Sjögren's Syndrome Detection With Machine Learning

Section 304

Final Deliverables

12/13/2023

Team:

Richard Yang (tyang296@wisc.edu)	Team Lead
Yousef Gadalla (ygadalla@wisc.edu)	Communicator
Brandon Drew (bsdrew2@wisc.edu)	BSAC
Dhruv Nadkarni (dnadkarni@wisc.edu)	BWIG
Siya Mahajan (mahajan24@wisc.edu)	BWIG
Aran Viswanath (viswanath3@wisc.edu)	BPAG

Abstract

Sjögren’s syndrome (SjS), a systemic autoimmune disease (SAD), manifests with exocrine gland dysfunction, particularly in the salivary and lacrimal glands, resulting in persistent dryness of the mouth and eyes [1, 2]. The current standard of care involves baseline salivary gland ultrasounds (of the submandibular and parotid glands) for potential SjS patients, with higher-risk individuals undergoing regularly scheduled ultrasounds. However, the Outcome Measures in Rheumatoid Arthritis Clinical Trials (OMERACT) ultrasound grading system introduces subjectivity and lacks nuance. To address this, a machine learning approach is proposed to reduce inter-reader variability and enhance diagnostic precision by detecting SjS directly from ultrasound images. The team employs a K Nearest Neighbour (KNN) model and a simple Convolutional Neural Network (CNN) model for initial assessment and a VGG-19-based model for final optimization. Results are summarized through accuracy metrics and confusion matrices, with the final model's performance evaluated relative to the baseline model. Concurrently, the team developed a separate model to score ultrasound with the OMERACT scale automatically. The SjS detection model exhibited 90% and 93% accuracy in positive and negative subjects, respectively. While the OMERACT scoring model achieved only 66.1% accuracy, it can be attributed to inherent inaccuracy stemming from inter-reader variability. Both models demonstrate practical speed and compactness suitable for disk storage.

Table of Contents

Introduction

Sjögren's syndrome (SjS) is a condition estimated to affect a significant population, ranging between 1 and 4 million individuals in the United States [3]. SjS is an autoimmune disease known for causing dryness of the eyes and mouth and can result in the immune system attacking other organs/tissues [4]. This can lead SjS patients to have an increased risk of lymphoproliferative diseases in which lymphocytes are uncontrollably produced [4]. Typically, patients are diagnosed after the age of 50, with increased prevalence among women [5]. While SjS currently lacks a cure, treatment options are tailored to the affected areas. Obtaining a precise and swift diagnosis with minimal invasiveness is crucial. Such a diagnosis plays a vital role in ensuring timely and suitable medical intervention, thereby reducing the associated risks of trauma, infection and delaying recovery.

Several diagnostic methods are employed to detect SjS, including blood and urine tests, Schirmer tear tests, Sialography, and Lip Biopsies. While these methods exhibit efficacy in SjS detection, they each present distinct accuracy, speed, and invasiveness challenges.

The OMERACT(Outcome Measures in Rheumatoid Arthritis Clinical Trials) Ultrasound Scoring System is another method of diagnosing SjS and encompasses a set of guidelines for interpreting ultrasound images of the parotid and submandibular glands [6]. While this approach minimizes invasiveness, it relies on human interpretation, introducing subjectivity into the diagnostic process and increasing inter-reader variability.

A real-time machine learning model that can analyze ultrasound images to provide a positive or negative SjS diagnosis and a predicted OMERACT score is desirable. It removes human subjectivity and allows for practical clinical use and quicker patient treatment access.

Background

The team’s client, Dr. Sara McCoy, is a faculty member in the Division of Rheumatology within the Department of Medicine. She is a clinical rheumatologist, the UW Health Sjögren’s clinic director, and a core member of the University of Wisconsin Carbone Cancer Center [7].

SjS is an autoimmune disorder generally characterized by two main symptoms: dryness of the eyes and mouth. This results in the manifestation of its most common complications: dental cavities, yeast infections, and vision problems, and is often found in conjunction with other rheumatic diseases [8].

The OMERACT scoring system utilizes salivary gland ultrasounds to diagnose SjS and other rheumatic diseases. It is characterized by a four-grade scoring system based on patients' parotid and submandibular glands, starting at 0, normal appearance, and going to 3, maximum change from the normal. Despite OMERACT's popularity amongst physicians, the scoring system is still not present in the 2016 American College of Rheumatology/European League Against Rheumatism (ACR/EULAR) classification criteria, so it is often used as an initial step to determine if a patient is at risk for SjS and if other tests should be performed to determine if the patient does or does not have SjS [9].

There are other methods that, while capable of diagnosing SjS, introduce invasive complications and accuracy issues. Blood and urine tests, for instance, may be susceptible to sample contamination, which could lead to unnecessary hospitalization and patients' exposure to unwarranted medications, posing both safety and financial risks [10].

The Schirmer Tear Test, which entails the insertion of a filter paper strip into the patient's eyelid to measure tear travel distance, can cause discomfort and the potential for infection due to the foreign body insertion [11]. Sialography, another imaging method, should be used sparingly, given its requirement for patient sedation, contrast dye injection into the salivary glands, and radiation exposure through X-rays. This procedure carries risks of salivary duct damage, swelling, and tenderness [12].

Lastly, the inner lip biopsy, involving the extraction and analysis of lip tissue, represents a significantly time-consuming and invasive procedure compared to an ultrasound scan. It necessitates surgery for tissue removal, followed by the separation and transfer of glands to a pathologist with specialized training for SjS diagnosis [13]. Consequently, this process may delay treatment until confirmation from the pathologist is obtained.

Machine learning has been used with increasing frequency in the medical field, and an important type of machine learning is a convolutional neural network. Convolutional neural networks (CNNs) are a type of machine learning algorithm that excel in image analysis and have current applications in various medical image and data analysis, such as electrocardiography and computed tomography angiography [14]. The key feature of CNN is the convolutional layers, which isolate key features from input images. Given a set of pre-trained weights, K, and the input image X, the analysis through convolutional layers follows equation 1, which takes the pre-trained weights and analysis of the image to output a feature map Y [15].

$$Y[m,n] = \sum_{i=1}^{K}\sum_{j=1}^{K} K[i,j]X[m-1+i, n-1+j] \qquad \text{Equation 1}$$

Another aspect of CNNs is the use of pooling layers. These layers resize the feature maps created by convolutional layers to isolate features on another level[15]. These layers also reduce the size of the feature map to allow for subsequent analyses through a fully connected layer[15].

The fully connected layer is created by connections from the input image to various features on the feature map with varying weights. This is analyzed by the output layer, and a predicted classifier is given.

During the training process, backward propagation is used to adjust the training weights the model creates [15]. This is done by comparing the output layer to the known classification of the training image. The error is calculated via the loss function, which is propagated backward through the model to adjust weights to improve the algorithm's accuracy [15]. Loss functions are typically tailored to the given task of the model, and for multiclass single-label tasks (meaning that only one classification is outputted to the image out of several possible classifiers), a categorical cross-entropy function is used [15]. The equation is given as

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top,\ l_n = -w_{y_n} log \frac{exp(x_{n,y_n})}{\sum_{c=1}^{C} exp(x_{n,c})} \cdot 1\{y_n\}\ [16] \qquad \text{Equation 2}$$

x is the input, y is the target, w is the weight, C is the number of classes, and N spans the mini-batch dimensions [15]. An optimizer is used to minimize the loss function. A common optimization algorithm is the gradient descent method [17]. This method looks at the relationship between the error and weights of the algorithm to calculate the partial derivative of the error in terms of the weight; this equals the gradient given by equation 3.

$$Gradient = \frac{\Delta E}{\Delta w} \qquad \text{Equation 3}$$

where E is the error and w is the weight [18]. The model utilizes the gradient to change the weights in order to minimize the error. To update the weights, the algorithm does this in steps dx determined by the learning rate (equation 4), with α is the learning rate of the algorithm [18]. If dx is too large, the algorithm could overstep the optimal weights needed to reach the minima of the loss function [18].

$$dx = \alpha * \left|\frac{dError}{dWeight}\right| \qquad \text{Equation 4}$$

Another critical aspect of training CNNs is batch training and epochs. With a large sample size, backward propagation training would take a long time, so models utilize batch training to reduce the run time [15]. Batch training allows a specified number of training data, referred to as the batch number, to be run through the algorithm before backward propagation occurs. An epoch is the number of times the training data is input into the algorithm, and by increasing the number of epochs, the algorithm is able to improve its accuracy [15].

Preliminary Designs

Baseline Algorithms

To evaluate the effectiveness of the final algorithm design, the team opted to also write an algorithm for a simplified machine learning program that will act as a baseline. This baseline will be utilized for comparison and to ensure that the accuracy of the final algorithm is significantly better than that of the baseline. These algorithms will be examples of supervised learning, meaning that they will be provided data that is already classified and they will analyze these data points to create their generalizations/predictive capabilities for new data sets. Four algorithms

were considered for the baseline model: support vector machine (SVM), K-nearest neighbor (KNN), random forest, and import vector machine (IVM).

The support vector machine is an algorithm that performs binary classification. The algorithm takes in the data and, based on the varying characteristics being observed, creates a hyperplane that separates the categories [13]. This hyperplane is then used to predict the classification of new data points. A limitation of the SVM is its usage with non-linear datasets, which makes hyperplane creation more difficult and inaccurate [19].

The K-nearest neighbor algorithm (KNN) is an algorithm that places the training data upon a grid based on some empirical value. The distance between this new data point and existing data is calculated when new data is added. Then, based on the known classifications of the K nearest data points, this new data point is assigned the value of the average classification [20]. For example, given K=1, the closest existing data point will be used to predict the diagnosis of the new data point. With a K=3, the average diagnosis of the three closest neighbors would be used to diagnose the new data point.

The random forest algorithm utilizes a series of classifications/data comparisons with weights that will be used in the final calculation of the diagnosis [21]. The tree is typically created by utilizing the training data, and random attributes are taken from the data to create a tree diagram with nodes and leaves. Given the importance of a given characteristic, a weight will be assigned to the characteristic to reflect said importance. As new data is added, the data will be placed into the tree and compared to the existing branches [21]. In the end, the score given by the tree will provide the diagnosis. Creating trees can be repeated to form several coexisting trees to improve the tree's accuracy. These trees would then output some scores, and the average diagnosis will be used to characterize the new data [21].

The import vector machine (IVM) is an updated version of the SVM developed to improve the SVM's limitation in nonlinear/complex data classification [22, 23]. This algorithm's central idea is data transformation utilizing a kernel method. This allows the data to be manipulated to create a more linear relationship, creating a hyperplane, even when the original data are nonlinear.

Final Algorithms

Five supervised learning algorithms were considered for the final algorithm: ResNet-50, deep neural networks (DNN), convolutional neural networks (CNN), VGG-19, and U-net.

ResNet-50 is a machine-learning algorithm with 50 layers. The algorithm uses these layers to analyze data to characterize better and separate the data. This algorithm utilizes a method known as "skip connections" that allows it to ignore specific layers that have been found to damage the framework and accuracy of the model [24]. This allows the algorithm to learn from itself and continually improve with more data sets.

DNNs are an algorithm type characterized by how it can analyze data and isolate important predictive features for the data [25]. This is done by having many layers that analyze the data and perform small transformations to the inputted data, which is done to mimic the brain's way of processing information.

CNNs are a subset of DNN specifically designed for image processing [14]. CNNs have many different layer types to fulfill the algorithm's image classification task: the first layer is a convolutional layer, which extracts physical information from the images. This data is then put through the pooling layers, simplifying the images and holding the most essential data features [15]. Finally, this is analyzed by connected layers that analyze the simplified image and data features to draw predictions from the data.

VGG-19 is a type of CNN and DNN with 19 pre-existing layers that are pre-trained [26]. The large number of layers allows this algorithm to excel at information extraction from data, which helps improve its accuracy [27]. This algorithm is used in many research papers for image classification and analysis [26, 27].

UNet is a CNN variant algorithm typically applied in processing biomedical images. The necessity of classifying medical images requires UNet to be good at analyzing each pixel of a given image. This helps the algorithm to locate areas of interest within a given image [28]. Once the area of interest has been located, convolutional layers are applied to isolate important features that can be used later for image analysis [28].

Preliminary Design Evaluation

Criteria Descriptions

Accuracy is the percentage of correct classifications in relation to total predictions made. This is an essential part of the algorithm, as it has to minimize the percentage of errors. If the algorithm were not accurate, data would be incorrectly labeled, resulting in either a false positive or false negative diagnostic. As a result, patients would either receive unnecessary treatment or not get any treatment at all.

Processing speed is how quickly a computer can process data or instructions. For machine learning, it is how quickly a model processes data, interprets data based on previous data, and produces a predicted output. The processing speed interacts with building complexity and compactness, as complexity is defined by how many layers and filters a program has, which impacts the speed at which a computer can process data. Programs that do not process, interpret, and produce an output based on the provided ultrasound images quickly and efficiently have a lower score.

Building Complexity looks at how complex the algorithm is to code. How many layers the learning algorithm has for data analysis impacts this. Programs with more layers and filters for analyzing data have lower scores reflecting their complexity.

Compactness looks at the size of the algorithm. A smaller algorithm size is better as it takes less space on a hard drive.

Scalability is the improvement potential of the algorithm with increasing dataset, i.e., performance = O(n), where n is the size of the dataset. This encompasses how easily the algorithm's structure can be adjusted, the model can be re-trained, the weights can be reset, and

the hyperparameters can be flexible. A larger improvement potential is desirable so that the algorithm's accuracy can scale with more complete training data.

Baseline Design Matrix

	SVM	KNN	Random Forest	IVM	Weights
Accuracy/Safety	14	18	16	16	20
Processing Speed	10.5	10.5	9	13.5	15
Building Complexity	8	9	5	7	10
Compactness	7	8	6	8	10
Scalability	7.5	12	9	12	15
Sum	67.14%	82.14%	64.29%	80.71%	70

Table 1*: Baseline Model Design Matrix

* green highlights indicate the highest scores in the category

Baseline Design Scoring

Scores for each algorithm were determined based on published data from papers and studies conducted by institutional and research bodies.

The support vector machine scored relatively low in accuracy due to SVM's limitation in non-linear data analysis. As the classification of salivary ultrasounds will likely not fall into a neat linear pattern, the SVM received a score of 14/20 for accuracy/safety. The processing speed of SVMs follows a time complexity of $O(n^2)$ [29]. This means as the number of data points increases, the amount of time that the SVM runs will increase quadratically. This is seen in SVM's low processing speed score. The low scalability score of SVM is connected to how adding more non-linear data will cause the hyperplane to be more inaccurate, especially as the hyperplane does not typically change with new data points.

The KNN had processing speeds similar to SVM's due to its requirement to draw new connections between each data point and existing data. The accuracy is deemed higher due to research studies comparing KNN vs SVM algorithms and finding improved KNN accuracy within EEG data reading and other medical data analysis [30]. This is seen in KNN's score of 18/20 for accuracy. The scalability score of KNN is relatively high because of KNN's ability to adapt new data into its predictive algorithm for diagnostic purposes.

In a study comparing KNN and random forest, the accuracy of random forest was found to be less than KNN (96% vs 84% accuracy) [31]. This is reflected in the random forest's score of 16/20. Due to the need for random trait selection from the data set and important scoring of these traits, the building complexity of the random forest is relatively low compared to the other algorithms.

The IVM has a higher score than the SVM due to the IVM being made as an improvement of the SVM in many categories. The IVM has been altered to utilize a kernel method to improve its

accuracy for the complex data that will likely be present in the ultrasound analysis of salivary glands. Also, as the processing speed is meant to have been improved, the processing speed is higher, and thus the score is higher. The only category in which the IVM is worse is the complexity category because the changes to the SVM to improve it require more complex coding and data transformation.

Final Design Matrix

	ResNet-50	DNN	CNN	VGG-19	U-net	Weights
Accuracy/Safety	18	18	16	18	16	20
Processing Speed	7.5	9	10.5	10.5	10.5	15
Building Complexity	7	4	6	7	6	10
Compactness	6	6	7	7	7	10
Scalability	10.5	12	10.5	9	10.5	15
Sum	70.00%	70.00%	71.43%	73.57%	71.43%	70

Table 2*: Final Model Design Matrix

* green highlights indicate the highest scores in the category

Final Design Scorings

As many of the algorithms are variations of each other, there are many criteria where they have similar values. The ResNet-50, DNN, and VGG-19 were relatively similar in accuracy due to their ability to multiple-layer analysis of images. CNN and U-net were scored slightly lower due to reports of decreased accuracy with less resolved images [14, 32].

The building complexity of all the algorithms was relatively low; however, ResNet-50 and VGG-19 scored slightly higher due to the pre-training that they received, which could potentially alleviate the building complexity. This analysis can change down the line as this pre-training could potentially hinder the ultrasound analysis of the algorithms.

Proposed Final Design

Baseline Models

Based on the scoring of the Baseline Model Design Matrix, it was found that the KNN algorithm would perform the best out of the four baseline models. This model was found to be extremely accurate when classifying medical data. In addition, the KNN algorithm is highly adaptive to new datasets because of its predictive algorithm. This allows the algorithm to remain accurate, even when given a new dataset. Overall, the high accuracy and predictivity of the KNN model allowed it to outperform the other three models.Additionally, KNN allowed the users to change certain parameters. The key change was with the weights parameter, as a change to “distance”

allowed the algorithm to put higher weightage on data points closer in 3D space instead of every data point being weighted equally.

In addition, a simple CNN model was developed to provide more information about the baseline. The architecture of the CNN model is summarized in Table 3. A total of two convolution layers are present, with three layers of fully connected layers.

“Conv2d” is a standard two-dimensional convolution operation, “MaxPool2d” is a two-dimensional pooling that keeps the maximum parameter, and “Linear” is a fully connected layer that applies a linear transformation to the input.

	Layer	**Output Shape**	**Number of Parameters**
1	Conv2d	[-1, 6, 251, 251]	156
2	MaxPool2d	[-1, 6, 125, 125]	
3	Conv2d	[-1, 16, 121, 121]	2,416
4	MaxPool2d	[-1, 16, 60, 60]	
5	Linear	[-1, 128]	7,372,928
6	Linear	[-1, 84]	10,836
7	Linear	[-1, 4]*	340
	Total params	7,386,676	
	Params size (MB)	28.18	
	Estimated Total Size (MB)	33.1	

*OMERACT Scoring model has the output shape of [-1, 4] while the SjS detection model has the output shape of [-1, 2]

Table 3: Architecture of the simple CNN model

Final Model

Based on the Final Model Design Matrix scoring, it was predicted that the VGG-19 model would perform the best for this project. Since VGG-19 is a pre-existing algorithm with pre-trained layers, it was found that this model would be less complex to build. That being said, the large amount of layers in this model allowed it to be highly accurate, even though the algorithm would be less complex to build than others. The VGG-19 model also is mainly used for image classification, which is an essential part of this project. Overall, the high accuracy and low building complexity of VGG-19 are what ultimately set it apart from the other algorithms and made it an excellent choice for the classification of ultrasound images.

Development Process

Materials

PyTorchVGG

PyTorch is a deep-learning framework with an optimized tensor library. As a result, a library of methods can be used for the VGG-19 model. Since PyTorch also maintains these tensor libraries, no major performance limitations exist. If problems arise, PyTorch will quickly update and resolve them. Another advantage to using PyTorch is the fast operating speed. This framework dramatically shortens neural network design, training, and testing portions. Finally, this framework is designed for use in Python. This makes PyTorch much easier for people to learn since Python is one of the widest used coding languages. This also allows Python's tools, like its debugging tools, to be used in PyTorch. Overall, PyTorch eases the process of creating, training and testing the VGG-19 model.

SklearnKNN

SKlearn is a Python library providing numerous resources for the K-Nearest-Neighbors classifiers. The library itself provides a basic KNN machine learning model, which is then customized by users via parameters to create a more advanced KNN model. Essentially, SKlearn allows a user to create a machine learning model without the fear of unknown bugs and errors in the realm of machine learning, as the library corrects and warns of those issues. Additionally, SKlearn provides users with a framework to assess a model's overall accuracy and individual data point accuracy. Overall, SKlearn for KNN eases creating, training, and testing the KNN model.

Dataset

After preprocessing, the dataset contains 4242 images. The exact composition is summarized in Table 4 and visualized in Figures 1, 2.

OMERACT Score	Number of Samples	Disease Status*	Number of Samples
0	243	0	30
1	1331	1	4211
2	2447	Total	4241
3	221		
Total	4242		

* 1 indicates that the subject is positive for SjS, 0 indicates that the subject is negative for SjS

**one image was removed from the Disease Status samples due to its incompatibility with the KNN; reasons are unknown

Table 4: Data Summary

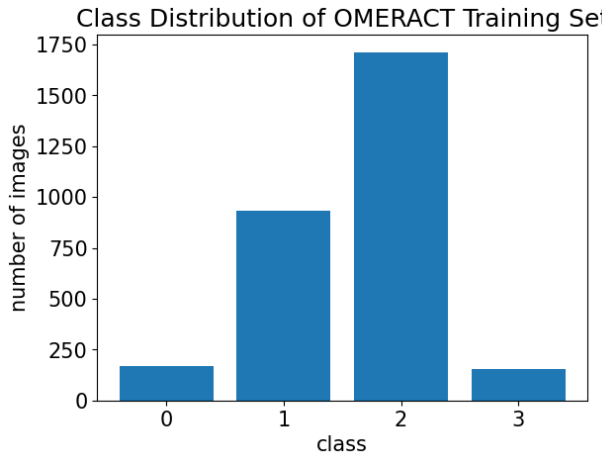


Figure 1: Distribution of OMERACT Training set

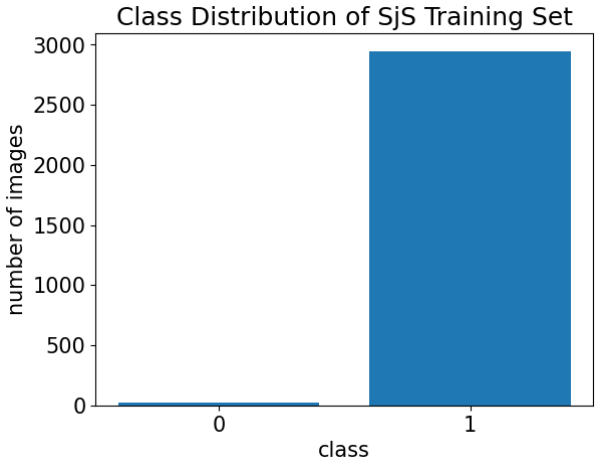


Figure 2: Distribution of SjS Training set

The substantial data imbalance between the number of SjS positive and negative training images presents a challenge for machine learning. In the case of the OMERACT model, it can be partially corrected by the Synthetic Minority Oversample Technique (SMOTE), downsampling the majority class, or pre-setting class weights for the loss function during model training. SMOTE up-samples for each minority class until all classes have a similar or equal number of samples. The team opted for pre-setting the class weights first, as this is less computationally expensive and preserves the most information. The team also implemented an experimental data augmentation technique.

To maximize the amount of information the model receives and mimic the ideal clinical usage of the algorithm, two ultrasound scans of the same patient, one of each gland, should be given to the model for analysis; however, while each SjS positive subject had multiple scans of different glands, the negative subject had only one scan each. Thus none of the negative subjects' scans can be paired into inputs for this algorithm, rendering this ideal testing impossible without more

data. Thus, the models are trained to detect SjS with only one ultrasound scan and disregard which gland it is from.

Image Preprocessing

Image Sorting

While the original images lacked organization based on glands, SjS presence, or OMERACT scores, many of them featured text overlay and associated data in CSV files. The team utilized Optical Character Recognition (OCR) along with the CSV data corresponding to the images, employing them to categorize the images into a system of nested folders, as illustrated in Figure 3. This organizational model provided the learning models with a well-structured dataset, categorizing images according to class labels.

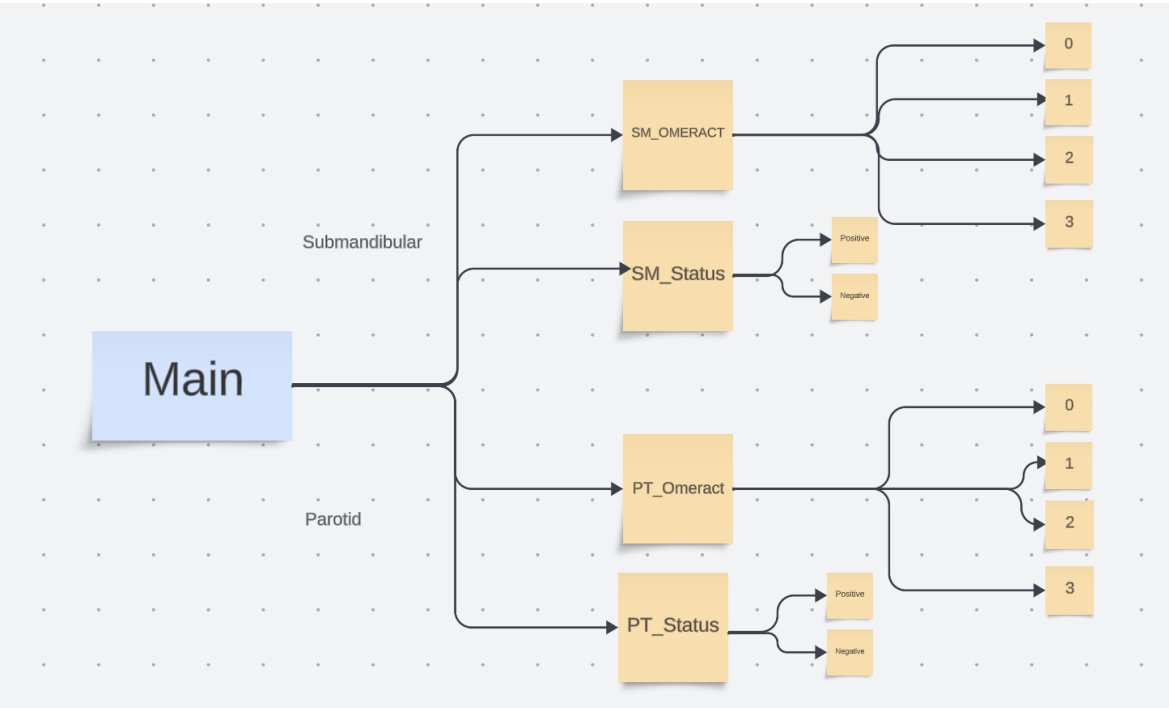


Figure 3: Data Divisions

The training images were sourced from two distinct datasets with varying formats. Dataset 1 comprised ultrasound images in DICOM format, containing patient data encoded into tags, one of which held the ultrasound image. This image featured an ultrasound depiction of the gland with overlaid text indicating the specific gland, along with extraneous information that required cropping. Additionally, this dataset included a complementary CSV file, supplying each patient's corresponding OMERACT score. The pixel data underwent conversion into JPEG format, followed by the application of an Optical Character Recognition (OCR) algorithm to extract the gland type. Subsequently, the images were cropped to the boundaries of the ultrasound depiction, and those with colored heat maps were excluded from the dataset

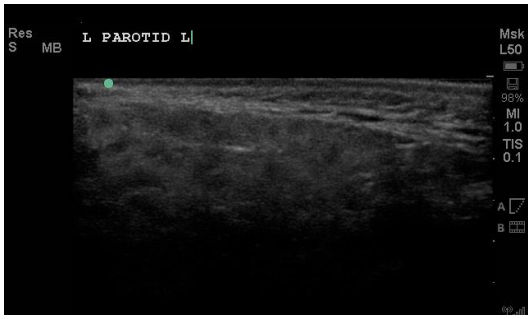


Figure 4-1: Original image from Dataset 1

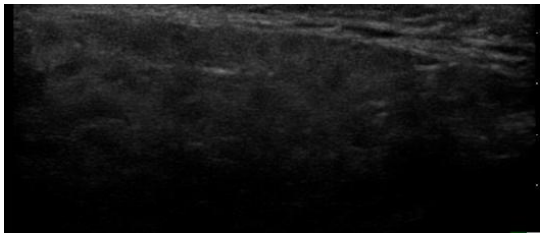


Figure 4-2: Processed image from Dataset 1

Dataset 2 comprised JPEG files containing ultrasound images, featuring a gland label overlaid onto each image. Optical Character Recognition (OCR) was applied to these images, with subsequent cropping to eliminate the overlaid text. A corresponding CSV file complemented this dataset, providing OMERACT scores and disease status data for each image.

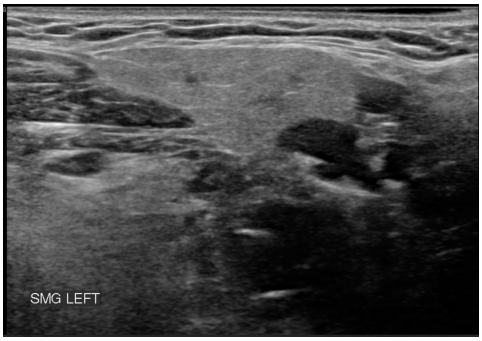


Figure 5-1: Original image from Dataset 2

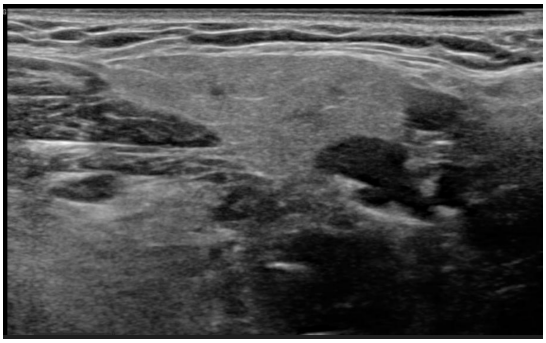


Figure 5-2: Processed image from Dataset 2

Data Augmentation

Due to a notable imbalance in class labels, data augmentation was implemented on the processed images within the minority classes of the training set to artificially equalize class data sizes. When employed properly, data augmentation can enhance training generalization, enabling the model to accommodate variations in data collection. The augmentation process involved random rotations within the range of -5º to 5º, horizontal flipping, and the application of a Gaussian filter for blurring.

It's important to note that this augmentation procedure has not undergone testing due to timing constraints. All tests conducted thus far have been performed using the non-augmented dataset.

Regularization

All the images must be of the exact dimensions for the model to process; therefore, regularization is needed to account for different image resolutions and encoding methods. Each

image is first resized for its shortest side to be 255 pixels long; then, it is center-cropped to a square of 255 by 255 pixels. It is converted to grayscale format, converted into tensors, and the pixel intensities are normalized to a mean of 0.5 and a standard deviation of 0.3.

The resizing parameters are chosen to maximize input resolution while not exceeding the 32 GB memory limitations. The normalization parameters are chosen to keep the all the images approximately identifiable and not over-exposed by visual inspection.

Model Training

As the KNN model does not require training, the following section is only relevant to the simple CNN and VGG-19 models.

Batch Size

Although some researchers have found that increasing the batch size has the same effect as decreasing the learning rate proportionally, others have discovered that batch size may also affect the ability of the model to generalize [33]. Defining small batch as 32-512 samples per batch, Keskar et al. state that large batch methods land in sharp minima, whereas small batch methods land in flat minima (Figure 6). Keskar found that, and also by conventional wisdom, the model performance increases with batch size to a certain threshold, after which the performance deteriorates [34].The exact size of the batch is dependent on the dataset.

In addition to the above considerations, the team also had hardware limitations. Increasing batch size results in a proportional increase in memory usage of the machine performing the training calculations. Through empirical testing, the simple CNN model can perform training with the entire training dataset in one batch, while the maximum batch size for the VGG-19 model is 512.

The team started each training with the maximum allowed batch size and decreased by a factor of 2 until the desired result was achieved.

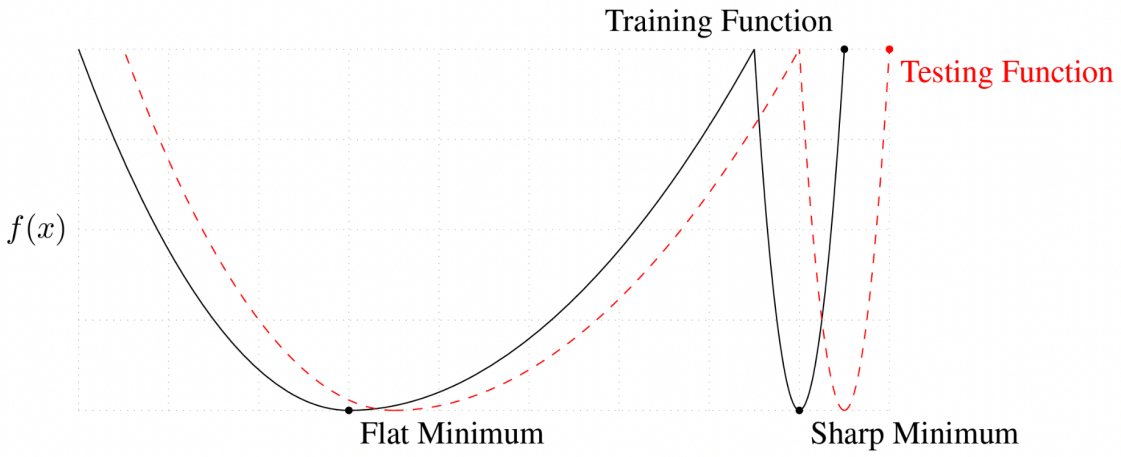


Figure 6: Flat Minimum and Sharp Minimum [34]

Learning rate

The learning rate, denoted by η, determines the step size taken into the gradient direction in backpropagation [15]. The ultimate goal of machine learning is to find the global minima of the loss function, and if the learning rate is too high, the model will step over the minima over and over again, resulting in oscillating behavior; if the learning rate is too low, the model progresses very slowly, resulting in excessive computations and may become stuck in a local minimum. As a result, the team employed a learning rate with the exponential decay value γ when the model requires a variable learning rate (Equation 5).

$$\eta = \eta_0 \times \gamma^t \quad [35] \qquad \text{Equation 5}$$

Loss Function

The team employed the cross-entropy loss function described by equation 6. The cross-entropy loss function allowed the team to specify the weight of each class when computing loss, which helped minimize the adverse effect of an unbalanced training sample. Class weights were computed using *sklearn.utils.class_weight.compute_class_weight,* which calculates the weight for each class using equation 6. C is the number of classes, n_{y_i} is the number of samples in class y_i, and w_{y_i} is the calculated weight of the class y_i.

$$\{w_{y_1},\ldots, w_{y_C}\} = \frac{total\ number\ of\ training\ samples}{C \times \{n_{y_1},\ldots,n_{y_C}\}} \quad [36] \qquad \text{Equation 6}$$

Optimizer

The ADAM optimizer was employed since it required less memory than its alternatives, and its hyperparameters typically required little tuning. Its implementation is well explained by Diederik P. Kingma and Jimmy Lei Ba [37].

Training Method

As there are no absolute rules in model training, the team implemented the following heuristic approach sequentially:

- Set the learning rate to the default 0.001
- Train for 20-70 epochs, dependent on time
- Print out loss/accuracy vs epochs graph

The graph is then interpreted, and adjustments were made accordingly and retrained.

Figure 7-1 shows the ideal training results, where loss plateaus after a certain amount of epochs. It informs the team to stop training before it plateaus to avoid overfitting.

Figure 7-2 shows that the model's learning rate is too high, it oscillates dramatically, and the loss never converges. The learning rate should, therefore, be reduced.

Figure 7-3 shows that although the model's loss is consistently decreasing, it never converges, which means that the learning rate is too low or it was not trained for a sufficient number of epochs.

Figure 7-4 shows that the learning rate is appropriate initially, but as training progresses, it becomes too large for the model. Thus, a variable learning rate is needed.

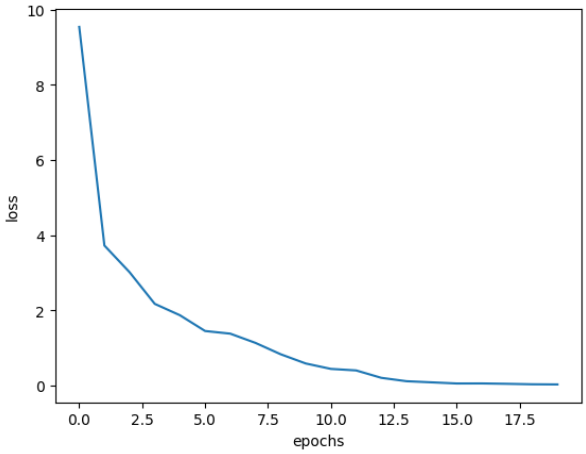


Figure 7-1: Ideal Training Progression

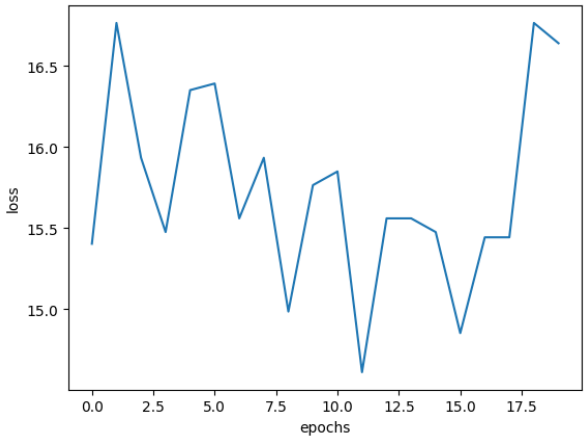


Figure 7-2: Learning Rate Too High

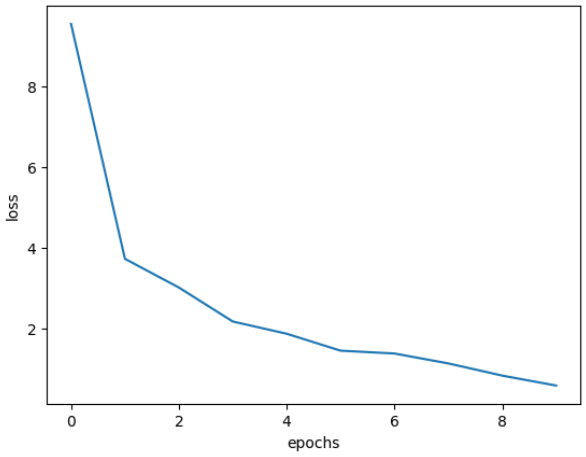


Figure 7-3: Learning Rate Too Low

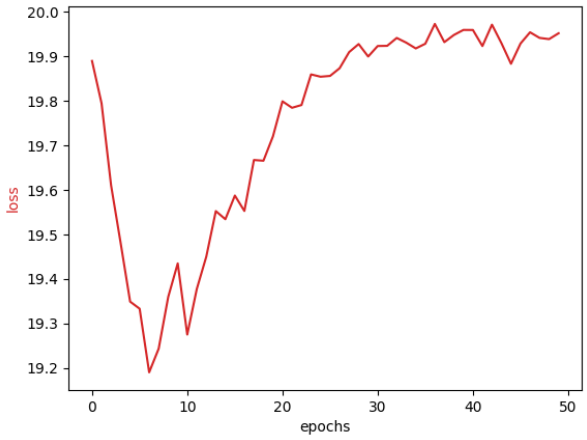


Figure 7-4: Variable Learning Rate Needed

Final Prototype

The architecture of the VGG-19 model is shown below in Table 5. It was modified to have an input depth of one instead of three since grayscale images only have a depth of one. The output layer is also modified from the original VGG-19 model to an output cardinality of 2 or 4 instead

of 1000 to accommodate the number of labels necessary for SjS detection and OMERACT scoring.

The SjS detection model is trained with η of 0.0001 batch size of 32 and γ of 0.9 for 50 epochs (Equation 5). The OMERACT scoring model is trained with η of 0.00001, batch size of 512 with no learning rate decay for 70 epochs, and then γ of 0.95 is applied for an additional 105 epochs.

Figure 8 visualizes the 17 convolution operations after training (the two MaxPool operations are not shown since no trainable parameters are associated with them). Each image shown is compressed into a single-layer image, which is represented by 64 to 512 layers in the model.

	Layer	Output Shape	Number of Parameters
1	Conv2d	[-1, 3, 253, 253]	30
2	Conv2d	[-1, 64, 253, 253]	1,792
3	ReLU	[-1, 64, 253, 253]	--
4	MaxPool2d	[-1, 64, 126, 126]	--
5	Conv2d	[-1, 128, 126, 126]	73,856
6	ReLU	[-1, 128, 126, 126]	--
7	MaxPool2d	[-1, 128, 63, 63]	--
8	Conv2d	[-1, 256, 63, 63]	295,168
9	ReLU	[-1, 256, 63, 63]	--
10	Conv2d	[-1, 256, 63, 63]	590,080
11	ReLU	[-1, 256, 63, 63]	--
12	MaxPool2d	[-1, 256, 31, 31]	--
13	Conv2d	[-1, 512, 31, 31]	1,180,160
14	ReLU	[-1, 512, 31, 31]	--
15	Conv2d	[-1, 512, 31, 31]	2,359,808
16	ReLU	[-1, 512, 31, 31]	--
17	MaxPool2d	[-1, 512, 15, 15]	--
18	Conv2d	[-1, 512, 15, 15]	2,359,808
19	ReLU	[-1, 512, 15, 15]	--
20	Conv2d	[-1, 512, 15, 15]	2,359,808
21	ReLU	[-1, 512, 15, 15]	--
22	MaxPool2d	[-1, 512, 7, 7]	--
23	AdaptiveAvgPool2d	[-1, 512, 7, 7]	--
24	Linear	[-1, 2048]	51,382,272
25	ReLU	[-1, 2048]	--
26	Linear	[-1, 512]	1,049,088
27	ReLU	[-1, 512]	--
28	Dropout	[-1, 512]	--
29	Linear	[-1, 4]*	2,052
30	Softmax	[-1, 4]*	--
	Total params	61,653,922	
	Params size (MB)	235.19	
	Estimated Total Size (MB)	308.45	

*OMERACT Scoring model has the output shape of [-1, 4] while the SjS detection model has the output shape of [-1, 2]

Table 5: Architecture of the VGG model

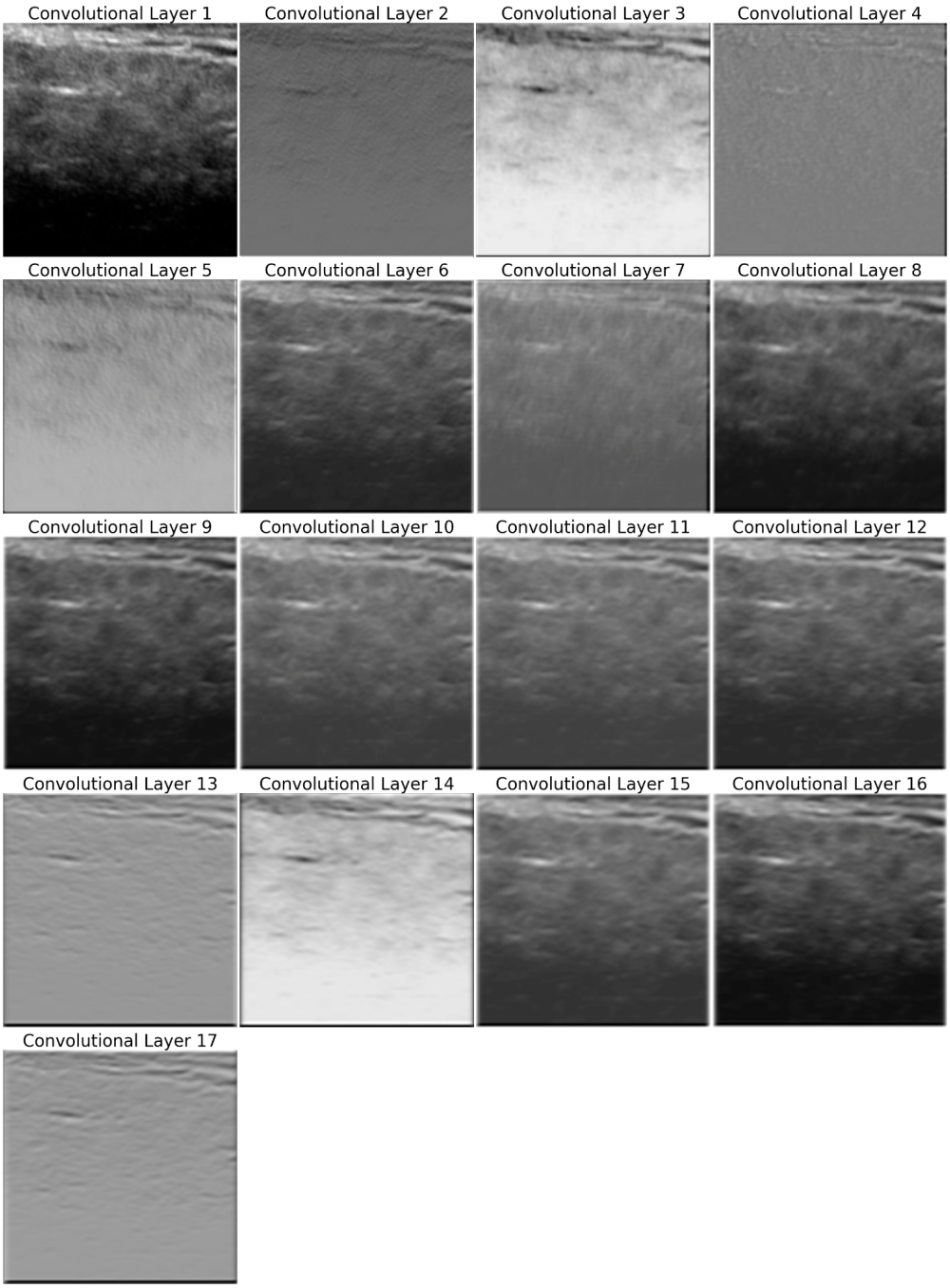


Figure 8: Visualization of the Trained VGG Convolutions

Testing

The testing procedure is the same across all three models. The image data had been split into separate folders during preprocessing before creating the model. As a result, the models have not "seen" any testing images before evaluation, enabling a non-biased assessment of their performances. Testing requires uploading the testing dataset into our model via the prediction

tool provided by the KNN and CNN built-in commands. Once the result was received from the model, it was compared to the actual data labels, and a confusion matrix was produced. This procedure was performed once for both variations: OMERACT scores and disease status for each model. The result of the final prototype testing is then compared to the baseline models.

Results

The test results of all three models are summarized in confusion matrices. The green squares represent the number of images correctly predicted, the red squares represent the number of images incorrectly predicted, the gray squares represent the percentage of correctly predicted images in each row/column, and the blue squares represent the overall accuracy.

Figure 9-1 is the confusion matrix of the KNN SjS detection model. It achieved 99.3% overall accuracy and 20% and 99.9% in negative and positive labels, respectively. The model's true negative rate is 55.1% to 104.7% more than its true positive rate with 95% confidence interval. This shows that the KNN model more accurately predicts negative labels.

Figure 9-2 is the confusion matrix of the KNN OMERACT scoring model. It achieved 54.5% overall accuracy and 56.8%, 50.8%, 61%, and 3% in scores 0-3 respectively.

Figure 9-3 is the confusion matrix of the CNN SjS detection model. It achieved 99.4% overall accuracy and 20% and 100% in negative and positive labels, respectively. The model's true negative rate is 104.8% to 55.2% more than its true positive rate with 95% confidence interval. This shows that the CNN model more accurately predicts negative labels.

Figure 9-4 is the confusion matrix of the CNN OMERACT scoring model. It achieved 46.7% overall accuracy and 36.5%, 44%, 53.3%, and 22.4% in scores 0-3, respectively.

Figure 9-5 is the confusion matrix of the VGG SjS detection model. It achieved 93% overall accuracy and 90% and 93% in negative and positive labels, respectively. The model's true negative rate is -15.6% to 21.7% more than its true positive rate with 95% confidence interval. This shows that the difference between positive and negative label accuracies is not statistically significant.

Figure 9-6 is the confusion matrix of the VGG OMERACT scoring model. It achieved 66.1% overall accuracy and 58.1%, 69.3%, 66.4%, and 53.7% in scores 0-3, respectively. The VGG OMERACT model's overall accuracy is 7.9% to 15.4% higher than the KNN OMERACT model with 95% confidence interval. This interval shows the VGG model performed statistically significantly better than the baseline KNN model.

In order to further elucidate the baseline performance of the KNN SjS detection model, ROC (Receiver Operating Characteristic) curves are plotted. The dataset is also separated into PT (parotid gland) images and SM (submandibular gland) images. The KNN model is then tested separately on the two datasets. This illustrates any difference in the two glands' ability to detect SjS.

Figure 9-7 is the ROC curve for the KNN PT model. It is a graph that illustrates the performance of binary classifier models at different thresholds. It demonstrates the plot of the true positive

rate in comparison to the false positive rate. Any point above the dotted line, the random classifier, is good at all thresholds whereas any point below it indicates error at certain thresholds. The blue line indicates a value of 0.73 for the AUC (Area Under the Curve), and since no negative rate is present the model performs well. Figure 9-8 is the ROC curve for the KNN SM model. The blue line indicates AUC of 0.66. This suggests that, at least within this dataset, PT is more predictive of SjS presence than SM. Figure 9-9 is the ROC curve for the KNN Mixed status model. The blue line indicates AUC of 0.66, and since no negative rate is present the model performs well.

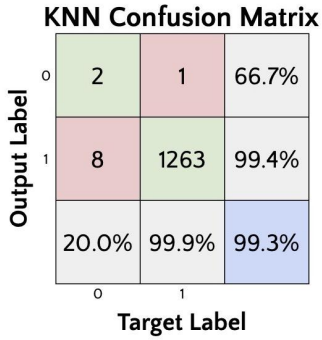


Figure 9-1: KNN SjS Detection Model

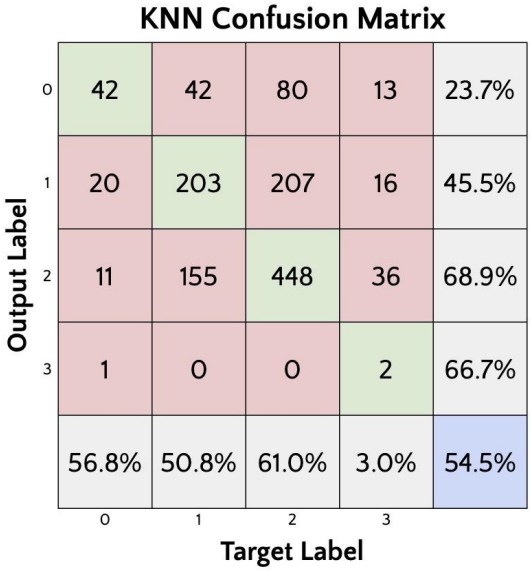
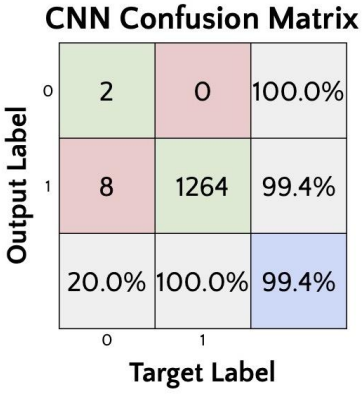


Figure 9-2: KNN OMERACT Scoring Model



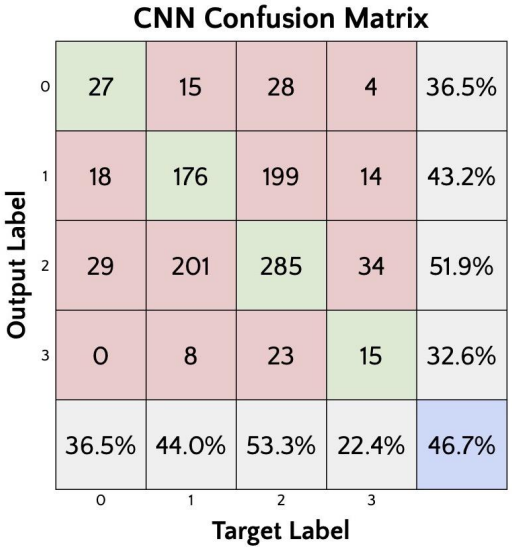


Figure 9-3: CNN SjS Detection Model

Figure 9-4: CNN OMERACT Scoring Model

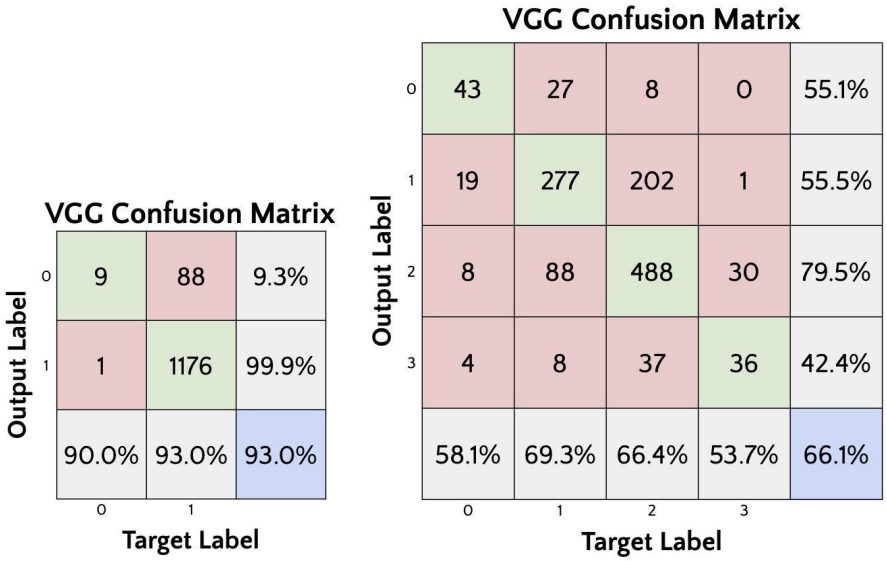
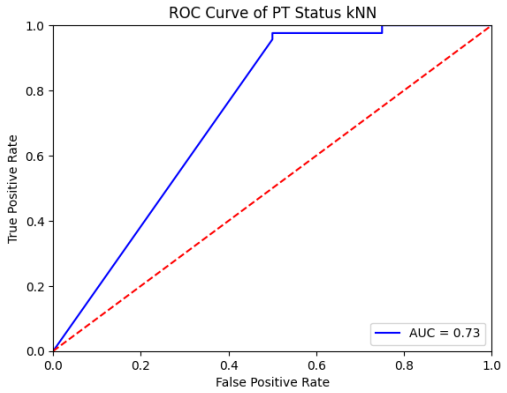


Figure 9-5: VGG SjS Detection Model

Figure 9-6: VGG OMERACT Scoring Model



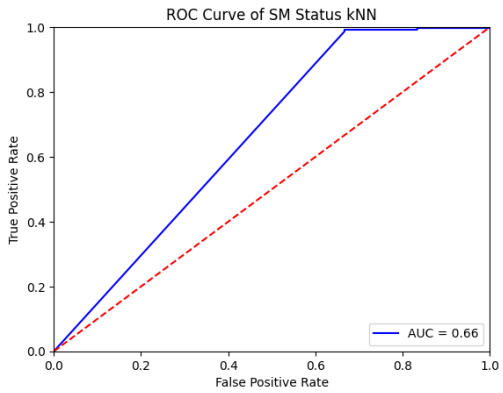


Figure 9-7: KNN PT ROC Curve

Figure 9-8: KNN SM ROC Curve

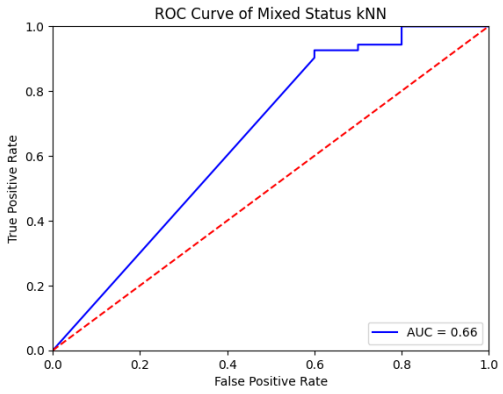


Figure 9-9: KNN Mixed ROC Curve

Discussion

SjS Detection Model

Both baseline models performed similarly with 99.3 % and 99.4% overall accuracy. The confusion matrices revealed that neither model performed well in classifying the negative subjects, achieving 20% accuracy in both cases. This is likely due to both models employing a strategy to choose the majority class label. Since over 99% of the test labels belong to the majority class, this strategy can guarantee a high accuracy without actually distinguishing between the two classes.

Despite having a lower overall accuracy than the baseline models, the VGG model achieved at least 90% accuracy in classifying negative and positive subjects. It is considerably more capable of distinguishing between the two class labels than the baseline models.

In the clinical setting, however, there exist more considerations than just the performance of the models. It is more desirable to produce false positive results than false negatives. While false positives may require the patient to obtain additional tests to confirm the diagnosis, false negatives leave the disease undetected and untreated in the patient. The VGG model has a 7% false negative rate, which is higher than the baseline rate of around 0%, but it is well within the range of false negative rates produced by physicians (S. McCoy, personal communication, December 8th, 2023).

Figure 10 is a training graph obtained from setting the learning rate to 0.0001 (a generally low learning rate). The model's behavior between the 9th and 20th epochs suggests the existence of exceptionally sharp minima. The first local minimum is associated with a narrow range of parameters in the model, since a small model update drives the loss from a local minimum to a local maximum. This is in sharp contrast to the second local minimum where numerous steps (a wide range of model parameters) correspond to similarly low loss. A transpose convolution layer that upsamples the images might be needed to increase resolution, increase parameter, thereby expand the first minimum.

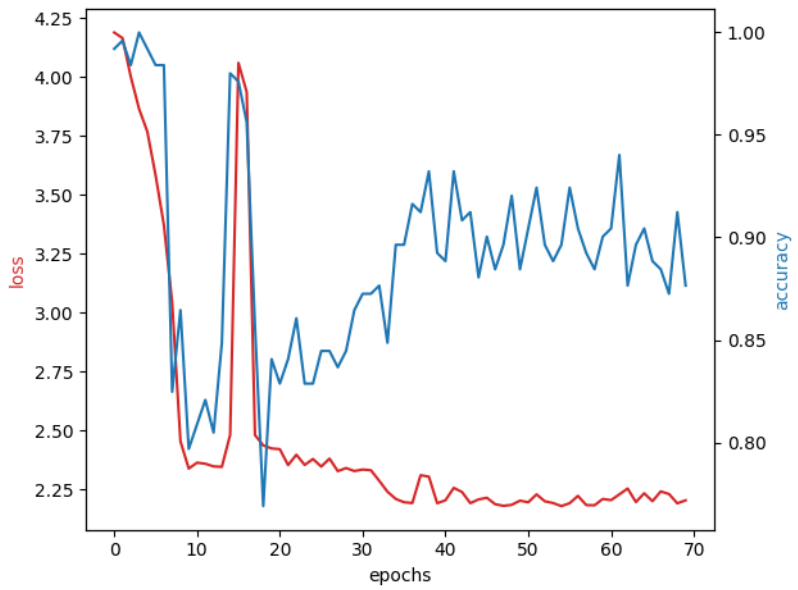


Figure 10: VGG SjS Detection Model with High Learning Rate

OMERACT Scoring Model

The low accuracy of the KNN suggests that the distance between the four clusters is short, and the distinctions are not well defined. Its confusion matrix also suggests that the four clusters overlap one another, which can be explained by physicians commonly not agreeing on the ultrasound scoring (S. McCoy, personal communication, December 8th, 2023). This is further supported by the observations that when the CNN and VGG models predict the wrong label, they mostly predict the scores adjacent to the target label.

On the one hand, if the classifications given to the models are not universally agreed upon and contain inaccuracies, then the results produced by the models will certainly be expected to contain the same inaccuracies. On the other hand, the models guarantee the same bias and error across all classifications, eliminating the variability produced by manual reviews.

Model Logistics

The processing speed of the model needs to be fast enough for real-time diagnoses. While this depends on the machines, an Apple Mac Studio with an M1 Max central processing unit can process 1273 images in under two minutes and thirty seconds with both VGG models, averaging around 0.1 seconds per image. This is fast enough for its intended usage.

The VGG models take up around 300 MB on the disk each, which is manageable for most modern machines.

Training & Evaluations

The training and testing of the current models treat each image as an individual subject. While this approach increases the amount of images in the dataset, it leaves the possibility of one patient's images belonging to both training and testing sets. As a result, the model would have already seen an ultrasound scan in the training phase similar to the one in the testing set. This could lead to overestimating the accuracy of the model.

Conclusions & Future Work

The team developed two VGG models for the task of detecting SjS with ultrasound images and ultrasound scoring with the OMERACT scale. The SjS detection model performed with 90% and 93% accuracy in positive and negative subjects, respectively. While the OMERACT scoring model only achieved 66.1% accuracy, it is expected due to the inherent inaccuracy stemming from inter-reader variability. Both models are fast enough for practical usage and compact for disk storage.

Both models can benefit from a more balanced dataset with more negative subjects so that they can be better trained and better evaluated. However, this is not always possible since it is much harder to recruit healthy subjects for ultrasound imaging than subjects already diagnosed with SjS.

An alternative model that takes multiple images as input is also a feasible architecture that increases the information given to the model. Combining multiple similar images is commonly done in MRI machine learning studies [38]. However, such a procedure aligns and merges the images before they are processed by the machine learning algorithm, which requires the images to be superimposable. This, although possible, would result in non-sensible output after merging spatially distinct ultrasound images.

An alternative approach that is more suitable to the SjS application is to perform separate convolutions on the inputs and merge the images after some number of operations, non-medical variants of which were demonstrated relatively recently [39-41]. A similar approach to this is to extract features from the images and apply a feed-forward network instead of a CNN. This has seen some success in breast cancer classifications [42, 43]. Another possibility is to apply CNN on numerical data such as age, gender, and height along with images [44].

The ideal algorithm would be able to not only take multiple images, but also numerical data on the patient as input. This system has not been used for medical image analysis, and would require experimentations with the parameters and hyperparameters. Most importantly, the question of when and how to merge the images must be addressed.

References

1. M. Ramos-Casals and J. Font, “Primary sjögren’s syndrome: Current and emergent AETIOPATHOGENIC concepts,” *Rheumatology*, vol. 44, no. 11, pp. 1354–1367, 2005. doi:10.1093/rheumatology/keh714

2. P. Brito-Zerón, S. Retamozo, and M. Ramos-Casals, “Síndrome de Sjögren,” *Medicina Clínica*, vol. 160, no. 4, pp. 163–171, 2023. doi:10.1016/j.medcli.2022.10.007

3. “Sjögren’s Disease | National Institute of Dental and Craniofacial Research,” www.nidcr.nih.gov. https://www.google.com/url?q=https://www.nidcr.nih.gov/health-info/sjogrens-disease&sa=D&source=docs&ust=1697072274306517&usg=AOvVaw3KghYeXwXBUyr2h7In5wgy (accessed Oct. 11, 2023).

4. Lorenzon, Michele et al. “Salivary Gland Ultrasound in Primary Sjögren's Syndrome: Current and Future Perspectives.” *Open access rheumatology : research and reviews* vol. 14 147-160. 1 Sep. 2022, doi:10.2147/OARRR.S284763

5. N. S. C. and O. Branch, “Sjögren’s Syndrome,” National Institute of Arthritis and Musculoskeletal and Skin Diseases, Apr. 07, 2017. https://www.niams.nih.gov/health-topics/sjogrens-syndrome/basics/symptoms-causes

6. V. Fana, U. M. Dohn, S. Krabbe, and L. Terslev, “Application of the omeract grey-scale ultrasound scoring system for salivary glands in a single-centre cohort of patients with suspected Sjögren’s syndrome,” *RMD Open*, vol. 7, no. 2, 2021. doi:10.1136/rmdopen-2020-001516

7. “Sara McCoy, MD, Phd,” | Department of Medicine, University of Wisconsin–Madison, https://www.medicine.wisc.edu/people-search/people/staff/5206/McCoy_Sara (accessed Dec. 9, 2023).

8. “Sjogren’s syndrome,” Mayo Clinic, https://www.mayoclinic.org/diseases-conditions/sjogrens-syndrome/symptoms-causes/syc-20353216 (accessed Oct. 11, 2023).

9. O. Al Tabaa et al., “Normal salivary gland ultrasonography could rule out the diagnosis of Sjögren’s syndrome in anti-ssa-negative patients with SICCA syndrome,” *RMD Open*, vol. 7, no. 1, 2021. doi:10.1136/rmdopen-2020-001503

10. “Blood Culture Contamination: An Overview for Infection Control and Antibiotic Stewardship Programs Working with the Clinical Laboratory.” Available: https://www.cdc.gov/antibiotic-use/core-elements/pdfs/fs-bloodculture-508.pdf

11. E. Staff, “Schirmer’s test: A test for dry eyes,” The Eye News, https://theyenews.com/schirmers-test/ (accessed Oct. 11, 2023).

12. A. Delgado, “Salivary duct stones: Causes, symptoms, and diagnosis,” Healthline, https://www.healthline.com/health/salivary-duct-stones#outlook (accessed Oct. 11, 2023).

13. “Labial gland (lip) biopsy,” Johns Hopkins Sjögren’s Center, https://www.hopkinssjogrens.org/disease-information/diagnosis-sjogrens-syndrome/labial-gland-lip-biopsy/ (accessed Oct. 11, 2023).

14. Al’Aref, Subhi J et al. “Clinical applications of machine learning in cardiovascular disease and its relevance to cardiac imaging.” *European Heart Journal,* vol. 40,24 (2019): 1975-1986. doi:10.1093/eurheartj/ehy404

15. J. Raitoharju, “Convolutional Neural Networks,” *Deep Learning for Robot Perception and Cognition*, pp. 35–69, 2022. doi:10.1016/b978-0-32-385787-1.00008

16. “Crossentropyloss,” CrossEntropyLoss - PyTorch 2.1 documentation, https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html (accessed Dec. 1, 2023).

17. N. Cui, “Applying Gradient Descent in Convolutional Neural Networks” *IOP Conf. Series: Journal of Physics*: Conf. Series 1004 2018. doi: 10.1088/1743-6596/1004/1/012027

18. A. Roy, “An introduction to Gradient Descent and Backpropagation,” Towards Data Science, https://towardsdatascience.com/an-introduction-to-gradient-descent-and-backpropagation-81648bdb19b2 (accessed 12/9/2023)

19. D. Westreich, J. Lessler, and M. J. Funk, “Propensity score estimation: Neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression,” *Journal of Clinical Epidemiology*, vol. 63, no. 8, pp. 826–833, 2010. doi:10.1016/j.jclinepi.2009.11.020

20. Sarkar, M, and T Y Leong. “Application of K-nearest neighbors algorithm on breast cancer diagnosis problem.” Proceedings. *AMIA Symposium* (2000): 759-63

21. L. Frederick, “Implementation of Breiman’s Random Forest Machine Learning Algorithm,” *ECE591Q Machine Learning Journal Paper*. Fall 2005.

22. Ji Zhu & Trevor Hastie (2005) “Kernel Logistic Regression and the Import Vector Machine,” *Journal of Computational and Graphical Statistics*, 14:1, 185-205, DOI: 10.1198/106186005X25619

23. Suthaharan, S. (2016). “Support Vector Machine”. In: Machine Learning Models and Algorithms for Big Data Classification. *Integrated Series in Information Systems*, vol 36. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7641-3_9

24. Mukherjee, S. "The Annotated ResNet-50" Toward Data Science. August, 2022. https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

25. A. R. N. Aouichaoui, R. Al, J. Abildskov, and G. Sin, “Comparison of group-contribution and machine learning-based property prediction models with uncertainty quantification,” *31st European Symposium on Computer Aided Process Engineering*, pp. 755–760, 2021. doi:10.1016/b978-0-323-88506-5.50118-2

26. Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *ICLR* 2015.

27. Cao, Zili et al. “BND-VGG-19: A deep learning algorithm for COVID-19 identification utilizing X-ray images.” *Knowledge-based systems* vol. 258 (2022): 110040. doi:10.1016/j.knosys.2022.110040

28. Zhang, J. “UNet- Line by Line Explanation” Towards Data Science. October 2019. https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5

29. Noble, W. “What is a support vector machine?,” *Nat Biotechnol* 24, 1565–1567 (2006). https://doi.org/10.1038/nbt1206-1565

30. M.N.A.H. Sha’abani, N. Fuad, N. Jamal, M. Ismail, “kNN and SVM Classification for EEG: A Review” In *ECCE2019 Proceedings of the 5th International Conference on Electrical, Control & Computer Engineering*, vol. 632, pp. 544-554, 2019.

31. M. Shapiee, M. Ibrahim, M.Razman, M. Abdullah, R. Musa, M. Hassan, A. Majeed, “The Classification of Skateboarding Trick Manoeuvres Through the Integration of Image Processing Techniques and Machine Learning,” *In ECCE2019 Proceedings of the 5th International Conference on Electrical, Control & Computer Engineering*, vol. 632, pp. 544-554, 2019.

32. Yin, X. X., Sun, L., Fu, Y., Lu, R., & Zhang, Y. “U-Net-Based Medical Image Segmentation,” *Journal of healthcare engineering*, 2022, 4189781. https://doi.org/10.1155/2022/4189781

33. Devansh, “How does batch size impact your model learning,” Medium, https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa (accessed Dec. 3, 2023).

34. N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” in *ICLR* 2017.

35. “Exponentiallr,” ExponentialLR - PyTorch 2.1 documentation, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ExponentialLR.html (accessed Dec. 4, 2023).

36. “Sklearn.utils.class_weight.compute_class_weight,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html (accessed Dec. 4, 2023).

37. Kingma, D. P. & Ba, J. Adam, “A method for stochastic optimization.” In *ICLR* 2015.

38. T. J. Hendrickson et al., “BIBSNet: A deep learning baby image brain segmentation network for MRI scans”, 2023. doi:10.1101/2023.03.22.533696

39. Z. Wang et al., “Multi-input convolutional network for Ultrafast Simulation of Field evolvement,” *Patterns*, vol. 3, no. 6, p. 100494, 2022. doi:10.1016/j.patter.2022.100494

40. Y. Sun, L. Zhu, G. Wang, and F. Zhao, “Multi-input convolutional neural network for flower grading,” *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–8, 2017. doi:10.1155/2017/9240407

41. M. Seeland and P. Mäder, “Multi-view classification with Convolutional Neural Networks,” *PLOS ONE*, vol. 16, no. 1, 2021. doi:10.1371/journal.pone.0245230

42. S. Pramanik, D. Bhattacharjee, and M. Nasipuri, “Multi-resolution analysis to differentiate the healthy and unhealthy breast using breast thermogram,” *2016 International Conference on Systems in Medicine and Biology (ICSMB)*, 2016. doi:10.1109/icsmb.2016.7915085

43. V. Madhavi and C. B. Thomas, “Multi-view breast thermogram analysis by fusing texture features,” *Quantitative InfraRed Thermography Journal*, vol. 16, no. 1, pp. 111–128, 2019. doi:10.1080/17686733.2018.1544687

44. R. Sánchez-Cauce, J. Pérez-Martín, and M. Luque, “Multi-input convolutional neural network for breast cancer detection using thermal images and clinical data,” *Computer Methods and Programs in Biomedicine*, vol. 204, p. 106045, 2021. doi:10.1016/j.cmpb.2021.106045

Appendix

Product Design Specifications

Machine Learning for Salivary Gland Ultrasound Scoring

Section 304

Product Design Specifications

09/22/2023

Team:

Richard Yang (tyang296@wisc.edu)	Team Lead
Yousef Gadalla (ygadalla@wisc.edu)	Communicator
Brandon Drew (bsdrew2@wisc.edu)	BSAC
Dhruv Nadkarni (dnadkarni@wisc.edu)	BWIG
Siya Mahajan (mahajan24@wisc.edu)	BWIG
Aran Viswanath (viswanath3@wisc.edu)	BPAG

Background

Sjögren's syndrome (SjS) is a systemic autoimmune disease (SAD) that causes dysfunction of the exocrine glands (mainly the salivary and lacrimal glands) with patients often showing persistent dryness of the mouth and eyes [1, 2]. According to estimations, two to four million people in the United States have SjS; however, only one million have been diagnosed, likely due to the nonspecific diagnostic guidelines and the heterogeneous nature of the disease [3]. The current standard of care of the client is to perform at least baseline salivary gland ultrasounds (of the submandibular and parotid glands) in patients who potentially have SjS. For some higher-risk individuals, regularly scheduled salivary gland ultrasounds are performed.

Function

The problem arises within the current Outcome Measures in Rheumatoid Arthritis Clinical Trials (OMERACT) ultrasound grading system, which requires subjective opinions and lacks nuance. As a result, a machine learning approach is proposed to reduce inter-reader variability and to provide a more exact prognosis. The proposed algorithm takes ultrasound grayscale images as input and outputs SjS positive or SjS negative.

Client requirements

The following is a list of client requirements:

- The algorithm needs to take ultrasound grayscale images as input and output binary labels of SjS positive or SjS negative.
- It is preferable that the algorithm can be processed in real-time, such that the physicians can receive the algorithm's output immediately after the patient's ultrasound procedure.
- Images must be de-identified before they can be used for training
- Generalizability to other Rheumatic diseases and Emergency Medical Technician (EMT) applications is preferable

Design requirements

1. Physical and Operational Characteristics

a. Performance requirements:

The product will be a machine learning program that is run on hospital computers and analyzes salivary gland ultrasound images. The program must provide an accurate classification of the images and determine whether the patient has SjS or not.

The program will be utilized in clinical settings post-ultrasound readings. This means that the device could potentially be used many times a day, depending on clinic hours and number of patients that need ultrasounds. To ensure that no long waits occur for patients, the machine learning algorithm should be able to generate results within 15 minutes. A first-in-first-out (FIFO) queue structure will be used to ensure that no tasks are skipped due to processing time.

b. Safety:

As this is a machine learning program, there should not be any safety concerns for users; however, as this algorithm will be utilized in diagnosing SjS, it is very important that the algorithm works properly. Otherwise, any missed diagnosis could result in patient's not receiving proper treatment for SjS, which potentially can cause increased health risks and concerns [4].

c. Accuracy and Reliability:

Since this is a highly adaptable product, it will gain accuracy as it is presented with more data. Thus it will be created to increase in reliability with additional time and usage. The models will be evaluated by first partitioning the dataset into training and validation sets with a 7:3 ratio respectively. The model will then be trained on the training set and evaluated with the validation set. The output of which will be put into confusion matrices and the accuracy results as well as Receiver Operating Chracteristic (ROC) curve will be generated.

A baseline performance (performance of a simple model with the same training data as the final model) will first be assessed using a support vector machine (SVM), and the goal is to perform better than the baseline with either a more complicated deep neural network (DNN) or an established model like the ResNet-50. Ideally, the accuracy should be greater than or equal to 95%.

In practice, especially in the early stages of the product, a physician's opinion might be needed to supplement the output of the algorithm.

d. Life in Service:

In light of a better scoring system, or a software/hardware change this product is not compatible with, this product may become obsolete. As a machine learning algorithm, however; it is can be updated by the team in the future when new data becomes available to improve performance.

e. Shelf Life:

Given that the system is updated in order to stay relevant with the software and hardware it will be run on, the shelf life of this product is infinite.

f. Operating Environment:

The product is designed to operate in clinical environments, primarily on computers that can run the code. The code can run on any operating system but requires Python to be installed on the computer for the program to run if the client prefers the program in a .py or .ipynb format. If the code is built as an executable software, no Python is required.

g. Ergonomics:

The sole restrictions would be the usage of an admissible computer, the requirement of Python dependent upon the client's preferred file format, and patient permission for their images to be run through the program.

h. Size:

As the product is software oriented, there are no physical size restrictions or requirements.

i. Weight:

The project design is software based, and thus weight is not applicable in terms of software. The weight required by the client ranges, as they require a workstation, whether a laptop or desktop, to run the software and process images.

j. Materials:

There only will be a software aspect to the product. So, since there will be no hardware, no physical materials are needed for this product. As for software, PyTorch will be used for the machine learning framework, and GitHub will be necessary for maintenance. Depending on the processing speed of the final model, a GPU module might be required to decrease processing time.

k. Aesthetics, Appearance, and Finish:

There is no hardware, so there will be no color, shape, or form texture requirements. This product consists of only software, so aesthetics, appearance, and finish are not applicable.

2. Production Characteristics

a. Quantity:

Only one program has to be written to fulfill the requirements. This program will then be used on any device the client wishes to use.

b. Target Product Cost:

Since this device only consists of software, there will be no manufacturing costs.

3. Miscellaneous

a. Standards and Specifications:

The project concerns human data; thus, a few issues must be addressed, namely the acquisition of human data, de-identification protocols, and working with de-identified data.

De-identified ultrasound images will be provided by the client; however, if any additional data acquisition is to take place, per 21 CFR 56.102, any data acquisition from human subjects shall fall under the definition of clinical investigation and:

> must meet the requirements for prior submission to the Food and Drug Administration under section 505(i) or 520(g) of the act, or need not meet the requirements for prior submission to the Food and Drug Administration under these sections of the act, but the results of which are intended to be later submitted to, or held for inspection by, the Food and Drug Administration as part of an application for a research or marketing permit. [5]

Human subject shall be defined as an individual who is or becomes a participant in this project, as the subject of ultrasound imaging [5]. In such a case, informed consent of the participants and IRB approval must be obtained. Per FDA guidelines, adequate information that allows an informed decision must be provided, participants' understanding of the aforementioned information should be facilitated, adequate time must be allocated for the participants to ask questions and discuss protocols with family and friends, and voluntary participation agreement must be obtained, and the participants should be updated with more information as research progresses [6].

In the case of working with de-identified data, which is defined as there is no reasonable basis to believe that the information can be used to identify an individual under 45 CFR 164.514, HIPAA

Privacy Rule “does not restrict the use or disclosure of de-identified health information, as it is no longer considered protected health information” [7, 8].

Per 45 CFR 164.514(b), HIPAA provides two de-identification methods: 1) Expert determination and 2) Safe harbor. The former requires “a person with appropriate knowledge of and experience with generally accepted statistical and scientific principles and methods for rendering information not individually identifiable” while the latter requires the removal of 18 types of identifiers, including but not limited to name, address, and phone number [7].

b. Customer:

The primary customers of this product are Hospitals, Rheumatologists, and EMTs.

c. Patient-related concerns:

This algorithm must provide accurate diagnoses to prevent the consequences of a false negative or false positive result. Minimizing the number of inaccurate results is crucial as false negatives can lead to a patient not receiving the treatment that they need and false positives can lead to patients being exposed to unnecessary treatments and medications. It is also important that patient health information is not disclosed without proper notice as outlined in 45 CFR 164.520 [9].

d. Competition:

Other methods of detecting SjS include blood and urine tests, Schirmer tear test, Sialography, Salivary scintigraphy, and biopsy [10-14]. While these tests are less subjective than the current OMERACT grading system, they are significantly more invasive and time consuming than ultrasound scans. Additionally, a patent titled 'Method for Developing a Machine Learning Model of a Neural Network for Classifying Medical Images' by Tienovix LLC claims protection for a machine learning model relating to Data Collection, Feature Definition, Image Analysis, Labeling, Data Splitting, Neural Network Training, Training Metrics, Threshold Evaluation, Validation Process, Validation Metrics, and Model Storage [15]. This patent describes a method for obtaining medical image data, including ultrasound images, and trains a machine learning model to analyze features in the image and validate that model’s accuracy with a training set. This method can be applied to diagnose SjS by training a machine learning model to recognize features of salivary gland ultrasound scans and grade them based on their characteristics. Another patent titled “Machine-aided workflow in ultrasound imaging”, protects the use of computer-aided classification to detect objects inside of the body [16]. While this patent describes the classification of organs in an ultrasound scan, a similar model could be used to distinguish the salivary glands in ultrasound scans of potential SjS patients.

Reference

1. M. Ramos-Casals and J. Font, “Primary sjögren’s syndrome: Current and emergent AETIOPATHOGENIC concepts,” Rheumatology, vol. 44, no. 11, pp. 1354–1367, 2005. doi:10.1093/rheumatology/keh714

2. P. Brito-Zerón, S. Retamozo, and M. Ramos-Casals, “Síndrome de Sjögren,” Medicina Clínica, vol. 160, no. 4, pp. 163–171, 2023. doi:10.1016/j.medcli.2022.10.007

3. S. S. Kassan and H. M. Moutsopoulos, “Clinical manifestations and early diagnosis of Sjögren syndrome,” Archives of Internal Medicine, vol. 164, no. 12, p. 1275, 2004. doi:10.1001/archinte.164.12.1275

4. Lorenzon, Michele et al. “Salivary Gland Ultrasound in Primary Sjögren's Syndrome: Current and Future Perspectives.” Open access rheumatology : research and reviews vol. 14 147-160. 1 Sep. 2022, doi:10.2147/OARRR.S284763

5. 21 CFR § 56.102, 1981

6. O. of the Commissioner, “Informed consent for clinical trials,” U.S. Food and Drug Administration, https://www.fda.gov/patients/clinical-trials-what-patients-need-know/informed-consent-clinical-trials (accessed Sep. 21, 2023).

7. 45 CFR § 164.514, 2000

8. O. for C. R. (OCR), “Methods for de-identification of phi,” HHS.gov, https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html#rationale (accessed Sep. 21, 2023).

9. 45 CFR § 164.520, 2000

10. “Blood and urine tests,” Johns Hopkins Sjögren’s Center, https://www.hopkinssjogrens.org/disease-information/diagnosis-sjogrens-syndrome/blood-and-urine-tests/ (accessed Sep. 19, 2023).

11. A.-L. Stefanski et al., “The diagnosis and treatment of Sjögren’s syndrome,” Deutsches Ärzteblatt international, 2017. doi:10.3238/arztebl.2017.0354

12. N. Ohbayashi, I. Yamada, N. Yoshino, and T. Sasaki, “Sjögren syndrome: Comparison of assessments with mr sialography and conventional sialography.,” Radiology, vol. 209, no. 3, pp. 683–688, 1998. doi:10.1148/radiology.209.3.9844659

13. I. Umehara, I. Yamada, Y. Murata, Y. Takahashi, N. Okada, and H. Shibuya, “Quantitative evaluation of salivary gland scintigraphy in Sjörgen’s syndrome,” Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine, vol. 40, no. 1, pp. 64–69, Jan. 1999, Accessed: Sep. 22, 2023. [Online]. Available: https://www.google.com/url?q=https://pubmed.ncbi.nlm.nih.gov/9935059/&sa=D&source=docs&ust=1695404911950832&usg=AOvVaw1wYoNw1_pR1FXK9wno3T62

14. “Diagnosing sjogren’s syndrome,” Patient Care at NYU Langone Health, https://nyulangone.org/conditions/sjogrens-syndrome/diagnosis (accessed Sep. 19, 2023).

15. W. R. Buras, C. S. Russell, and K. Q. Nguyen, “Method for developing a machine learning model of a neural network for classifying medical images.” https://patents.google.com/patent/US11017695B2 (accessed Sep. 22, 2023).

16. 라오빔바 and 구라카이스마일 엠, “Machine-aided workflow in ultrasound imaging.” https://www.google.com/url?q=https://patents.google.com/patent/KR20190053807A/en&sa=D&source=docs&ust=1695404911953234&usg=AOvVaw0FAXbca4VpQdMv60yQYrQ2 (accessed Sep. 22, 2023).