then normalized to have a mean pixel of zero. Finally, each pixel value was divided by the standard deviation across all pixels in the corresponding image.

Data Augmentation

Although our database is large within the domain of HN, as well as in comparison to other diagnostic aid studies, it is still a small amount of data by machine learning standards. This problem becomes even more apparent when we consider the number of classes, and the amount of variability within each of the classes. When there is insufficient data, ML models in general are at a higher risk of simply memorizing the training data rather than learning generalizable rules from it, resulting in overfitting where model performance decreases when applied to new data.

Data augmentation was applied to artificially expand the dataset by applying affine transformations to it, and can be used to reduce overfitting in small data problems (Cireşan, Meier, & Schmidhuber, 2012, p.; Krizhevsky et al., 2012; Simard, Steinkraus, & Platt, 2003). The variance introduced by the transformations allows the model to learn invariant features and generalize better to testing data. Some of the models' data were augmented by applying rotations of up to 45º, horizontal and vertical flips, as well as

width and height shifts of up to 20% (51 pixels). Not all models used data augmentation, and data augmentation was only applied to the training data and not the testing data. We will note which models used data augmentation.

Single Input Convolutional Neural Networks

All models were trained to minimize the categorical cross-entropy across classes using the nonlinear optimization algorithm Adam (Kingma & Ba, 2014); unless otherwise indicated, all models were trained with a learning rate of 0.01 and a decay of 10^{-6}. The decay parameter reduces the weight vectors towards zero during each iteration which stabilizes the learning process. After the final convolutional layer, global average pooling was applied to decrease the number of parameters in the model and prevent overfitting in each model. Finally, each layer in all models used a rectifying nonlinearity (ReLU), except for the last layer which used the softmax activation function for classification. All models were created using the Keras neural network API with TensorFlow.

Model 1. The CNN contained 5 convolutional layers, a fully connected layer of 512 units, and a final output layer of 5 units (see Figure 13 for a visual representation of the architecture). The first convolutional layer contained 16 filters, the second and third layer contained 32 filters and the fourth and fifth layer contained 64 filters. The filters in all layers had input patches of 3 x 3 pixels, and a stride length of 1 pixel was used in all convolutional layers. Each convolutional layer was followed by batch normalization and then max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels. A dropout of

0.5 was used for the fully connected layer to reduce overfitting and promote the learning of independent features. Data augmentation was not applied.

Model 2. The number of dense nodes was changed from 512 units to 400 units. All other aspects of Model 2's architecture was the same as Model 1. Data augmentation was not applied.

Model 3. The number of dense nodes was changed from 400 units to 350 units All other aspects of Model 3's architecture was the same as Model 1. Data augmentation was not applied.

Model 4. Based on the performance of Models 1-3, Model 3 was dropped from further consideration. The architecture of Model 4 was the same as Model 1 except data augmentation was applied to the inputs.

Model 5. The architecture of Model 5 was the same as Model 2 except data augmentation was applied to the inputs.

Model 6. Since Model 5 was the current best performing architecture, Model 4 was dropped from consideration. We simplified the architecture of Model 5 by decreasing the number of convolutional layers. Model 6 contained 4 convolutional layers, a fully connected layer of 400 units, and a final output layer of 5 units. Model 6 used a learning rate of 0.01 and a decay of 1e-4.

The first convolutional layer contained 8 filters, the second and third layer contained 16 filters and the fourth layer contained 32 filters. The filters in all layers had input patches of 3 x 3 pixels, and a stride length of 1 x 1 pixel was used in all

convolutional layers. Each convolutional layer was followed by batch normalization and then max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels. A dropout of 50% was used for the neurons in the fully connected layer. Data augmentation was applied to the input images.

Model 7. Rather than simplifying a CNN by decreasing the number of layers, we decreased the number of filters in each layer. Therefore, Model 7 still had 5 layers, a fully connected layer of 400 units, and an output layer of 5 units. The model was trained with a learning rate of 0.01 and a decay of 10^{-6}.

The first and second convolutional layers contained 8 filters, and the third, fourth and fifth layers contained 16 filters. The filters in all layers had input patches of 3 x 3 pixels, and a stride length of 1 x 1 pixel was used in all convolutional layers. Each convolutional layer was followed by batch normalization and then max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels. A dropout of 0.5 was used for the fully connected layer, and data augmentation was applied to the inputs.

Model 8. The architecture was the same as Model 5, except class weights were introduced to adjust for imbalanced classes – although the imbalance was relatively small. Class weights influence the magnitude of the gradient calculated during backpropagation. We used SFU class 3 as a reference since it was the most common class, and calculated ratios across the whole dataset to increase the importance of the under-represented classes. The ratios were as follows: 1:3.68 (SFU 0), 1:1.76 (SFU I), 1:1.16 (SFU II), 1:1 (SFU III), 1:2.46 (SFU IV). The model was trained to minimize the *sparse categorical*

cross-entropy, which is used when data is labelled using integers rather than one-hot encoding, across classes using Adam with a learning rate of 0.01 and a decay of 1e-6. Data augmentation was applied to the inputs.

Fused Convolutional Neural Networks

Fused CNN were investigated to see whether providing two different US views of the kidney from the same patient during the same visit would improve performance. Physicians usually have access to multiple different US images of kidneys at different angles and are therefore able to come to a diagnosis by combining information from multiple views. Similarly, it was hypothesized that providing two different US views to a CNN to simultaneously process would allow the network to learn correlations between the images, thus improving classification performance (Dolata, Mrzygłód, & Reiner, 2017, p.). All models were created using the Keras neural network API with TensorFlow.

Model 9. The Fused CNN contained 5 convolutional layers in total, a fully connected layer of 400 units, and a final output layer of 5 units. After the final convolutional layer, global average pooling was applied to reduce the number of parameters in the model. Each layer used ReLU, except for the last layer which used the Softmax activation function for classification. The model was trained to minimize the categorical cross-entropy across classes using Adam with a learning rate of 0.01 and a decay of 1e-5.

The first convolutional layer was a shared layer containing 16 filters each with input patches of 3 x 3 pixels, and a stride length of 1 x 1 pixel. This shared layer was applied to both inputs (sagittal and transverse images). Therefore, the shared filter was

learning the same features within both images, however they were not overlapping and learning from both the sagittal and transverse images. After both inputs had been convolved, the output was concatenated, batch normalization was performed, and max pooling with a 3 x 3 pixel input and a stride of 2 x 4 pixels was implemented to reduce the output to a square matrix.

Layers two and three contained 32 filters, and layers four and five contained 64 filters. Each filter had input patches of 3 x 3 pixels, and a stride length of 1 x 1 pixel. Each of these convolutional layers was followed by batch normalization and then max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels. A dropout of 0.5 was used for the fully connected layer, and data augmentation was applied to both sets of inputs. See Figure 14 for a visual representation of this architecture.

Model 10. We simplified the architecture by decreasing the number of filters in each of the convolutional layers, however, all other aspects of the architecture were the same as Model 9. The first shared convolutional layer contained 8 filters, the second and third layers contained 16 filters, and the fourth and fifth layers contained 32 filters.

Model 11. We increased the number of filters in each convolutional layer. The first shared convolutional layer contained 12 filters, the second and third layers contained 24 filters, and the fourth and fifth layers contained 48 filters. All other aspects of the architecture were the same as Model 9.

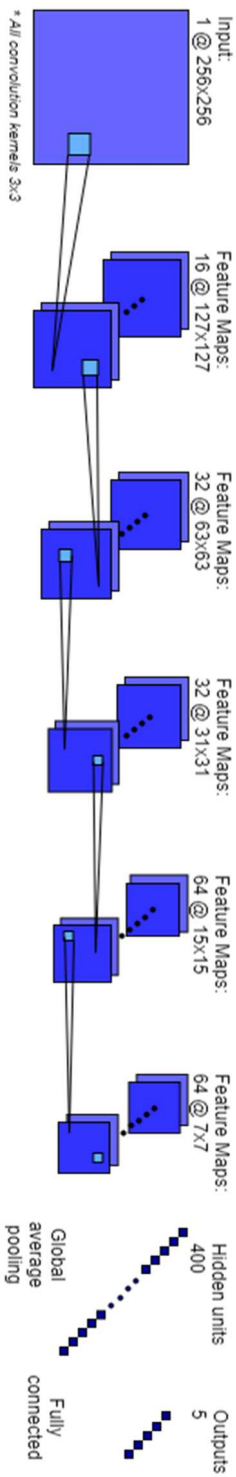

Figure 13. A visual representation of Model 1's architecture, including all convolutional and dense layers.

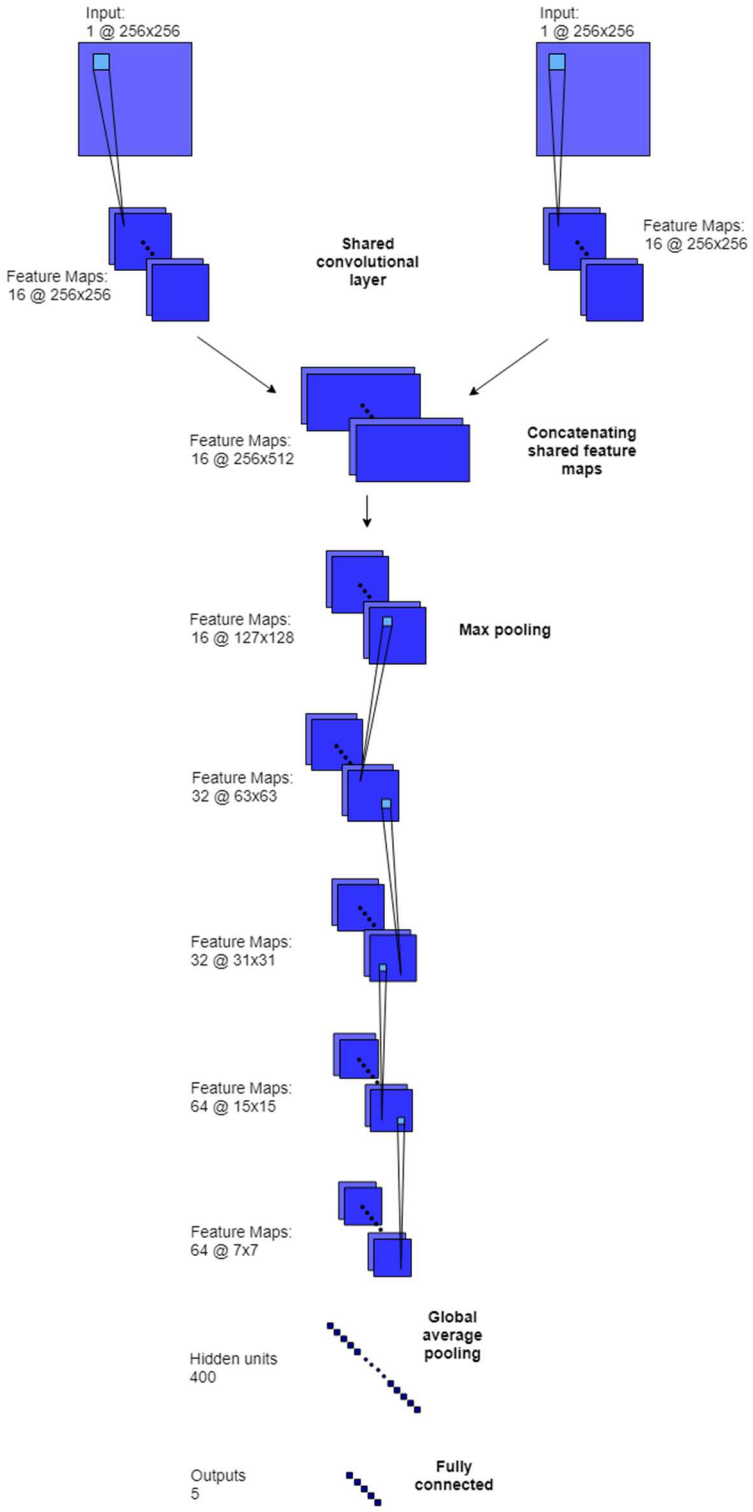

Figure 14. A visual representation of Model 9's fused architecture, including all convolutional and dense layers.

Model 12. To investigate whether the generalization would improve if the model was trained on simultaneously presented stereo image pairs, the fused CNN was trained on two separate streams of data fused after each of their respective third convolutional layers. Each input passed through 5 convolutional layers, a fully connected layer of 400 units, a fully connected layer of 200 units, and a final output layer of 5 units. After the final convolutional layer, global average pooling was applied to reduce the number of parameters in the model. Each layer used ReLU, except for the last layer which used the softmax activation function for classification.

The first layer for each input consisted of 8 filters, followed by batch normalization and then max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels. The second layer in both streams contained 24 filters, followed by batch normalization and then max pooling with 3 x 3 pixel input and a stride length of 2 x 2 pixels. In each stream, layer three contained 24 filters. After this set of convolutions, the outputs from both streams were concatenated together and then underwent batch normalization and max pooling with 3 x 3 pixel input and a stride length of 2 x 4 pixels to reduce the output to a square matrix. Layers four and five both contained 48 filters and were each followed by batch normalization and max pooling with 3 x 3 pixel input and a stride length of 2 x 2. A dropout of 0.5 was used for the fully connected layer, and data augmentation was applied to both sets of inputs.

Model 13. We increased the number of filters in each convolutional layer. Convolutional layer 1 in both streams contained 16 filters, convolutional layers 2 and 3

contained 32 filters, and convolutional layers 4 and 5 used 64 filters. The rest of Model 13's architecture was the as Model 12.

Model 14. An extra dense layer of 200 units was added after the first dense layer of 400 units. All other architectural aspects were the same as Model 13.

Model 15. We changed the number of dense units in the first dense layer from 400 units to 500 units. All other architectural aspects were the same as Model 14.

Results

Each model was evaluated using a 5-fold cross validation loop. The images were shuffled and then split into five sets, and all images belonging to a patient always remained in the same set to ensure that within-patient similarities would not cause the model to overfit. Each set was used as the test set once, and in that case the remaining four sets were combined in to the corresponding training set. We evaluated model performance using the average accuracy and the average F1 score across the five different test sets (see Tables 4 and 5).

Single Input Models

Model 5, which used data augmentation during training and contained 400 dense units, was the best performing model with an average accuracy of 49% $\pm$ 1 and an average F1 score of 0.48 (see Table 4). Model 5 only made errors greater than two grades (e.g. predicted SFU grade I, but the correct class was SFU grade III) 8% of the time, and

the rest of the predictions (43%) were clustered within 1 grade of the correct SFU diagnosis (see Figure 15). Finally, Model 5 did not overfit (see Figure 16 and 17).

Table 4.
5-way SFU classification results for the single input CNN. Average accuracy is reported with the standard deviation of the accuracy scores across the 5 folds.

Model	**Description**	**Accuracy**	**F1**
Model 1	512 dense units	44 ± 2	0.44
Model 2	400 dense units	44 ± 2	0.42
Model 3	350 dense units	43 ± 2	0.42
Model 4	512 dense, data augmentation	48 ± 1	0.45
Model 5	400 dense, data augmentation	**49 ± 1**	**0.48**
Model 6	Fewer layers	45 ± 1	0.42
Model 7	Fewer filters	47 ± 1	0.44
Model 8	Class with fewer images weighted higher	46 ± 2	0.45

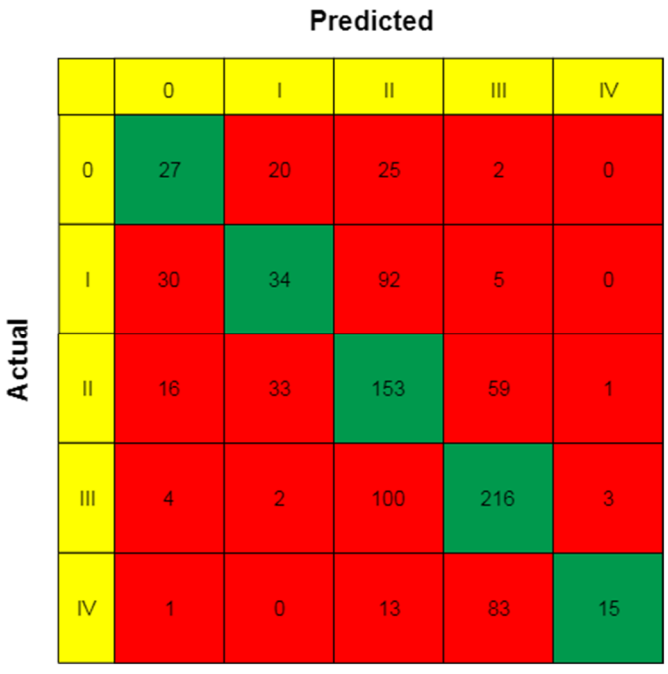

Figure 15. Example confusion matrix from a typical cross-validation run when performing 5-way classification using Model 5.

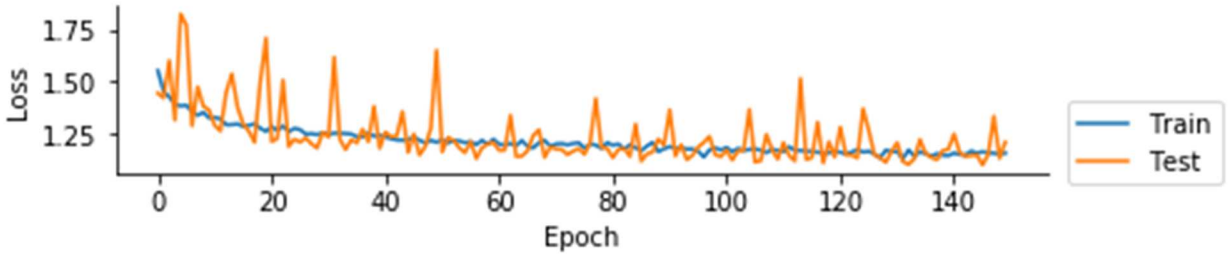

Figure 16. Loss from a typical cross-validation run when performing 5-way classification using Model 5, the highest performing single input CNN.

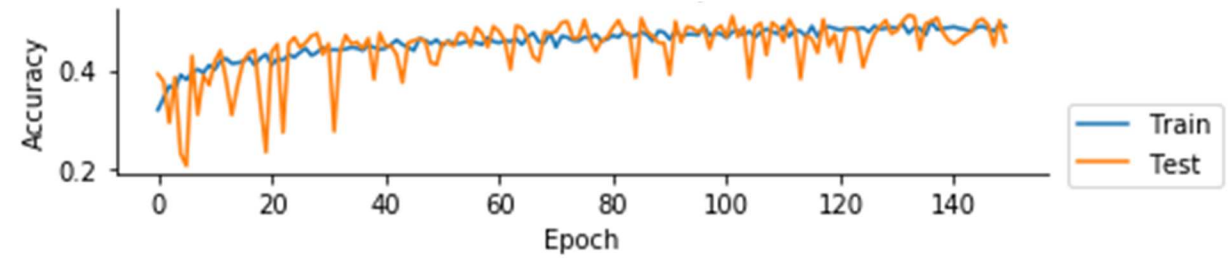

Figure 17. Accuracy from a typical cross-validation run when performing 5-way classification using Model 5, the highest performing single input CNN.

Fused Models

Model 9, our original fused model, was the top performing fused model with an average accuracy of 49% $\pm$ 2 and an average F1 score of 0.48 (see Table 5). Model 9 made errors greater than 2 grades 10% of the time, and the rest of the predictions (41%) clustered within 1 grade of the correct SFU diagnosis (see Figure 18). Model 9 did tend to overfit by 4-11% between model training and testing (see Figure 19 and 20).

Table 5.

5- way SFU classification results for the fused CNN. Average accuracy is reported as a percentage with the standard deviation of the accuracy scores across the 5 folds.

Model	**Description**	**Accuracy**	**F1**
Model 9	Original fused model	**49 ± 2**	**0.48**
Model 10	Fewer filters per layer	48 ± 4	0.47
Model 11	Moderate filters per layer	49 ± 3	0.46
Model 12	Fused later, 400 and 200 dense units, fewer filters	47 ± 4	0.44
Model 13	Fused later, 400 and 200 dense units, more filters	49 ± 2	0.47
Model 14	Dense layers of 500 and 200	48 ± 2	0.43
Model 15	Moderate filters, dense layer of 500	47 ± 3	0.45

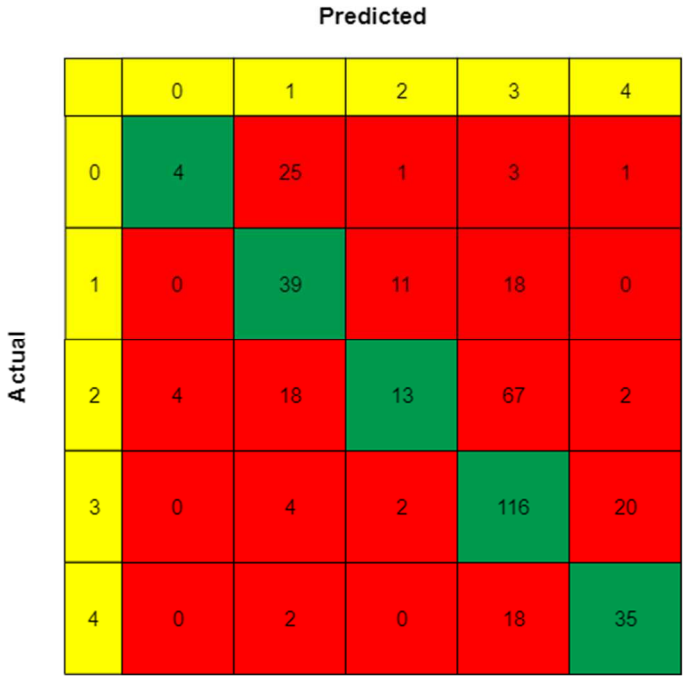

Figure 18. Example confusion matrix from a typical cross-validation run when performing 5-way classification using Model 9.

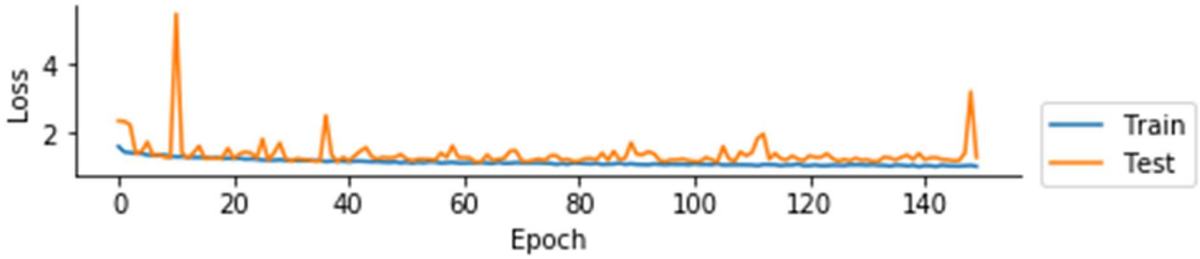

Figure 19. Loss from a typical cross-validation run when performing 5-way classification using Model 9, the highest performing fused CNN.

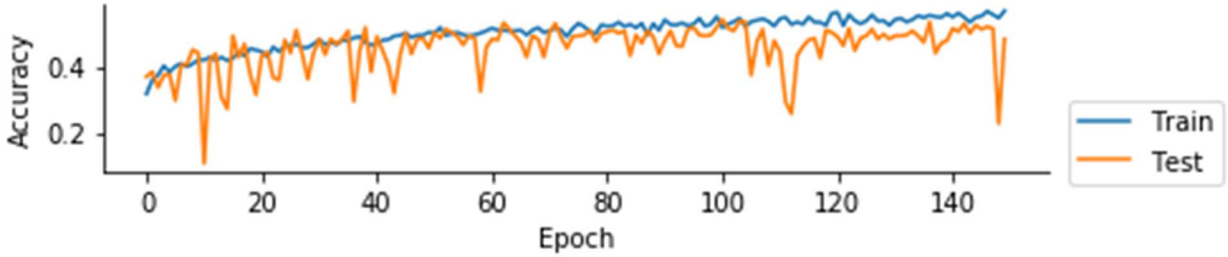

Figure 20. Accuracy from a typical cross-validation run when performing 5-way classification of Model 9, the highest performing fused CNN.

Discussion

Comparing Top Model Performances

The top performing model overall was Model 5, our best single input model, and the top performing fused model was Model 9. Considering the standard deviations of the accuracies, both models performed similarly, which is surprising considering simultaneous processing of two different views of the same object usually improves performance (Dolata et al., 2017). Although model performance in terms of accuracy and F1 scores were similar between these two models, there are important differences between them that must be considered.

Model 9 was more complex than Model 5. It required more input images, merged two separate streams of data, and required an extra and larger dense layer to reach a comparable accuracy to Model 5. These characteristics meant that Model 9 had a greater number of parameters, making it prone to overfitting, which was what we observed (see Figure 11). The overfitting in Model 9 was moderate, with a difference in accuracy of approximately 4-11% between training and testing, however, it was still present, unlike in the more simplified Model 5.

Along with Model 9 having a greater number of parameters, it also requires two paired images as input for both training and testing. Using two images per training/testing example effectively reduced our dataset by a factor of two, which could also explain the overfitting we observed in Model 9. When there is insufficient data, ML models in general are at a higher risk of simply memorizing the training data rather than learning from it, thus resulting in overfitting. This phenomenon is especially true of neural network-based models and is directly related to the large numbers of parameters in these types of models. Therefore, the fact that Model 9 contains more parameters *and* requires more data makes Model 9 a less effective model.

Although both models performed similarly in terms of accuracy and F1 score, Model 9 also made a larger number of errors greater than two SFU grades than did Model 5. Therefore, taking all of this into consideration, it does appear that Model 5 is superior. Model 9 is an expensive model to train and US images, or any medical images, do not come cheaply. Our dataset is small by deep learning standards since medical data is not easy to come by. Until we can share medical data across institutions to increase the size

of medical databases, CNN and other ML algorithms must be applied with care to these datasets. Therefore, to maximize our database and produce fewer larger errors, we believe that Model 5 is our current best option.

We developed many CNN, both single input and fused, but our search for the best CNN was not exhaustive. Model 5 can likely still be improved with an exhaustive search; however, we believe that the results of our model architecture exploration provide an excellent starting point to begin to fine tune both the architecture and hyperparameters of Model 5. Various incremental changes were made to our models to arrive at our best model (Model 5), and we will discuss the important changes to understand why they did or did not improve our models' performances.

Model Changes and Their Effect on Performance

Number of dense layer nodes. In both our single input and fused models, we varied the number of nodes in our first dense layers. Three separate models were developed to assess how many dense nodes would result in the best model performance. We assessed models with 512, 400 and 350 nodes. We found that the Model 3 with 350 dense nodes performed the worst out of these three, however, the Models 1 and 2 with 512 and 400 nodes respectively performed similarly. However, both Models 1 and 2 were still overfitting, therefore we continued to investigate architectures with both 512 and 400 dense nodes and other techniques were applied to minimize overfitting.

Data augmentation. We began to augment our training data by applying flips, rotations and shifts in Model 4, and found that it provided the largest increase in model

performance. Therefore, all the following models applied data augmentation to the training data. Applying these augmentations to our training data artificially expanded the dataset and introduced more variability. Overfitting can occur when models learn/memorize specific instances of the data. Introducing variability and expanding our training set provided more training samples for our model, which increased the number of backpropagation iterations, but is also required our model to learn general rules since the input images were all augmented differently and could not be simply memorized. Since the model learned rules rather than instances, it was better able to generalize to our testing data. Data augmentation eliminated any overfitting in model 5, and increased model performance, therefore, all subsequent models applied data augmentation to the input images. After testing data augmentation on models with 400 and 512 dense units, it was found that the model with 400 dense units performed best, therefore all subsequent models had 400 units.

Simplifying our models. The first few single input models used five convolutional layers. This was based on a previous model that was developed for HN US classification (Dhindsa et al., 2018). However, five convolutional layers is quite deep, therefore we attempted to simplify the models by reducing the number of convolutional layers. After reducing the model to four convolutional layers we found that model performance noticeably dropped. We hypothesize that this could be because with fewer layers, the feature maps of the final convolutional layer were larger (15 x 15 pixels). Therefore, the receptive fields of the final convolutional layer of this shallower model were smaller and couldn't learn as complex of features as the deeper models.

To accommodate this possibility, we also briefly investigated whether reducing the number of filters in each layer, rather than reducing the number of layers themselves, would benefit model performance. Reducing the number of filters slightly reduced model accuracy. Extracting more features from the input images resulted in improved performance and therefore we did not continue to reduce the depth of our model, or the number of filters in each layer.

Misclassified Images and Their Implications

Most of the images in our dataset were classified correctly or within one value of their correct grade by our best performing model (Model 5). However, 8% of the time images were classified greater than two values away from their correct grade. The fact that such a small percentage of images were misclassified in this manner is very positive, however, we were curious why our model was making such large mistakes on such a small subset of images.

We visually inspected the images that were badly misclassified and although there were some where our model simply made the wrong diagnosis, there were several instances where Model 5 did in fact make the correct diagnosis. However, the provided label was incorrect, and therefore the model's diagnosis was deemed 'incorrect'. All observed instances of mislabelling were confirmed by a senior pediatric urologist. Some of the instances were obvious. For example, one of the kidney US was SFU grade 4, and was predicted as such, however it was labelled as SFU grade 0 (see Figure 21). Other instances were slightly less extreme (see Figure 22).

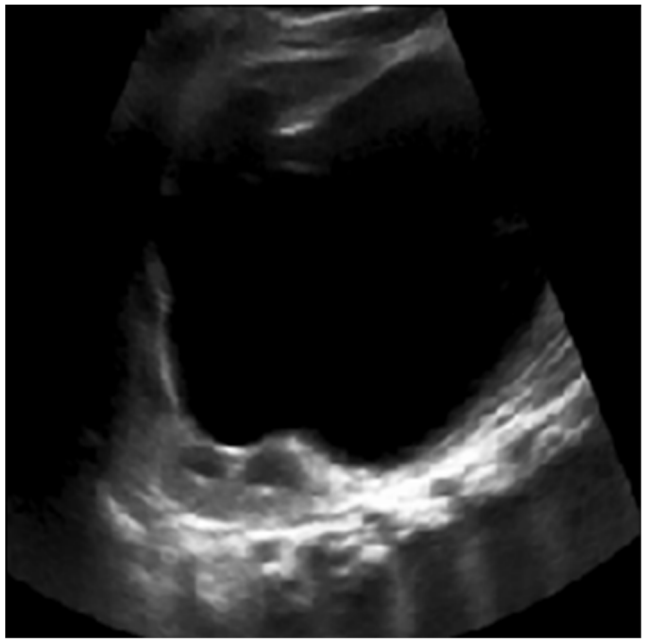

Figure 21. An US image that was classified as SFU grade 4 by our model, which was the correct grade, however, the supplied grade label was SFU 0.

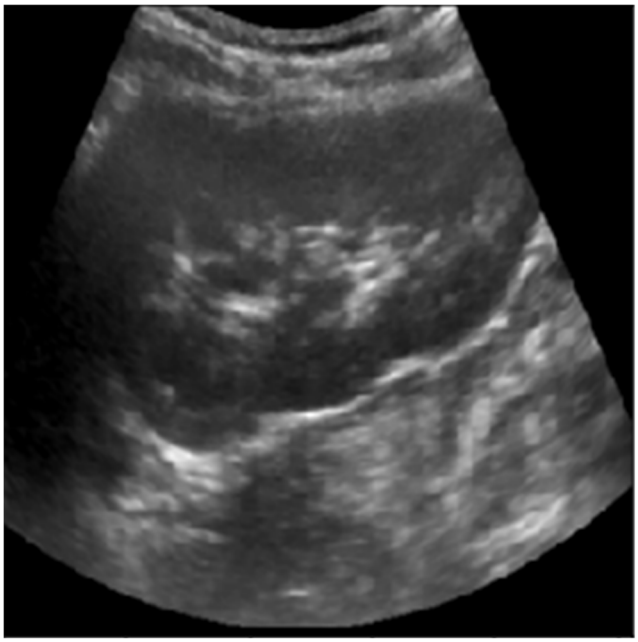

Figure 22. An US image that was classified as SFU grade 0 by our model, which was the correct grade, however, the supplied grade label was SFU 2.

Individual inspections of each image would be required to determine what percentage of our dataset is affected by mislabelling. Nevertheless, this finding has strong implications for our best model. Incorrectly labelled images impact our accuracy calculations as we found, however, they also greatly impact model training. During training weights are updated as the model attempts to minimize a chosen cost function. The weights are updated based on model error, therefore, when mislabelled data passes through the network it can negatively influence how the weights are updated since the errors are not accurate in all instances. Therefore, it's very likely that our model didn't properly converge and find the correct weights to minimize the cost function.

The only way to solve this problem is to go through and fix all mislabelled images. It's likely that many of the mislabelled images will have predicted scores that are greater than two values away from their labelled values, therefore we can use Model 5 to verify these images by inspecting its predictions. Targeting this smaller subset of images and updating their respective labels will be most effective since this subset likely contains the most errors. However, it cannot be ruled out that there are no mislabelled images that fall within one value of the correct score, therefore eventually these images will also need to be inspected. Although we are unsure how pervasive this issue is, we believe that once the image labels have been updated we will see a significant increase in our model's performance.

Although this small subset of badly misclassified images is concerning, most of the time our model either correctly classified or misclassified our US images by 1 SFU grade. This finding makes sense since considering that SFU grades are a discrete

approximation of severity, a continuous phenomenon. Furthermore, physicians do tend to confuse adjacent SFU grades, especially the intermediate ones. Therefore, although we hypothesize that cleaning this small subset of images will improve performance, this is not to say that our best model performs poorly. Our model has learned useful characteristics and is performing similarly to a physician.

Based on our findings, it would be interesting to examine the 'confidence' of our models output. Our model outputs a set of probabilities for each SFU grade, and then classifies the images as the grade with the highest probability. It is likely that in many instances two adjacent SFU grades have similar probabilities, meaning that the model thinks the image falls between two different SFU grades. However, like physicians, the model must choose one grade to classify the image, even if the probabilities are very close (e.g. 45% and 42%) and the image appears to lie somewhere in the middle. Examining the individual probabilities for each SFU grade would provide a way of placing each image along a continuous scale and may provide insight into how 'confident' the model is in each of it's classifications, and what sub features place an image on the upper or lower spectrum of each SFU grade.

Limitations and Future Directions

The findings from the current study are limited in that we did not conduct an exhaustive search for the best CNN architecture. We made a variety of incremental architectural changes to our models, however, there are many combinations that were not attempted due to the feasibility of a manual search. We believe that our exploration provides a strong starting point for further investigation. Now that there is a general

understanding of how architectural/hyper-parameter changes impacted model performance (e.g. decreasing depth, the number of filters in each layer etc.), an exhaustive search, where all combinations of architectural/hyper-parameter changes that were made in the current study are tested, should be performed. We believe that grid search, which is an exhaustive search through a subset of hyperparameter space, could be conducted. Grid search has been criticized by Bergstra and Bengio (2012) because often times feature space is so large that it can be computationally expensive to conduct. However, we believe that since an initial educated search has been conducted, the feature space has shrunk to a reasonable size to conduct an exhaustive grid search. There are infinite different CNN architectures, however, basing our grid search off the results of the current study should result in an optimal CNN architecture with improved accuracy for classifying HN US images.

Another major limitation of the current study was the discovery that our dataset contains mislabelled images. The mislabelling no doubt impacted training, and thus overall model performance. Although our current best model is not at an acceptable level for clinical use, it can be used to verify our image labels. Optimizing our current best architecture along with verifying our image labels should result in significantly higher HN classification accuracy.

Chapter 6: Conclusion

Currently, HN diagnosis is highly subjective and unreliable. Considering how important accurate diagnosis is for patient care and overall well-being, this problem must be addressed. Machine learning based diagnostic aids are becoming increasingly popular for medical image-based diagnosis, and their usage as a second opinion has been shown to improve physicians' diagnostic accuracy. We believe that HN can benefit from such a diagnostic aid as it will provide consistent and objective feedback to physicians for diagnostic consideration.

HN is diagnosed using US images. Therefore, we hypothesize that developing a CNN based diagnostic aid will produce the best results, since CNN are currently the leading model type for image recognition and classification tasks. To our knowledge, developing a CNN based diagnostic aid that can be applied to US images has not been done before, and therefore the current thesis conducted two exploratory studies to investigate two important methodological considerations, namely US image preprocessing and model architecture. Most recommendations for medical image preprocessing for developing diagnostic aids are geared towards classic ML techniques and not neural networks. Therefore, we investigated whether two common recommendations, image segmentation and textural extraction, are beneficial and improve performance when they are applied to CNN input images. Our results showed that image segmentation and textural extraction did not improve model performance, and therefore might not be required when using CNN medical image classification. The results of this study also suggested that background features from the US image that are not associated

with the kidney itself might be useful for diagnosis. Further studies should be conducted to assess whether our model utilized background information, and whether physicians utilize cues from the background of the US image. If physicians do regularly use background information, it would help to explain why inter-rater reliability is so poor. Furthermore, if CNN find relevant information from the background of HN US images, it suggests that these diagnostic features should be standardized for physicians to use.

The second study investigated CNN architectures. There are infinite combinations of CNN architectures/hyperparameters, therefore the goal was to investigate how various changes impacted performance in an attempt to find an optimal model for HN classification. Our search resulted in a best model with 49% 5-way classification accuracy. Physician accuracy for 5-way HN classification is unknown, however, we believe that our model performs well considering the low-inter rater reliability of the SFU classification system and the fact that 92% of images were either correctly classified or within one grade of the correct diagnosis. Interestingly, while investigating our models we found that our database contained mislabelled images, which is not surprising considering the data was entered into our database manually. Our inspection revealed that many of the images that were classified greater than two grades away from their "correct" SFU grade were mislabelled.

This finding has important implications. Training our models with mislabelled images negatively affects model training, and results in lower accuracy scores during testing. However, since many of the mislabelled images tend to be badly misclassified, we can utilize our current model to verify our images by inspecting the images that are

classified greater than two grades away from their provided SFU label. Therefore, although our current best model is not ready for clinical use as diagnostic aid, it can be used as an aid for data curation.

Overall, the current studies have provided insight into important methodological considerations for developing a diagnostic aid for HN. Although the current model is not yet at an appropriate level for clinical use, it can be applied to verify the accuracy of our database. Once our images and their respective labels have been verified, we can further optimize our model architecture by conducting an exhaustive hyperparameter search. We hypothesize that the combination of these two changes will significantly improve model performance and bring our diagnostic aid closer to clinical application.

References

Akram, M. U., Tariq, A., Anjum, M. A., & Javed, M. Y. (2012). Automated detection of exudates in colored retinal images for diagnosis of diabetic retinopathy. *Applied Optics*, *51*(20), 4858. https://doi.org/10.1364/AO.51.004858

Baaziz, N., Abahmane, O., & Missaoui, R. (2010). Texture feature extraction in the spatial-frequency domain for content-based image retrieval. *ArXiv:1012.5208 [Cs]*. Retrieved from http://arxiv.org/abs/1012.5208

Bayes, M., & Price, M. (1763). An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, *53*(0), 370–418. https://doi.org/10.1098/rstl.1763.0053

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Chan, H. P., Doi, K., Vyborny, C. J., Schmidt, R. A., Metz, C. E., Lam, K. L., … MacMahon, H. (1990). Improvement in radiologists' detection of clustered microcalcifications on mammograms. The potential of computer-aided diagnosis. *Investigative Radiology*, *25*(10), 1102–1110.

Cicero, M., Bilbily, A., Colak, E., Dowdell, T., Gray, B., Perampaladas, K., & Barfett, J. (2017). Training and Validating a Deep Convolutional Neural Network for Computer-Aided Detection and Classification of Abnormalities on Frontal Chest Radiographs. *Investigative Radiology*, *52*(5), 281–287. https://doi.org/10.1097/RLI.0000000000000341

Cireşan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column Deep Neural Networks for Image Classification. *ArXiv:1202.2745 [Cs]*. Retrieved from http://arxiv.org/abs/1202.2745

Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, *61*(1), 38–59. https://doi.org/10.1006/cviu.1995.1004

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8609–8613). https://doi.org/10.1109/ICASSP.2013.6639346

Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM.

Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233–240). New York, NY, USA: ACM. https://doi.org/10.1145/1143844.1143874

Dhindsa, K., Smail, L. C., McGrath, M., Braga, L. H., Becker, S., & Sonnadara, R. R. (2018). Grading prenatal hydronephrosis from ultrasound imaging using deep convolutional neural networks. In *15th Conference on Computer and Robot Vision*

(pp. 80–87). Toronto, Canada: IEEExplore. Retrieved from https://bibbase.org/network/publication/dhindsa-smail-mcgrath-braga-becker-sonnadara-gradingprenatalhydronephrosisfromultrasoundimagingusingdeepconvolutionalneuralnetworks-2018

Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, *31*(4), 198–211. https://doi.org/10.1016/j.compmedimag.2007.02.002

Dolata, P., Mrzygłód, M., & Reiner, J. (2017). Double-stream Convolutional Neural Networks for Machine Vision Inspection of Natural Products. *Applied Artificial Intelligence*, *31*(7–8), 643–659. https://doi.org/10.1080/08839514.2018.1428491

Eckroth, J. (2017). Deep learning | CSCI 431. Retrieved July 17, 2018, from http://csci431.artifice.cc/notes/deep-learning.html

Edwards, G. J., Cootes, T. F., & Taylor, C. J. (1998). Face recognition using active appearance models. In *Computer Vision — ECCV'98* (pp. 581–595). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0054766

Eva, K. W., Norman, G. R., Neville, A. J., Wood, T. J., & Brooks, L. R. (2002). Expert-Novice Differences in Memory: A Reformulation. *Teaching and Learning in Medicine*, *14*(4), 257–263. https://doi.org/10.1207/S15328015TLM1404_10

Fernbach, S. K., Maizels, M., & Conway, J. J. (1993). Ultrasound grading of hydronephrosis: Introduction to the system used by the society for fetal urology. *Pediatric Radiology*, *23*(6), 478–480. https://doi.org/10.1007/BF02012459

Giger, M. L. (2018). Machine Learning in Medical Imaging. *Journal of the American College of Radiology*, *15*(3), 512–520. https://doi.org/10.1016/j.jacr.2017.12.028

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (pp. 315–323). Retrieved from http://proceedings.mlr.press/v15/glorot11a.html

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Retrieved from http://www.deeplearningbook.org

Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., … Zalaudek, I. (2018). Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*. https://doi.org/10.1093/annonc/mdy166

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., … Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv:1207.0580 [Cs]*. Retrieved from http://arxiv.org/abs/1207.0580

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition* (Vol. 1, pp. 278–282 vol.1). https://doi.org/10.1109/ICDAR.1995.598994

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, *195*(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455

Huynh, B. Q., Li, H., & Giger, M. L. (2016). Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *Journal of Medical Imaging*, *3*(3), 034501. https://doi.org/10.1117/1.JMI.3.3.034501

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv:1502.03167 [Cs]*. Retrieved from http://arxiv.org/abs/1502.03167

Irem Turkmen, H., Elif Karsligil, M., & Kocak, I. (2015). Classification of laryngeal disorders based on shape and vascular defects of vocal folds. *Computers in Biology and Medicine*, *62*, 76–85. https://doi.org/10.1016/j.compbiomed.2015.02.001

Jalalian, A., Mashohor, S. B. T., Mahmud, H. R., Saripan, M. I. B., Ramli, A. R. B., & Karasfi, B. (2013). Computer-aided detection/diagnosis of breast cancer in

mammography and ultrasound: a review. *Clinical Imaging*, *37*(3), 420–426. https://doi.org/10.1016/j.clinimag.2012.09.024

Keays, M. A., Guerra, L. A., Mihill, J., Raju, G., Al-Asheeri, N., Geier, P., … Leonard, M. P. (2008). Reliability Assessment of Society for Fetal Urology Ultrasound Grading System for Hydronephrosis. *The Journal of Urology*, *180*(4), 1680–1683. https://doi.org/10.1016/j.juro.2008.03.107

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *ArXiv:1412.6980 [Cs]*. Retrieved from http://arxiv.org/abs/1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Laplace, P. S. de. (1814). *Théorie analytique des probabilités*. Courcier.

Le Cun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Le Cun, Yann. (1986). Learning Process in an Asymmetric Threshold Network. In *Disordered Systems and Biological Organization* (pp. 233–240). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-82657-3_24

Le Cun, Yann, Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Legendre, A. M. (1805). Legendre et la méthode des moindres carrés. Retrieved July 5, 2018, from http://www.bibnum.education.fr/mathematiques/algebre/legendre-et-la-methode-des-moindres-carres

Li, F., Aoyama, M., Shiraishi, J., Abe, H., Li, Q., Suzuki, K., … Doi, K. (2004). Radiologists' performance for differentiating benign from malignant lung nodules on high-resolution CT using computer-estimated likelihood of malignancy. *AJR. American Journal of Roentgenology*, *183*(5), 1209–1215. https://doi.org/10.2214/ajr.183.5.1831209

Lin, M., Chen, Q., & Yan, S. (2013). Network In Network. *ArXiv:1312.4400 [Cs]*. Retrieved from http://arxiv.org/abs/1312.4400

Livens, S., Scheunders, P., Wouwer, G. van de, & Dyck, D. V. (1997). Wavelets for texture analysis, an overview, 581–585. https://doi.org/10.1049/cp:19970958

Loizou, C. P., Theofanous, C., Pantziaris, M., & Kasparis, T. (2014). Despeckle filtering software toolbox for ultrasound imaging of the common carotid artery. *Computer Methods and Programs in Biomedicine*, *114*(1), 109–124. https://doi.org/10.1016/j.cmpb.2014.01.018

Mar, V. J., & Soyer, H. P. (2018). Editorial. *Annals of Oncology*. https://doi.org/10.1093/annonc/mdy193

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (1983). *Machine Learning: An Artificial Intelligence Approach*. Berlin Heidelberg: Springer-Verlag. Retrieved from //www.springer.com/gp/book/9783662124079

Miller, J. W. (2018). Face detection with Active Shape Models (ASMs). File Exchange. Retrieved December 16, 2017, from https://www.mathworks.com/matlabcentral/fileexchange/62766-face-detection-with-active-shape-models-asms

Minsky, M. (1954). *Neural nets and the brain-model problem*. Retrieved from Unpublishes doctoral dissertation, Princeton University, NJ

Minsky, M., & Papert, S. (1969). *Perceptrons: An essay in computational geometry*. Cambridge MA: The MIT Press.

Mistry, D. (2013). DISCRETE WAVELET TRANSFORM USING MATLAB. *International Journal of Computer Engineering & Technology (IJCET)*, *4*, 252–259.

Mudali, D., Teune, L. K., Renken, R. J., Leenders, K. L., & Roerdink, J. B. T. M. (2015). Classification of Parkinsonian Syndromes from FDG-PET Brain Data Using Decision Trees with SSM/PCA Features [Research article]. https://doi.org/10.1155/2015/136921

Nagi, J., Ducatelle, F., Caro, G. A. D., Cireşan, D., Meier, U., Giusti, A., … Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE International Conference on Signal and Image*

Processing Applications (ICSIPA) (pp. 342–347). https://doi.org/10.1109/ICSIPA.2011.6144164

Natarajan, B., Casida, M. E., Genovese, L., & Deutsch, T. (2011). Wavelets for Density-Functional Theory and Post-Density-Functional-Theory Calculations. *ArXiv:1110.4853 [Cond-Mat, Physics:Physics]*. Retrieved from http://arxiv.org/abs/1110.4853

Ng, A. (2016). IBM's Watson gives proper diagnosis after doctors were stumped. Retrieved January 19, 2018, from http://www.nydailynews.com/news/world/ibm-watson-proper-diagnosis-doctors-stumped-article-1.2741857

Nguyen, H. T., Benson, C. B., Bromley, B., Campbell, J. B., Chow, J., Coleman, B., … Stein, D. R. (2014). Multidisciplinary consensus on the classification of prenatal and postnatal urinary tract dilation (UTD classification system). *Journal of Pediatric Urology*, *10*(6), 982–998. https://doi.org/10.1016/j.jpurol.2014.10.002

Nguyen, H. T., Herndon, C. D. A., Cooper, C., Gatti, J., Kirsch, A., Kokorowski, P., … Campbell, J. B. (2010). The Society for Fetal Urology consensus statement on the evaluation and management of antenatal hydronephrosis. *Journal of Pediatric Urology*, *6*(3), 212–231. https://doi.org/10.1016/j.jpurol.2010.02.205

Nieminen, A., Heinonen, P., & Neuvo, Y. (1987). A New Class of Detail-Preserving Filters for Image Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(1), 74–90. https://doi.org/10.1109/TPAMI.1987.4767873

Olazaran, M. (1996). A Sociological Study of the Official History of the Perceptrons Controversy ,

A Sociological Study of the Official History of the Perceptrons Controversy. *Social Studies of Science*, *26*(3), 611–659. https://doi.org/10.1177/030631296026003005

Rickard, M., Easterbrook, B., Kim, S., Farrokhyar, F., Stein, N., Arora, S., … Braga, L. H. (2017). Six of one, half a dozen of the other: A measure of multidisciplinary inter/intra-rater reliability of the society for fetal urology and urinary tract dilation grading systems for hydronephrosis. *Journal of Pediatric Urology*, *13*(1), 80.e1-80.e5. https://doi.org/10.1016/j.jpurol.2016.09.005

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv:1609.04747 [Cs]*. Retrieved from http://arxiv.org/abs/1609.04747

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Sainath, T. N., Mohamed, A. r, Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8614–8618). https://doi.org/10.1109/ICASSP.2013.6639347

Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices in convolutional neural networks applied to visual document analysis, *3*, 958–962.

Song, Q., Zhao, L., Luo, X., & Dou, X. (2017). Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images [Research article]. https://doi.org/10.1155/2017/8314740

Sontag, E., & Hector, S. (1989). Backpropagation Can Give Rise to Spurious Local Minima Even for Networks without Hidden Layers, *3*, 91–106.

Spiegel, M., Hahn, D. A., Daum, V., Wasza, J., & Hornegger, J. (2009). Segmentation of kidneys using a new active shape model generation technique based on non-rigid image registration. *Computerized Medical Imaging and Graphics*, *33*(1), 29–39. https://doi.org/10.1016/j.compmedimag.2008.10.002

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, *15*(1), 1929–1958.

Sun, S., Bauer, C., & Beichel, R. (2012). Automated 3-D Segmentation of Lungs With Lung Cancer in CT Data Using a Novel Robust Active Shape Model Approach. *IEEE Transactions on Medical Imaging*, *31*(2), 449–460. https://doi.org/10.1109/TMI.2011.2171357

Swenson, D. W., Darge, K., Ziniel, S. I., & Chow, J. S. (2015). Characterizing upper urinary tract dilation on ultrasound: a survey of North American pediatric radiologists' practices. *Pediatric Radiology*, *45*(5), 686–694. https://doi.org/10.1007/s00247-014-3221-8

Tesi, A., & Gori, M. (1992). On the Problem of Local Minima in Backpropagation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *14*(1), 76–86. https://doi.org/10.1109/34.107014

Thanoon, F. H. (2015). Robust Regression by Least Absolute Deviations Method. *International Journal of Statistics and Applications*, *5*(3), 109–112.

Wan, K.-W., Lam, K.-M., & Ng, K.-C. (2005). An accurate active shape model for facial feature extraction. *Pattern Recognition Letters*, *26*(15), 2409–2423. https://doi.org/10.1016/j.patrec.2005.04.015

Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016). Deep Learning for Identifying Metastatic Breast Cancer. *ArXiv:1606.05718 [Cs, q-Bio]*. Retrieved from http://arxiv.org/abs/1606.05718

Wang, W., Shan, S., Gao, W., Cao, B., & Yin, B. (2002). An Improved Active Shape Model for Face Alignment. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces* (pp. 523–). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/ICMI.2002.1167050

Woodward, M., & Frank, D. (2002). Postnatal management of antenatal hydronephrosis. *BJU International*, *89*(2), 149–156. https://doi.org/10.1046/j.1464-4096.2001.woodward.2578.x

Zanetta, V. C., Rosman, B. M., Bromley, B., Shipp, T. D., Chow, J. S., Campbell, J. B., … Nguyen, H. T. (2012). Variations in management of mild prenatal hydronephrosis among maternal-fetal medicine obstetricians, and pediatric

urologists and radiologists. *The Journal of Urology*, *188*(5), 1935–1939. https://doi.org/10.1016/j.juro.2012.07.011

Appendix A

Backpropagation

Backpropagation is a method to calculate the gradients that are needed to update all of the weights, include those that are hidden, in a neural network. Since gradient descent relies on backpropagation, the cost function must be continuous and differentiable. Therefore, a step function is not appropriate and other differentiable activation functions, such as sigmoid or softmax, should be used depending on whether the task is binary or multi-classification respectively. Using continuous functions allows for the approximation of changes in the cost function.

Using the backpropagation algorithm, the training of neural networks is done as follows:

1. After the forward propagation, calculate the error signal of the final output layer at all neurons by calculating the gradient of the cost function with respect to each output where z_j^L is the output of the final layer (L) at neuron j, σ is the activation function, a_j^L is the activation of the final layer at neuron j, and C is the cost function (which can vary), and $\odot$ indicates component-wise multiplication:

$$e_j^L = \frac{\delta C}{\delta a_j^L}\sigma'\left(z_j^L\right) \quad (13)$$

2. Calculate the error of each neuron in each layer using backpropagation where l represents an earlier layer in the network:

$$e_j^l = \left(w_j^{l+1}\right)^T e_j^{l+1} \odot \sigma'\left(z_j^l\right) \quad (14)$$

3. Calculate the derivative of the cost function with respect to the weights and biases:

$$\frac{\delta C}{\delta w_{jk}^{l}} = a_k^{l-1} e_j^l \quad (15) \qquad \frac{\delta C}{\delta b_j^{l}} = e_j^l \quad (16)$$

4. Update the weights and biases according to the delta rule (Equations 17 and 18). λ represents the learning rate, or more informally, the size of the weight and bias change that will be made after each forward pass through the network.

$$\Delta w_l = -\lambda \, \Sigma_j \, e_j^l \left(a_j^{l-1}\right)^T \quad (17) \qquad \Delta b_l = -\lambda \, \Sigma_j \, e_j^l \quad (18)$$

Appendix B

Active Shape Model Algorithm

1. Using a subset of renal US images, save the X and Y coordinates of representative points along the edges of each of the kidneys. This will result in a $2n\ x\ 1$ vector (z_j) for each US where each US has the same number of points (n) organized in the same order:

$$z^j = \{(x_1^j, y_1^j), (x_2^j, y_2^j), (x_3^j, y_3^j), \dots, (x_n^j, y_n^j)\} \quad (19)$$

2. All shapes will be translated and centered at (0,0).
3. Fix one shape (e.g. z^1), and scale so that $|z^1| = 1$.
4. Scale and rotate all other shapes to align with this shape using Procrustes Analysis where $\hat{x}_i^j$ and $\hat{y}_i^j$ represent the new point locations for a given kidney, and s^j represent the sets of points

$$a^j = \frac{z^j \cdot z^1}{\|z^j\|^2} \quad (20)$$

$$b^j = \sum_{i=1}^{n} \left(x_i^j y_i^1 - x_i^1 y_i^j\right) / \left\|z^j\right\|^2 \quad (21)$$

$$s^j = \sqrt{(a^j)^2 + (b^j)^2} \quad (22)$$

$$\theta^j = tan^{-1}\left(\frac{b^j}{a^j}\right) \quad (23)$$

$$\begin{bmatrix} \hat{x}_i^j \\ \hat{y}_i^j \end{bmatrix} = s^j \begin{bmatrix} cos\theta^j & sin\theta^j \\ -sin\theta^j & cos\theta^j \end{bmatrix} \begin{bmatrix} x_i^j \\ y_i^j \end{bmatrix} \quad (24)$$

5. Principal Components Analysis (PCA) is used to reduce the dimensionality of the data.
    a. Compute the mean of the data:

$$\mu = \frac{1}{s} \sum_{i=1}^{s} z^i \quad (25)$$

b. Compute the covariance of the data:

$$\Sigma = \frac{1}{s-1}\sum_{i=1}^{s}\left(z^i - \mu\right)\left(z^i - \mu\right)^T \quad (26)$$

c. Compute the eigenvalues and eigenvectors of Σ:

$$\left(\lambda_j, v_j\right) such\ that\ \lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots\ \Sigma v_j = \lambda_j v_j \quad (27)$$

6. Each eigenvalue λ_j gives variance of data in the direction v_j. Compute total variance

$$T = \sum_{i=1}^{s}\lambda_j \quad (28)$$

7. Choose K largest eigenvalues to account for a desired proportion of variance.
8. We can now approximate any of the kidneys as where P is a matrix of the eigenvalues from the covariance matrix, and b defines a small number of parameters for the active shape model:

$$z = \mu + Pb \quad (29)$$

9. To fit this model to new data:
   a. Initialize $b = 0$
   b. Generate model points: $x = \mu + Pb$
   c. Search around each x_i for best nearby image point y_i using gradients to find edges.
   d. Fit new parameters (s, θ, t, b) to y using equations from step 4.
   e. Enforce constraint that $\|b_i\| < 3\lambda_i$ to ensure that shapes are reasonable.
   f. Update model parameters: $b = P^T(y - \mu)$
   g. Iterate until convergence.