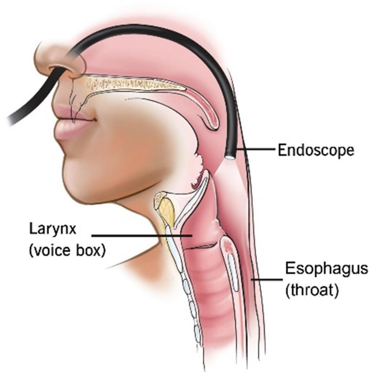
Software Engineering Department
Ort Braude College
Course 61771: Extended Project in Software Engineering

Laryngoscopy image analysis using deep learning approach



In Partial Fulfillment of the Requirements for
Final Project in Software Engineering (Course 61771)
Karmiel – June 2020

Authors:

Sagi Vishnia
Arie Shnaiderman

Contents

1. INTRODUCTION

Many diseases are diagnosed by identifying visual signs. The doctor examines which visual signs exists and determines the type and the severity based on those signs. When making a diagnosis based on visual signs, there are factors that can cause a wrong diagnosis, such as limited time to perform the test, the complexity of identifying the severity of the symptom, and dependence on the human eyes which can lead to subjective interpretation of the symptoms. As a result, the doctor may misinterpret the symptoms and end up making a wrong diagnosis.

Nowadays, uses of image processing and deep learning applications in the medical world are getting more and more popular. With the help of applications of image processing systems, the possible errors made during the diagnosis can be avoided, and the medical data can be examined in shorter time and be more detailed as well.

In our project, we focused on Laryngopharyngeal reflux (LPR) disease and diagnosing the disease by analyzing endoscopic images.

LPR [1] is the result of retrograde flow of gastric contents to the larynx. The larynx, commonly called the voice box, is an organ in the top of the neck involved in breathing, producing sound, and protecting the trachea against food aspiration. When swallowing, the food passes down the throat through the esophagus (swallowing tube) and into the stomach. A muscle called the lower esophageal sphincter controls the opening between the esophagus and the stomach. The muscle remains tightly closed except for the swallowing operation. When this muscle fails to close, the acid-containing contents of the stomach can travel back up into the esophagus. This backward movement is called reflux [2], as shown in Figure 1. Signs and symptoms of LPR include red, swollen, or irritated larynx, sore throat, sensation of a lump in the throat, difficulty swallowing etc.

Laryngoscopy [3] is the first-line investigation method that is usually performed to make a diagnosis of LPR. During the examination, laryngoscope with a camera passes through the patient nose and feeds back images of the larynx to a monitor. The laryngoscope allows the doctor to directly look inside the throat and diagnose if the patient has LPR. The procedure might be very unpleasant for the patient, as a result, the patient constantly swallows, and it causes the laryngoscope to move outside the larynx area. The doctor has limited time to perform the procedure and during this time he needs to insert the laryngoscope to the patient nose, look at the monitor, examinee the pictures and to diagnose if the patient has LPR or not. As a result of those difficulties and the complexity of the procedure, the chance of making wrong diagnoses is high. When failing to recognize LPR, patients have prolonged symptoms and delayed healing, while in case of wrong positive LPR diagnosis, the patient is going to suffer from not required treatment that may cause harm and unnecessary costs.

This project is a part of a bigger scale project. Initial steps in the study of the disease have already been taken and various tools that try to diagnose some of the symptoms of the disease have been developed.

The goal of our project was to build a tool for analyzing the endoscopic images obtained from the endoscopic examination to achieve more accurate diagnosis. In this project we used convolutional neural network (CNN), that gets the endoscopic pictures of the patient and give the doctors an objective diagnosis if he has LPR or not.

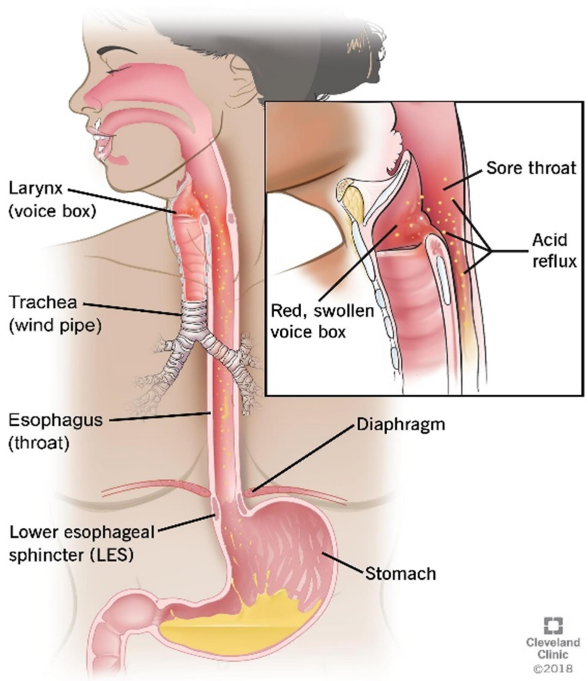


Fig. 1: example of the reflux process

2. THEORY

2.1 LPR Diagnosis Using the Reflux Finding Score (RFS)

Even though the symptoms and findings of laryngopharyngeal reflux (LPR) have been described, the clinical diagnosis is sometimes elusive and difficult. Symptoms can occur in the absence of conclusive laryngeal physical findings, and they can be nonspecific. Furthermore, some of the symptoms used to diagnose the disease can be caused not only by LPR, but also by other diseases such as laryngospasm, bronchiectasis, and chronic rhinosinusitis [4]. As a result of the difficulties that described above, the diagnosis of LPR is based on a combination of factors, including symptoms, laryngeal findings, and diagnostic test results.

In order to resolve the difficulty to diagnose LPR and to identify the most specific laryngoscopy signs, the Reflux Finding Score (RFS) [4] was developed based on the findings of fiberoptic laryngoscopy. This scale evaluates eight items that comprise the most common laryngoscopy findings in patients with LPR: subglottic edema, ventricular obliteration, erythema or hyperemia, vocal fold edema, generalized laryngeal edema, posterior commissure hypertrophy, granuloma or granulation tissue and excess mucus in the larynx. Each item is scored according to severity, location and presence or absence. Then we sum over all 8 scores for a maximal total score of 26. Patients presenting a score of 7 or higher are classified as having LPR. The scoring scheme can be viewed in Table 1.

Currently, doctors widely use RFS in order to diagnose LPR. When performing the endoscopy, the doctors looks through the monitor on the patient's larynx, grades the visual symptoms according to the RFS table, and finally summarizes the score and gives a diagnosis.

Reflux Finding Score

Subglottic Edema	2 = present 0 = absent	
Ventricular Obliteration	2 = partial 4 = complete	
Erythema/Hyperemia	2 = arytenoids only 4 = diffuse	
Vocal Fold Edema	1 = mild 2 = moderate 3 = severe 4 = polypoid	
Diffuse Laryngeal Edema	1 = mild 2 = moderate 3 = severe 4 = obstructing	
Posterior Commissure Hypertrophy	1 = mild 2 = moderate 3 = severe 4 = obstructing	
Granuloma/Granulation	2 = present 0 = absent	
Thick Endolaryngeal Mucus	2 = present 0 = absent	
	Total:	

Table 1 – Reflux Finding Score (RFS)

In our project, we relied on these RFS signs to automatically diagnose LPR in the patient. The signs of finding reflux finding are:

1. Subglottic edema (pseudo sulcusvocalis):

One of the most common laryngeal findings of LPR is pseudo sulcusvocalis, as shown in Figure 2. This term is rather self-descriptive, and it refers to edema of the undersurface of the vocal fold that extends from the anterior commissure to the posterior larynx and creates the appearance of a groove or sulcus. A vocal fold with pseudosulcus has been described as having the appearance of a partially

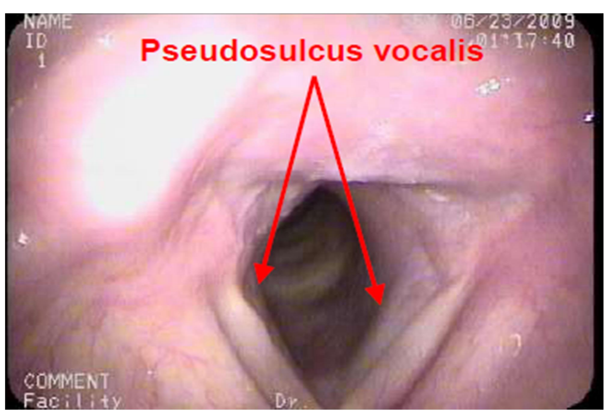


Fig. 2: Larynx with subglottic edema symptom. The symptom marked with red arrow.

open hot-dog bun.

2. Ventricular obliteration

The laryngeal ventricle is the space between the true and false vocal folds, as shown in Figure 3. When both sets of vocal folds become swollen, this space can become diminished or completely obliterated. With ventricular obliteration, the medial edge of the ventricular bands usually becomes broad and swollen.

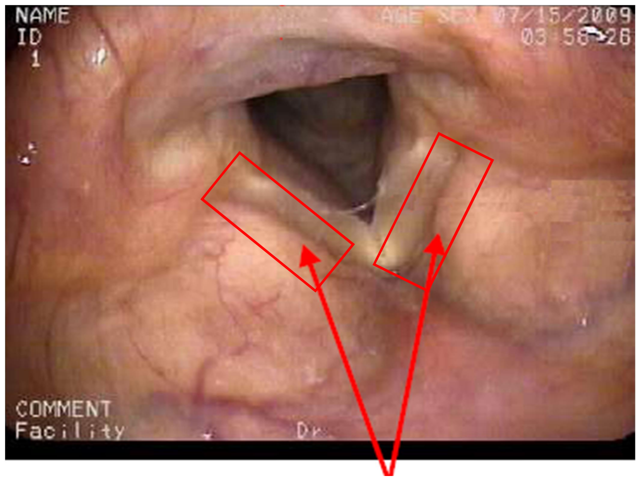


Fig. 3: Larynx with ventricular obliteration symptom. The symptom marked with red arrow and red box.

3. Erythema/hyperemia

Erythema is redness of the skin or mucous membranes, caused by hyperemia (increased blood flow) in superficial capillaries. This symptom is not unique to LPR alone and it occurs with any skin injury, infection, or inflammation. Although an attempt has been made to quantify laryngeal erythema, the evaluation of redness on endoscopy is often difficult and depends on the type of endoscope, light

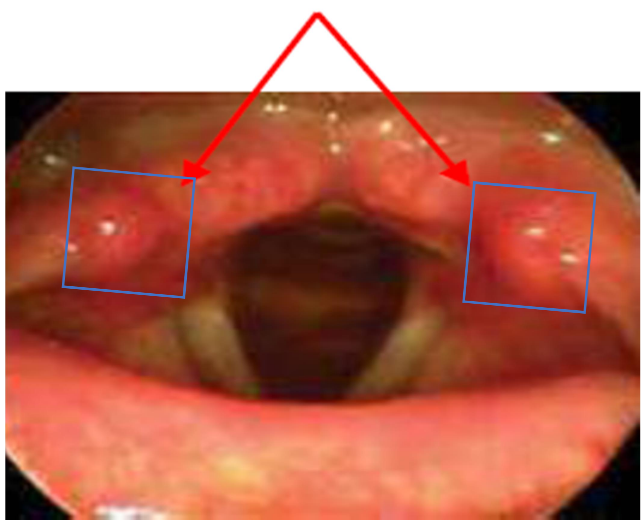


Fig. 4: Larynx with Erythema/hyperemia symptom. The symptom marked with red arrow and blue box.

source and video monitor utilized. Example to the symptom is shown in Figure 4.

4. Vocal fold edema

Edema is an abnormal accumulation of fluid in the body’s tissue and can cause severe pain. In LPR, vocal fold edema can range from mild to end-stage polypoid degeneration. Grade 1 edema is characterized by rounding of the free edges, and grade 2 by pseudosulcus. Grade 3 edema features sessile changes, and grade 4 represents polypoid degeneration. The vocal fold edema can be seen in

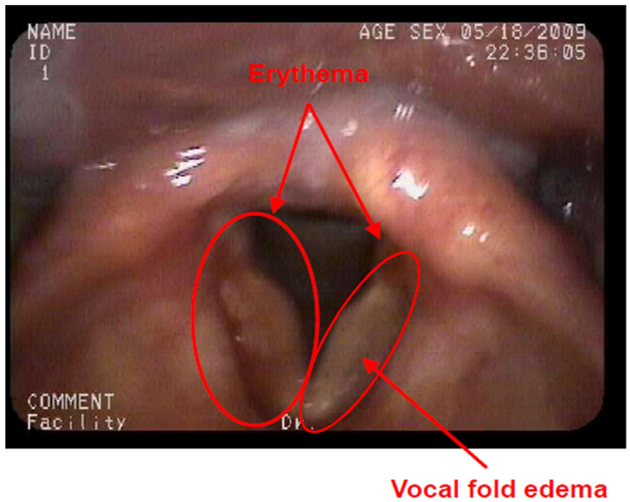


Fig. 5: Larynx with vocal fold edema symptom.

Figure 5.

5. Diffuse laryngeal edema

The presence of diffuse laryngeal edema is a somewhat subjective parameter and refers to the relative ratio of the endolaryngeal airway to the whole larynx, as shown in Figure 6. Grade 1 denotes any degree of diffuse laryngeal edema. In grade 2 laryngeal edema, the lumen is encroached, usually by posterior laryngeal hypertrophy. Grade 3 represents diffuse pachydermialaryngis ("elephant skin" of the larynx), wherein the ratio of the airway to the overall laryngeal diameter is less than one-half. Grade 4 denotes some degree of clinical airway obstruction.

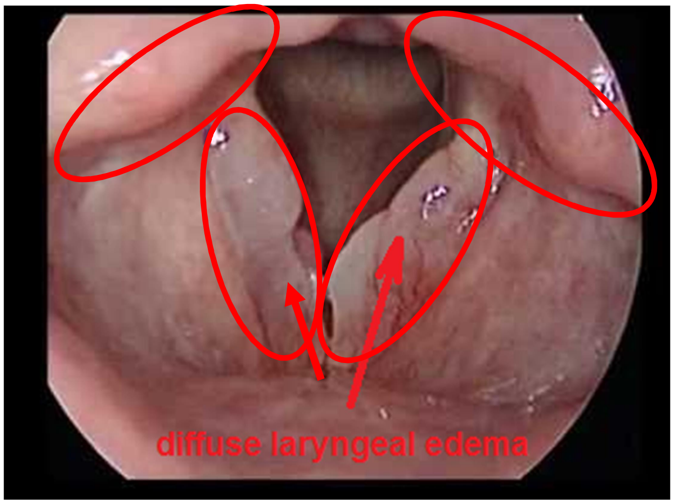


Fig. 6: Larynx with diffuse laryngeal edema symptom.

6. Posterior commissure hypertrophy

Posterior commissure hypertrophy is edema of the back side of the larynx. Mucosal hypertrophy of the posterior commissure epithelium is graded as mild when there is a mustache-like appearance of the posterior commissure mucosa, moderate when the posterior commissure is swollen enough to create a straight line across the back of the larynx, severe when there is bulging of the posterior larynx into the airway, and obstructing when a significant portion of the airway is obliterated, as shown in Figure 7.

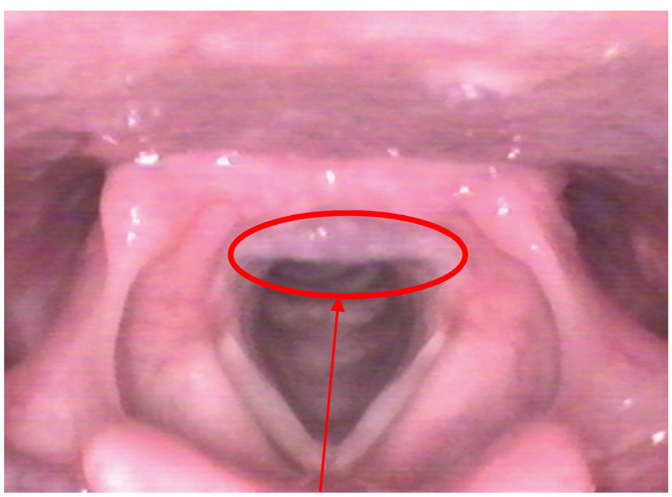


Fig. 7: Larynx with posterior commissure hypertrophy symptom.

7. Granuloma/ Granulation

Granulomas caused by backflow of stomach fluids to the voice box (LPR) or by trauma from forceful contact of vocal folds during vocal misuse or overuse. Granuloma or granulation tissue anywhere in the larynx is graded as a positive LPR finding, as shown in Figure 8.

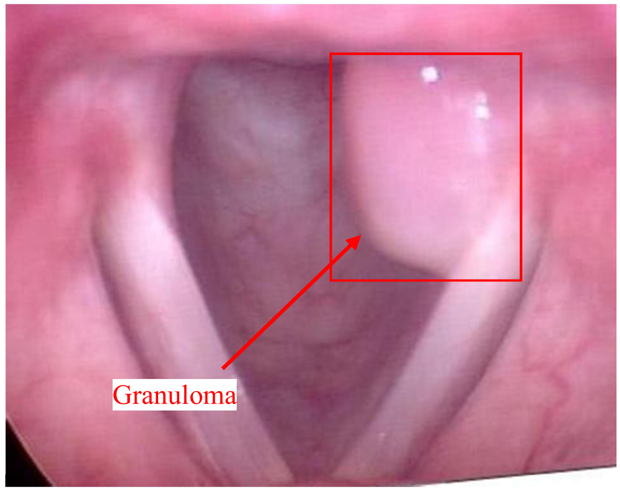


Fig. 8: Larynx with granuloma/ granulation symptom.

8. Thick Endolaryngeal Mucus

Thick, white endolaryngeal mucus on the vocal folds or elsewhere in the endolarynx is graded as a

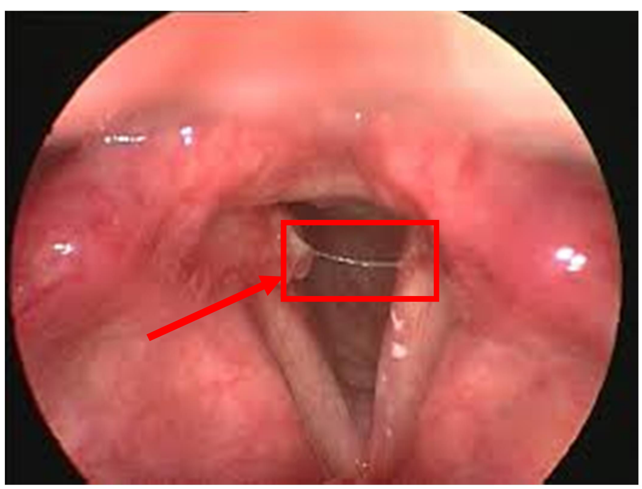


Fig. 9: Larynx with thick endolaryngeal mucus symptom.

positive physical finding, as shown in Figure 9.

2.2 Deep Learning Using Convolutional Neural Network

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

2.2.1 Artificial Neural Network – ANN

Artificial Neural Networks (ANNs) [5] are computational processing systems of which are heavily inspired by way biological nervous systems (such as the human brain) operate. ANNs are mainly comprised of a high number of interconnected computational nodes (referred to as neurons), of which work entwine in a distributed fashion to collectively learn from the input in order to optimize its final output .The basic structure of a ANN can be modelled as shown in Figure 10. We would load the input, usually in the form of a multidimensional vector to the input layer and then will distribute it to the hidden layers. The hidden layers will then make decisions from the previous layer and weigh up how a stochastic change within itself detriments or improves the final output, and this is referred to as the process of learning.

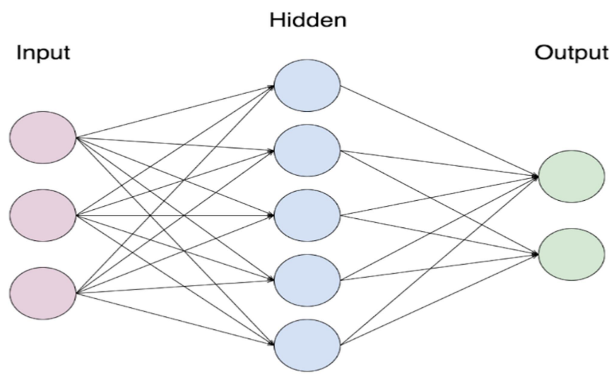


Fig. 10: Basic structure of ANN

2.2.2 Convolutional Neural Network – CNN

Convolution neural network [5] is designed and aimed towards identification of two-dimensional image information. The system consists of an input layer, convolutional layers, a sample layers (pooling) and an output layer. Furthermore, the layer architecture may be duplicated and be built one after another in order to create a module to the user’s needs. The input to the Convolution layer is two-dimension matrix. The convolution uses different filters to extract various features. The filter is a smaller matrix which moves through the image, multiply each pixel value with the filter on it and summarize it as shown in Figure 11. The result is smaller image with the new weight. In the polling level, new matrix is built with values from every filtered cell. The process is repeated multiple times depending on the user's definition. When we view an image, we view it from left to right or top to bottom to understand the different features of the image. Our brain combines different local features that we scanned to classify the image. This is exactly how CNN works.

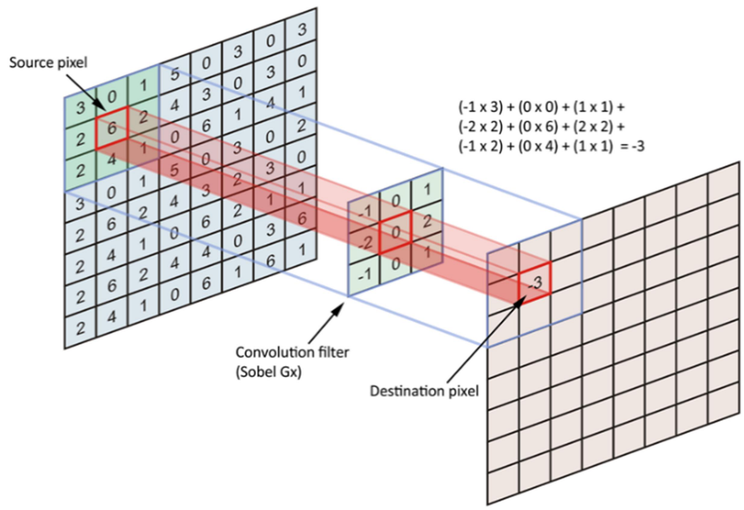


Fig. 11: An example of the convolution process using a filter on the input matrix

2.2.3 Forward

As the name suggests, the input data is fed in the forward direction through the network. Each hidden layer accepts the input data, processes it as per the activation function and passes to the successive layer.

In order to generate some output, the input data should be fed in the forward direction only. The data should not flow in reverse direction during output generation otherwise it would form a cycle and the output could never be generated [6].

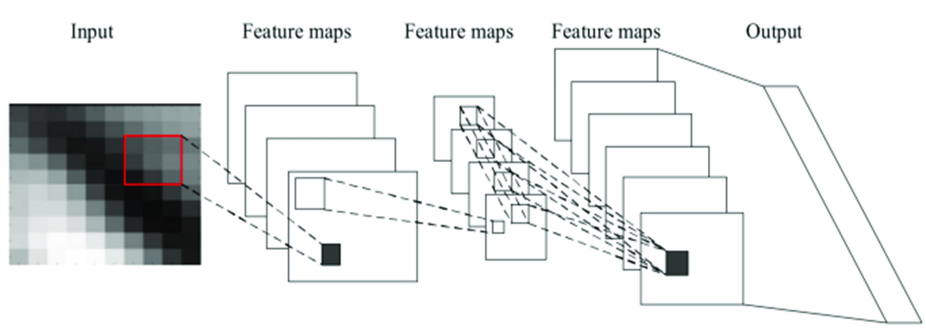
2.2.4 Back propagation

Back propagation is the opposite of Forward as it feeds the network backwards. BP network can be used to learn and store a great deal of mapping relations of input-output model, and no need to disclose in advance the mathematical equation that describes these mapping relations. Its learning rule is to adopt the steepest descent method in which the back propagation is used to regulate the weight value and threshold value of the network to achieve the minimum error sum of square.

Back-propagation is the essence of neural net training. It is the practice of fine-tuning the weights of a neural net based on the error rate obtained in the previous iteration. Proper tuning of the weights ensures lower error rates, making the model reliable by increasing its generalization.

2.2.5 CNN architecture

There are four basic types of layers used to build CNN architectures [7]: input layer, convolutional layer, pooling layer, and fully connected layer. Normally, a full CNN architecture is obtained by stacking several of these layers. An example of typical CNN architecture with two feature stages is



shown in Figure 12.

The convolutional neural network layers are [5]:

1. Input layer

The input layer is the layer that loads input and produces output used to feed convolutional layers. Some transformations such as mean-subtraction and feature-scaling can be applied [7].

2. Convolutional layer

The Convolutional layer is the core building block of CNN. When the data hits a convolutional layer, the layer convolves each filter across the spatial dimensionality of the input to produce a 2D activation map.

3. Pooling layer

These layers are responsible for down sampling the spatial dimension of the input. There is one pooling layer after each convolutional layer. All of them are set to use a 2×2 receptive field (spatial extent) with a stride of 2. The first pooling layer uses the most common max operation over the receptive field and the other two perform average pooling (Figure 13).

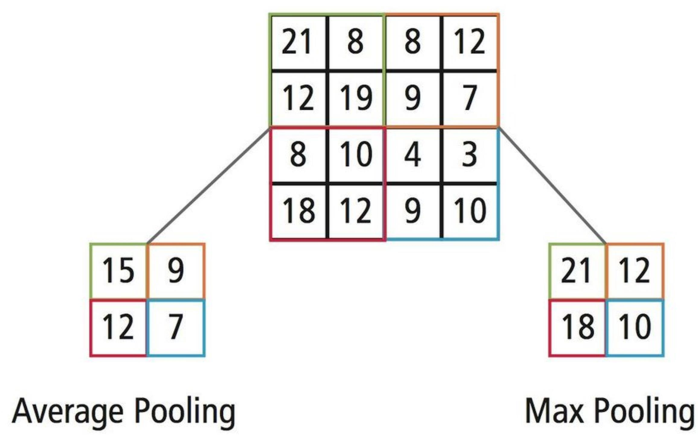
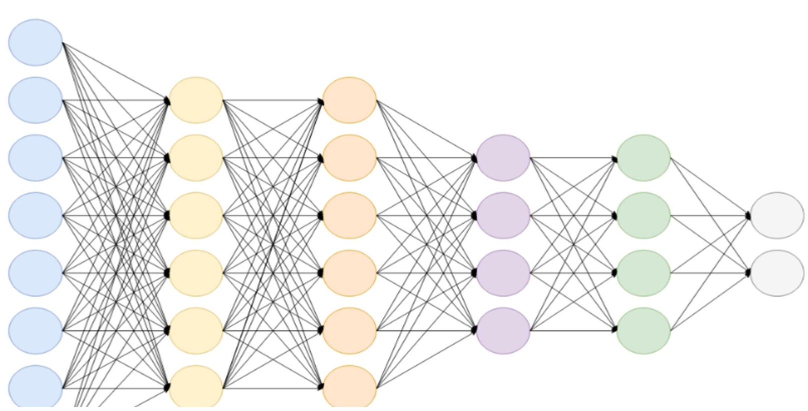


Fig. 13: Max and Average pooling examples

4. Fully connected layer

Fully connected layers [8] are an essential component of convolutional neural networks (CNNs), which have been proven very successful in recognizing and classifying images for computer vision. The CNN process begins with convolution and pooling, breaking down the image into features, and analyzing them independently. The result of this process feeds into a fully connected neural network



structure that drives the final classification decision. The fully connected layer is connected to all previous activation's layers as shown in Figure 14.

2.3 YOLO Network

YOLO (You Only Look Once) [9], is a network for object detection. The object detection task consists in determining the location on the image where certain objects are present, as well as classifying those objects. Previous methods for object detection, like R-CNN, perform this task in multiple steps. This can be slow to run because each individual component must be trained separately. By reframe the object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities, YOLO does it all with a single neural network. In this work we focused on the third version of YOLO (YOLOv3), when the project started, it was the newest version.

2.3.1 YOLO operations

The system divides the input image into an $S\,X\,S$ grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. The previous versions of YOLO had difficulty in identifying small objects. In order to solve that problem, YOLOv3 makes prediction across three different grid cell scales for each image. The different scales will be calculated by two parameters, input size and stride. Input size means the height and width of an image in pixels. Stride is the number of pixels that the filters shifts over the input matrix. When the stride equal to one, we move the filters one pixel at a time and so on. For YOLOv3 [10], the strides were defined to the sizes of 32, 16 and 8. The scales will be calculated as the input size divided by the stride. For example, for input size of 416 x 416 the three scales of detection will be calculated as: $\frac{416}{32} = 13, \frac{416}{16} = 26, \frac{416}{8} =$

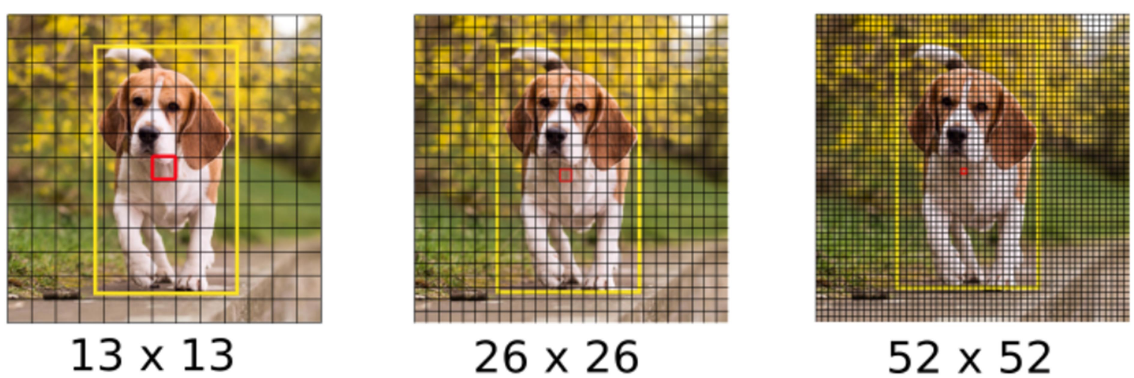
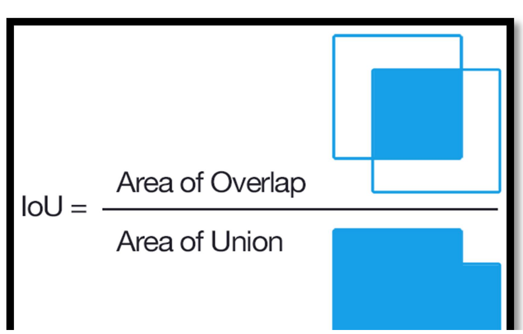


Fig. 15: Example to the different scales in YOLO

52. The result is 13x13, 26x26 and 52x52 as shown in Figure 15.

Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence



scores reflect how confident the model is that the box contains an object and how accurate it thinks the box is that it predicts. The confidence scores define as the probability that the bounding box contains an object multiply by the IOU value ($\Pr(Object) * IOU$). IOU means intersection over union. Intersection over union is an evaluation metric used to measure the accuracy of an object detector. The IOU value calculating by division of the overlapping area, between the predicted bounding box and the ground-truth bounding box, with the area of the union of those boxes as shown Figure 16. If no object exists in that cell, the confidence scores should be zero.

Each bounding box consists of 5 predictions: p_c, b_x, b_y, b_h, b_w . The value of p_c is the confidence prediction represents the IOU between the predicted box and any ground truth box. The values of b_x and b_y represent the center of the object, relative to the grid cell location with value between zero to one. The values of b_h and b_w are the width and height of an object that predicted relative to the whole image.

YOLOv3 predicts bounding boxes using dimension clusters as anchor boxes [10]. Anchor boxes are a set of predefined bounding boxes of a certain height and width. These boxes are defined to capture the scale and aspect ratio of specific object classes. Anchor boxes typically chosen based on object sizes in the training datasets and they allow the system to detect multiple objects, objects of different scales and overlapping objects. By pick better priors for the network to start with, it easier for the network to learn to predict good detections. Instead of choosing priors by hand as in YOLOv1, YOLOv3 runs k-means clustering on the training set bounding boxes to automatically find good priors.

The network predicts 4 location coordinates for each bounding box, t_x, t_y, t_w, t_h, relative to the location of the grid cell. The values of (t_x, t_y) represent the center of the predicted bounding box, the values of (t_w, t_h) represent the width and height. The following formula describes how the network output is transformed to obtain bounding box predictions:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w * e^{t_w}$$

$$b_h = p_h * e^{t_h}$$

(c_x, c_y) are the top-left coordinates of the grid cell, (p_w, p_h) are the anchors dimensions for the box. To bounds the ground truth to fall between 0 to 1, the furmula use a sigmoid function (σ) to constrain the network's predictions to fall in this range. An example of calculating the values is shown

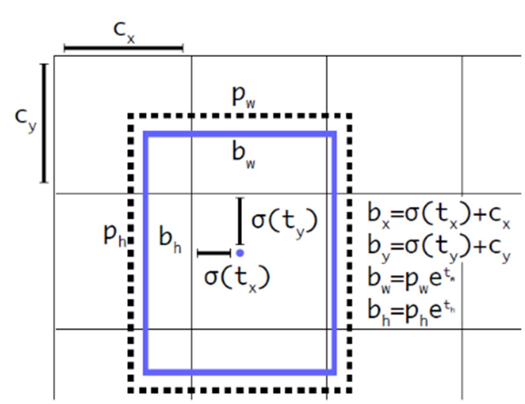


Fig. 17: Calculating the values for the anchor boxes

in Figure 17.

For each bounding box, the system also predicts C conditional class probabilities, $\Pr(Class_i|Object) * IOU$. These probabilities mean that if that cell contains an object, that object belongs to $Class_i$. In YOLOv3 the system trained to recognized 80 different classes ($C = 80$). YOLOv3 tensor will be as follows: $[(SxS)x[3 * (4 + 1 + 80)]$. YOLO predict 3 boxes pre cell at each scale, 4 bounding box offsets, 1 confidence prediction and 80 class predictions.

YOLO predicts 3 bounding boxes for each grid cell in each scale, for an image of size 416 x 416, YOLO predicts (52 x 52 x 3) + (26 x 26 x 3) + (13 x 13 x 3) = 10647 bounding boxes. In order to filter irrelevant bounding boxes, the system uses object confidence score. The object confidence score calculates for each bounding box by multiplying the confidence scores (p_c) with the class probabilities (C). The object confidence score represents how much the system confident in the prediction that the object belongs to that class. Boxes having scores below a threshold of 0.5 are ignored. The next step will be to use non-maximum suppression (NMS) to eliminate multiple detection of the same image. NMS will pick the bounding box with the largest confidence scores (p_c), it will check the IOU value between that bounding box to all other bounding boxes and discard any remaining box with IOU greater than or equal to 0.5.

2.3.2 Network design

YOLOv3 network [11] makes use of only convolutional layers, making it a fully convolutional network. It uses a variant of Darknet, which originally has 53 convolutional layers. Another 53 layers are stacked onto it, dedicated to resizing, concatenation residual skip connections and upsampling the input and to prepare it for detection. The upsampled layers concatenated with the previous layers help preserve the features, which help in detecting small objects. The total number of layers in the network is 106 layers. The full network architecture shown in Figure 18.

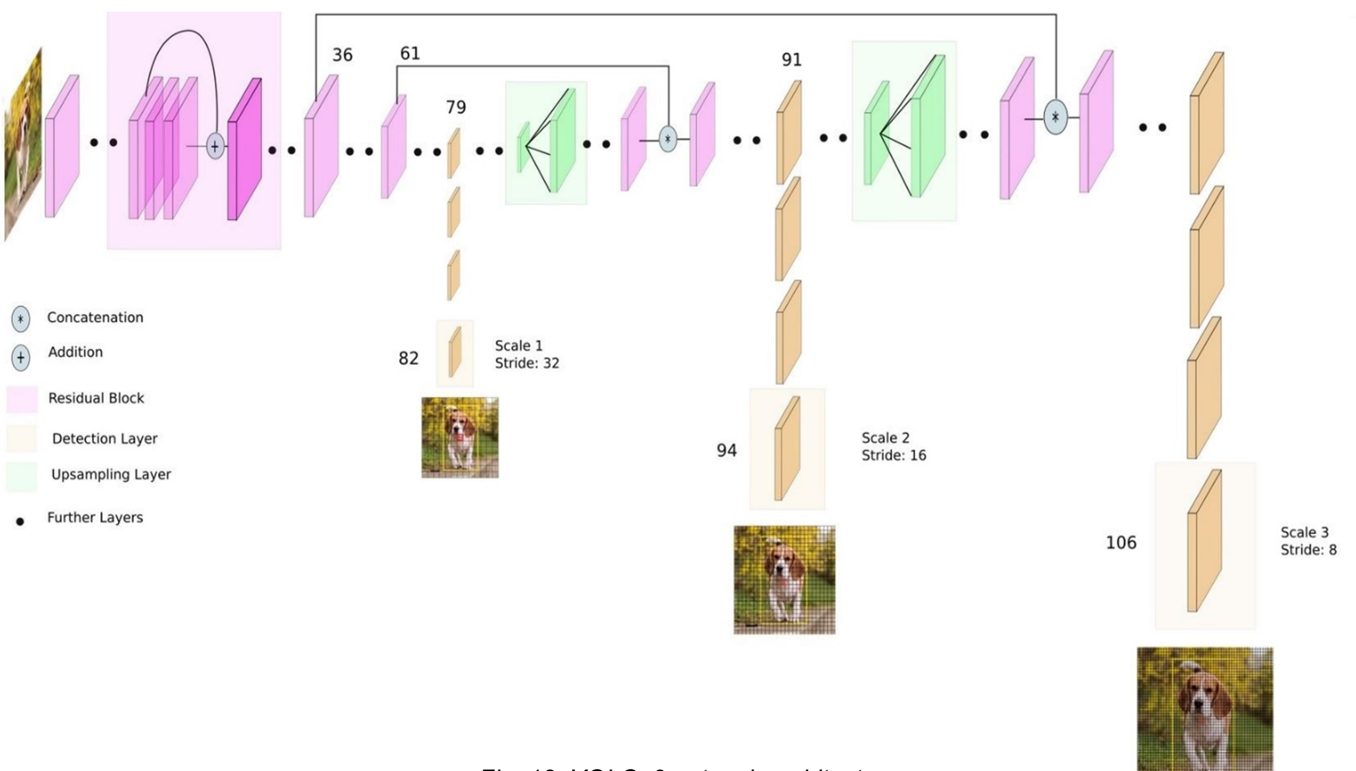


Fig. 18: YOLOv3 network architecture

2.3.3 Loss function

During training the network optimize the bounding boxes using localization loss function which measures the errors in the predicted boundary box locations and sizes. The loss function is:

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]$$

$$+\ \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\widehat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\widehat{h}_i}\right)^2\right]$$

Where $(x_i\,, y_i\,, w_i, h_i)$ are the ground truth coordinates and $(\ \hat{x}_i\,,\ \hat{y}_i\,, \widehat{w}_i\,, \widehat{h}_i)$ are the predicted coordinates. λ_{coord} increase the weight for the loss in the boundary box coordinates, the default value is $\lambda = 5$. 1_{ij}^{obj} equals 1 if the j boundary box in cell i is responsible for detecting the object, otherwise is 0. S represent the number of grid cells and B represent the number of boundary boxes.

To optimize the class prediction, during training the network uses binary cross-entropy loss as shown in the formula:

$$H = -\frac{1}{n}\sum_{i=1}^{n} y_i * \log\big(p(y_i)\big) + (1-y_i) * \log(1-p(y_i))$$

Where y_i is the label of class i ($i = 1$ if the class is in predicted image, otherwise $i = 0$), $p(y_i)$ is the predicted probability for that class.

2.4 Rotation Transformation

Rotation is a transformation [12] in which the object is rotated around a fixed point. The direction of rotation can be clockwise or counterclockwise. The fixed point in which the rotation take place is called the center of rotation. The amount of rotation made is called the angle of rotation Θ, we calculated it by using the tangent function. For any rotation, we need to specify the center, the angle and the direction of rotation. We used a rotation matrix. In linear algebra, a rotation matrix is a matrix that is used to perform a rotation in Euclidean space.

Rotation algorithm:

Input: Array of coordinates.

Step 1: Find the Direction of rotation: examine the first (x_1, y_1) and last (x_2, y_2) coordinates of the array, if y_1 is bigger then y_2 rotation direction is set to counterclockwise and the rotation matrix will be

$R(\theta) = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$, in any other case it will be set to clockwise and the rotation matrix will be

$R(-\theta) = \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}$.

Step 2: Find the rotation angle: calculate the angle Θ as $\tan(\frac{|x_2|}{|y_2|})$.

Step 3: Rotate the array: each coordinate in the array is multiplied by the rotation matrix, in order to multiply the coordinate with a matrix it will be written as vector $\boldsymbol{v}$, for example the counterclockwise rotation of the coordinate (x,y) will be $\boldsymbol{R} * \boldsymbol{v} = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}\mathrm{x}\cos\theta + y(-\sin\theta)\\ x\sin\theta + y\cos\theta\end{bmatrix}$.

Output: Array of rotated coordinates.

2.5 Inputs and Operation of Our System

In our project we used YOLOv3 network to detect LPR symptoms from endoscopic images. The images were received from the 250 videos of endoscopic examinations performed on LPR patients in Nahariya Hospital. The images were extracted from the videos. In addition, to increasing the number of images we used data augmentation [13]. Data augmentation is a strategy that enables to significantly increase the diversity of data available for training models, without collecting new data. We used data augmentation techniques such as cropping, and horizontal flipping.

The inputs for our system are two images. First, the image from the endoscopic test (as shown in Fig. 19), and second, image that received after a mask was used on the endoscopic image to color the larynx in white and the rest of the image in black (as shown in Fig. 20). The matched mask image will be received from a different project. The mask had helped us to find the coordinates that represent the relevant area where each symptom is located. Each area was cut from the original image according to

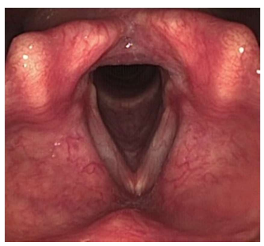


Fig. 19: the original image that was received from the endoscopic test.

Fig.20: masked version of the original image.

the coordinates and then was sent to YOLO network for detection.

Some symptoms have several degrees of severity and therefore several scores are possible. Because the lack of images for some of the symptoms, and since there is no complete agreement between diagnoses of different doctors regarding to the severity of the symptoms, we trained our networks to determine only whether the symptom exists or not. If symptom was found, we gave him score of 2, based on the doctor average results. If the symptom is not found his score set to 0. The system flow chart shown in Figure 21.

2.6 Algorithms for Symptoms Areas

To perform the diagnosis, our system gets an image of the larynx area, and cut only the area that is relevant to each symptom. The following section will describe and demonstrate the algorithms we used to find each symptom.

During the explanation we will use the term “triangle”. The triangle is the form created by the vocal folds. The triangle marked on the larynx shown in Figure 22. We will also be referred to the upper side of the triangle as the upper boundary. The input to the algorithms is larynx image and array of three coordinates for the triangle.

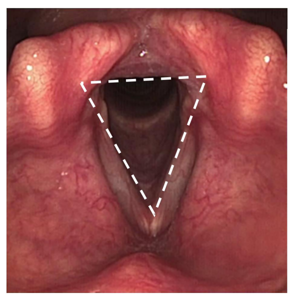


Fig. 22: The triangle created by the vocal fold. The uppe... the upper boundary.

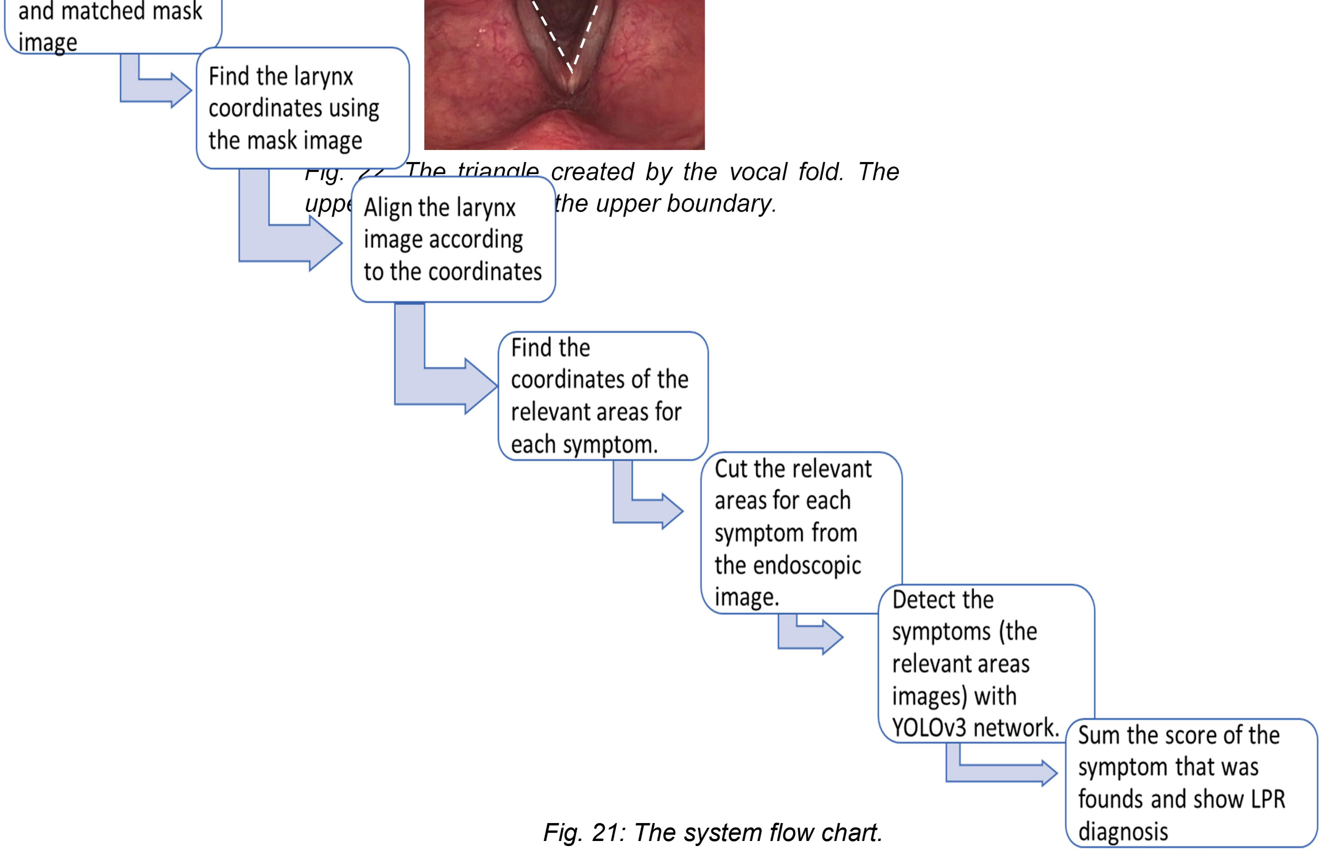


Fig. 21: The system flow chart.

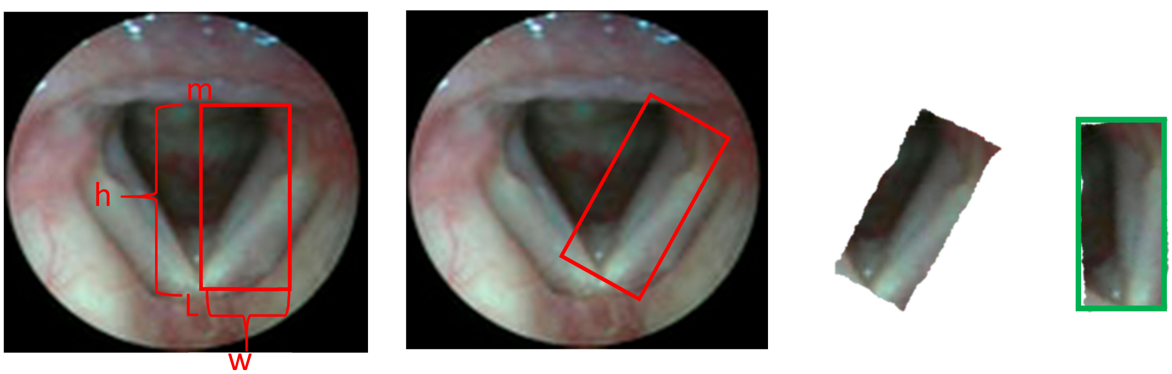
2.6.1 Relevant area selection for subglottic edema

The subglottic edema located in the inner side of the vocal folds. The edema can be found on the inner side of each one of the vocal folds. Therefore, from each endoscopic image we cut each side of the vocal fold separately. We defined each cut image to be independent representation of that symptom. If the symptom was detected on one side, the system determined that the symptom is present, even if the symptom does not detect in the other side. Selecting the symptom from the right side of the vocal fold is described in the following algorithm:

Input: Larynx image.jpg, Triangle vertexes coordinate
Step 1: Calculate the midpoint of the triangle upper boundary vertexes.
Step 2: Calculate the height between the midpoint to the lower vertex of the triangle, marked as ‘h’.
Step 3: Set the rectangle height, height = ‘h’.
Step 4: Calculate the width between midpoint to the upper right vertex of the triangle, marked as ‘w’.
Step 5: Set the rectangle width, width = ‘w’.
Step 6: Set the vertexes coordinates of the rectangle.

  Step 6.1: Set the rectangle top left vertex to be equal to the midpoint of the triangle upper boundary vertexes**.**
  Step 6.2: Set the rectangle bottom left vertex to be equal to the triangle lower vertex.
  Step 6.3: Calculate the values of top right vertex and the bottom right vertex according to the height and width values and respectively to the left vertices of the rectangle.

Step 7: Rotate the rectangle to the angle of the vocal fold by using the rotation algorithm described in section 2.4.
Step 8: Cropped the Larynx image.jpg according the rectangle coordinates received after the rotation.
Step 9: Align the cropped image using the rotation algorithm described in section 2.4.
Output: Image contain the relevant area for subglottic edema.



(a) The rectangle applied to the image. the midpoint marked with the letter ‘m’, the lower vertex marked with the letter ‘L’, the height marked with the letter ‘h’ and the width with the letter ‘w’.

(b) The rectangle applied to the image after the rotation.

(c) The image cropped according to the rectangle coordinates.

(d) Align the cropped image.

Fig. 23: Relevant area selection for subglottic edema – right vocal fold.

2.6.2 Relevant area selection for ventricular obliteration

Like the subglottic edema symptom, the ventricular obliteration located from both side of the vocal folds but located on their outer side. For this symptom we also cut each side of the vocal fold separately and defined it to be independent representation of that symptom. If the symptom was detected on one side, the system determined that the symptom is present. Selecting the symptom from the right side of the vocal fold is described in the following algorithm:

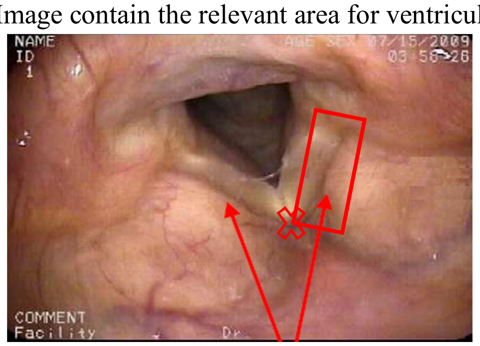
Input: Larynx image.jpg, Triangle vertexes coordinate
Step 1: Calculate the midpoint of the triangle upper boundary vertexes.
Step 2: Calculate the height between the midpoint to the lower vertex of the triangle, marked as ‘h’.
Step 3: Set the rectangle height, height = ‘h’.
Step 4: Calculate the width between midpoint to the upper right vertex of the triangle, marked as ‘w’.
Step 5: Set the rectangle width, width = ‘w’.
Step 6: Set the rectangle to the coordinates

> **Step 6.1:** Set the rectangle top left vertex to be equal to the midpoint of the triangle upper boundary vertexes**.**
> **Step 6.2:** Set the rectangle bottom left vertex to be equal to the triangle lower vertex.
> **Step 6.3:** Calculate the values of top right vertex and the bottom right vertex according to the height and width values and respectively to the left vertices of the rectangle.

Step 7: Rotate the rectangle to the angle of the vocal fold by using the rotation algorithm described in section 2.5.
Step 8: Find the rectangle bottom left vertex position:

> **Step 8.1:** Start at the triangle lower vertex and move down by twenty pixels.

Step 9: Calculate the offset between the rectangle bottom left coordinate to the coordinate found in step 8.1.
Step 10: Calculate the rectangle coordinates according to the offset, so that the bottom left vertex of the rectangle will be equal to the coordinate found in step 8.1.
Step 8: Cropped the Larynx image.jpg according the rectangle coordinates.
Step 9: Align the cropped image using the rotation algorithm described in section 2.4.
Output: Image contain the relevant area for ventricular obliteration.



(a) The red rectangle received from the algorithm located around the symptom. The rectangle bottom left vertex position, that have been found using the algorithm in step 8, marked with red X.

(b) The final image produces by the algorithm. The red arrow marks the symptom.

Fig. 24: Relevant area selection for ventricular obliteration.

2.6.3 Relevant area selection for erythema

The erythema uppers on the arytenoids in the upper side of the larynx. If the symptom was detected in one side of the larynx, the system determined that the symptom is present. To cut the area we used the algorithm described below:

Input: Larynx image.jpg, Triangle vertexes coordinate
Step 1: Find the rectangle top left vertex:

> **Step 1.1:** Start at the triangle upper right coordinate and move up by forty pixels.
> **Step 1.2:** Set the current location as the rectangle top left vertex.

Step 2: Find the rectangle bottom left vertex:

Step 2.1: Start at the triangle upper right coordinate and move down by forty pixels.
Step 2.2: Set the current location as the rectangle bottom left vertex.

Step 3: Find the rectangle top right vertex:
Step 3.1: Start at the rectangle upper left coordinate and move right by fifty pixels.
Step 3.2: Set the current location as the rectangle top right vertex.

Step 4: Find the rectangle bottom right vertex:
Step 4.1: Start at the rectangle upper left coordinate and move right by fifty pixels.
Step 4.2: Set the current location as the rectangle bottom right vertex.

Step 5: Cropped the Larynx image.jpg according the rectangle coordinates.

Output: Image that contain the right arytenoids area of the symptom.

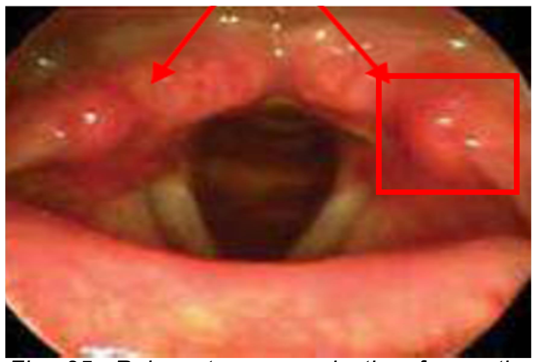


Fig. 25: Relevant areas selection for erythema. After finding the relevant areas, the red rectangle will be the area that selected for the arytenoids.

2.6.4 Relevant area selection for vocal fold edema

For the vocal fold edema, we also cut each side separately. We defined each cut image to be independent representation of that symptom. Selecting the symptom right side vocal fold is described in the following algorithm:

Input: Larynx image.jpg, Triangle vertexes coordinate

Step 1: Find the rectangle top left vertex:

Step 1.1: Calculate the midpoint of the triangle upper boundary vertexes.

Step 1.2: Set the rectangle top left vertex to be equal to the midpoint.

Step 2: Find the rectangle bottom left vertex:
Step 2.1: Start at the triangle lower vertex coordinate and move down by twenty pixels.
Step 2.2: Set the current location as the rectangle bottom left vertex.

Step 3: Set the rectangle top right vertex to be equal to the triangle upper right coordinate.

Step 4: Set the rectangle bottom right vertex corresponding to the other vertices.

Step 5: Rotate the rectangle to the angle of the vocal fold by using the rotation algorithm described in section 2.4.

Step 6: Cropped the Larynx image.jpg according the rectangle coordinates that have been received.

Step 7: Align the cropped image using the rotation algorithm described in section 2.4.

Output: Image contain the right side of the vocal fold edema area.

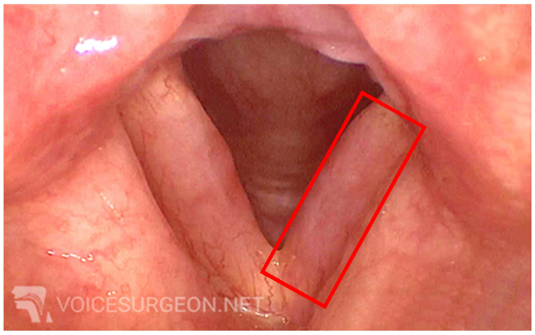


Fig. 26: Relevant area selection for vocal fold edema. After following the algorithm, the red rectangle located around the symptom area. In training the class labels will contain the whole image of the relevant area (same area as the red rectangle)

2.6.5 Relevant area selection for diffuse laryngeal edema

The symptom refers to the relative ratio of the endolaryngeal airway to the whole larynx. Therefore, we defined the relevant area for the symptom to contain the entire triangle of the larynx area. By training the network on the all larynx area and comparing it to healthy larynx, the network identifies the ratio of larynx with the symptom and without him. The selection of the larynx area described in the following algorithm:

Input: Larynx image.jpg, Triangle vertexes coordinate
Step 1: Find the rectangle top right vertex:
- **Step 1.1:** Start at the triangle upper right coordinate and move up by two pixels.
- **Step 1.2:** Move right by four pixels.
- **Step 1.3:** Set the current location as the rectangle top right vertex.

Step 2: Find the rectangle top left vertex:
- **Step 2.1:** Start at the triangle upper left coordinate and move up by two pixels.
- **Step 2.2:** Move left by four pixels.
- **Step 2.3:** Set the current location as the rectangle top left vertex.

Step 3: Set the value of the rectangle bottom left coordinate:
- **Step 3.1:** Set the value of the vertical axis component to be equal to value of the vertical axis component of the triangle lower vertex.
- **Step 3.2:** Set the value of the horizontal axis component to be equal to the value of the horizontal axis component of the rectangle top left coordinate.

Step 4: Set the value of the rectangle bottom right coordinate:
- **Step 4.1:** Set the value of the vertical axis component to be equal to value of the vertical axis component of the triangle lower vertex.
- **Step 4.2:** Set the value of the horizontal axis component to be equal to the value of the horizontal axis component of the rectangle top right coordinate.

Step 5: Cropped the Larynx image.jpg according the rectangle coordinates.
Output: Image contain the larynx area.

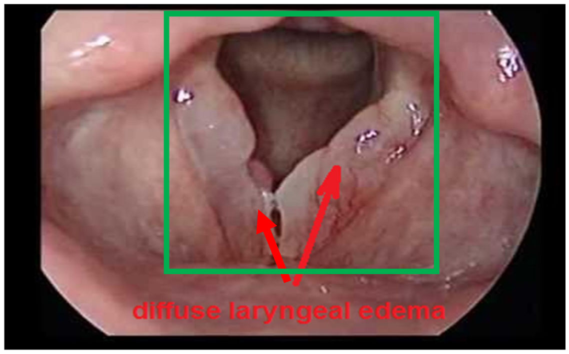


Fig. 27: Relevant area selection for diffuse laryngeal edema. After following the algorithm green rectangle located around the larynx area. the edema symptom marked with arrows. the class labels will be the same area as the green rectangle.

2.6.6 Relevant area selection for posterior commissure hypertrophy

The posterior commissure hypertrophy located in the back side of the larynx.

Input: Larynx image.jpg, Triangle vertexes coordinate
Step 1: Find the rectangle bottom right vertex:
- **Step 1.1:** Start at the triangle upper right coordinate and move down by twenty pixels.
- **Step 1.2:** move left by ten pixels.
- **Step 1.3:** Set the current location as the rectangle bottom right vertex.

Step 2: Find the rectangle bottom left vertex:
- **Step 2.1:** Start at the triangle upper left coordinate and move down by twenty pixels.
- **Step 2.2:** move right by ten pixels.
- **Step 2.2:** Set the current location as the rectangle bottom left vertex.

Step 3: Find the rectangle top left vertex:
- **Step 2.1:** Start at the rectangle bottom left coordinate and move up by forty pixels.
- **Step 2.2:** Set the current location as the rectangle top left vertex.

Step 4: Find the rectangle top right vertex:
- **Step 4.1:** Start at the rectangle bottom right coordinate and move up by forty pixels.
- **Step 4.2:** Set the current location as the rectangle top right vertex.

Step 5: Cropped the Larynx image.jpg according the rectangle coordinates.
Output: Image contain the posterior commissure hypertrophy relevant area.

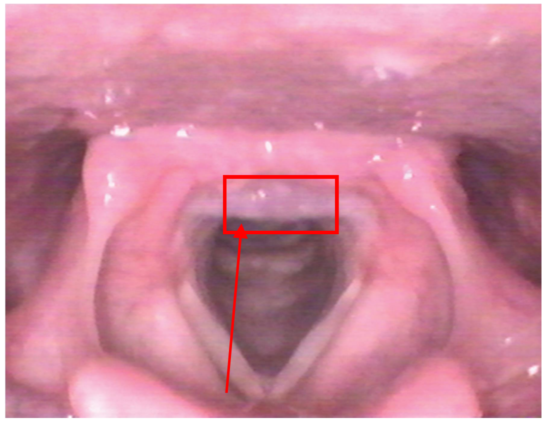
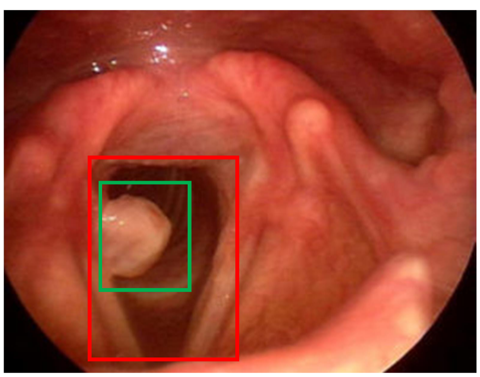


Fig. 28: Relevant area selection for posterior commissure hypertrophy. The red triangle around the symptom mark the relevant area. The class labels will contain the whole image of the relevant area (same area as the red rectangle).

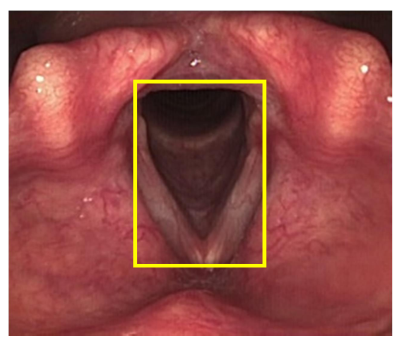
2.6.7 Relevant area selection for granuloma

The granuloma symptom usually appears on the inside of the triangle and can be found in each side of it. Therefore, we chose the coordinates for our mask to surround all the triangle area and contain minimal part as possible of the vocal folds. The relevant area selection was according the algorithm described in section 2.6.5 (the relevant area selection for diffuse laryngeal edema).

In training, after we cut the relevant area, we labeled only the symptom himself as $Class_2$. For $Class_1$, we cut the same relevant area from healthy image and labeled the whole larynx as $Class_1$.



(a) The red rectangle marks the relevant area. The green rectangle is $Class_2$ label.



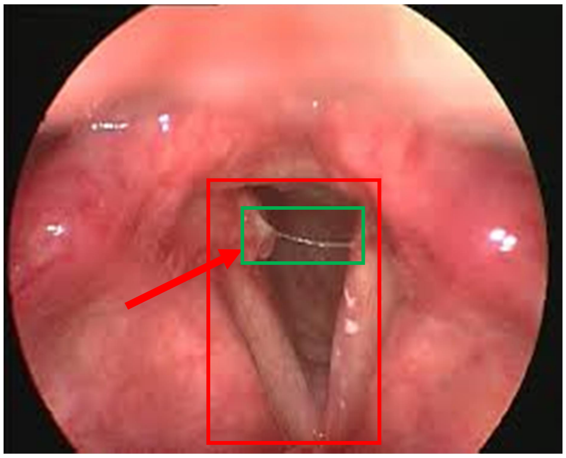
(b) $Class_1$ for granuloma training set will be the whole larynx area as marked with the yellow rectangle.

Fig. 29: Relevant area selection for granuloma.

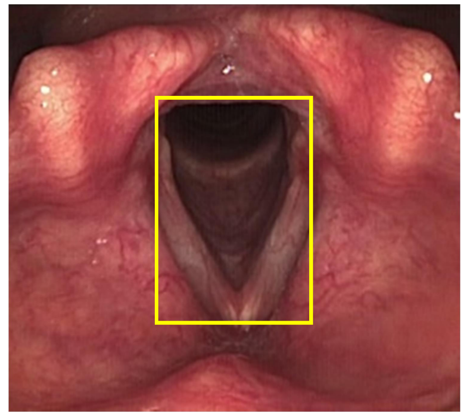
2.6.8 Relevant area selection for thick endolaryngeal mucus

The thick endolaryngeal mucus also appears on the inside of the triangle. It can be found anywhere within the boundaries of the triangle. Therefore, in this case we selected our area of interest to contain all the triangle area by using the algorithm described in section 2.6.5.

In training, after we cut the relevant area, we labeled only the symptom himself as $Class_{12}$. For $Class_1$, we cut the same relevant area from healthy image and labeled the whole larynx as $Class_1$.



(a) The red rectangle marks the relevant area. The green rectangle is $Class_2$ label.



(b) $Class_1$ for endolaryngeal mucus training set will be the whole larynx area as marked with the yellow rectangle.

Fig. 30: Relevant area selection for thick endolaryngeal mucus.

3. SOFTWARE ENGINEERING DOCUMENTS

3.1 Requirements (Use Case)

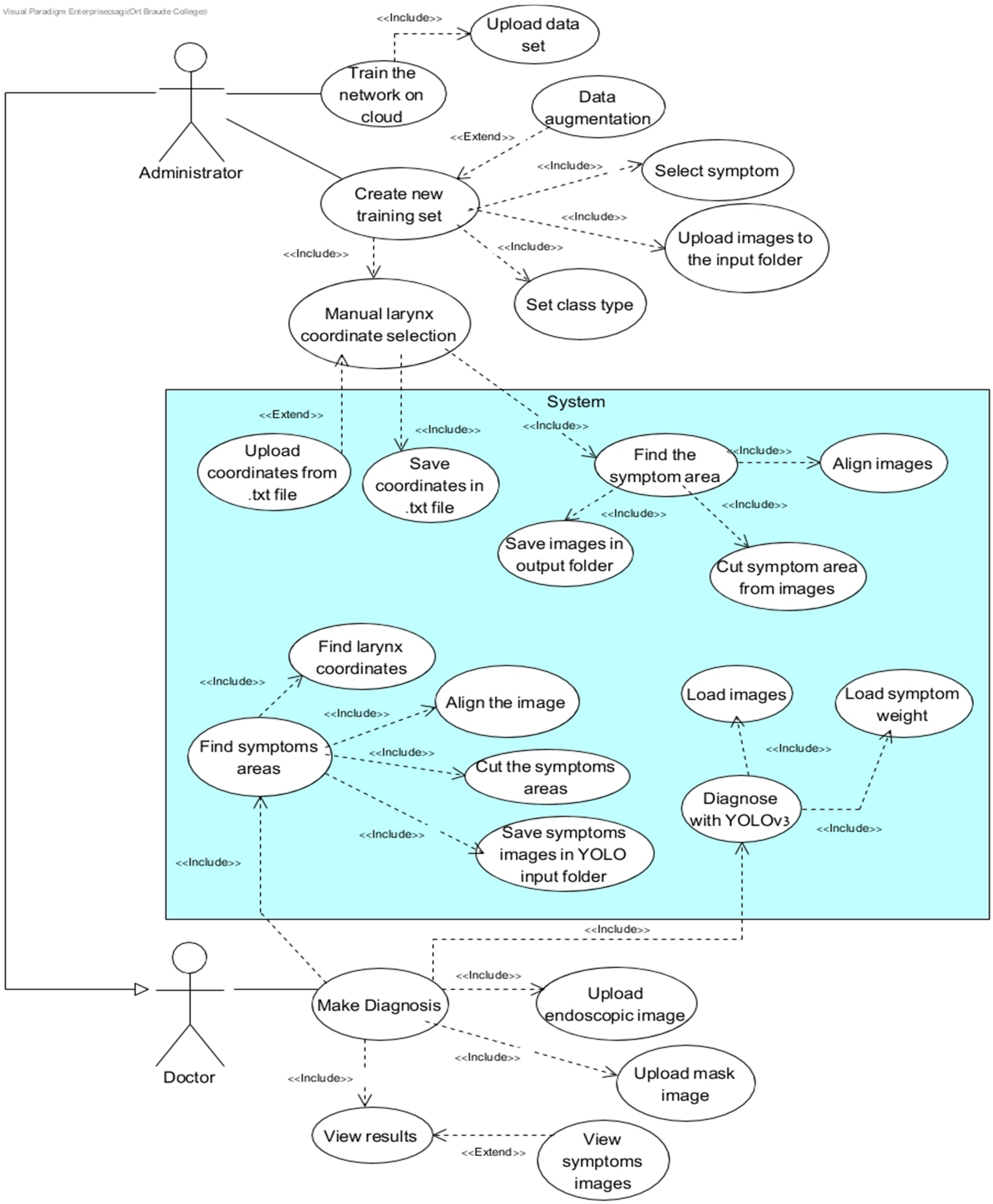


Fig. 31: Use case diagram.

3.2 Design (GUI, UML diagrams)

3.2.1 GUI

The GUI of the system allows the user (the doctor) to diagnose the endoscopic image in a simple and friendly way.

The first screen to open when the system is turned on is the welcome screen as shown in Figure 32. This screen provides a brief explanation to the purpose of the system. The doctor enters the system by pressing the "Continue" button.

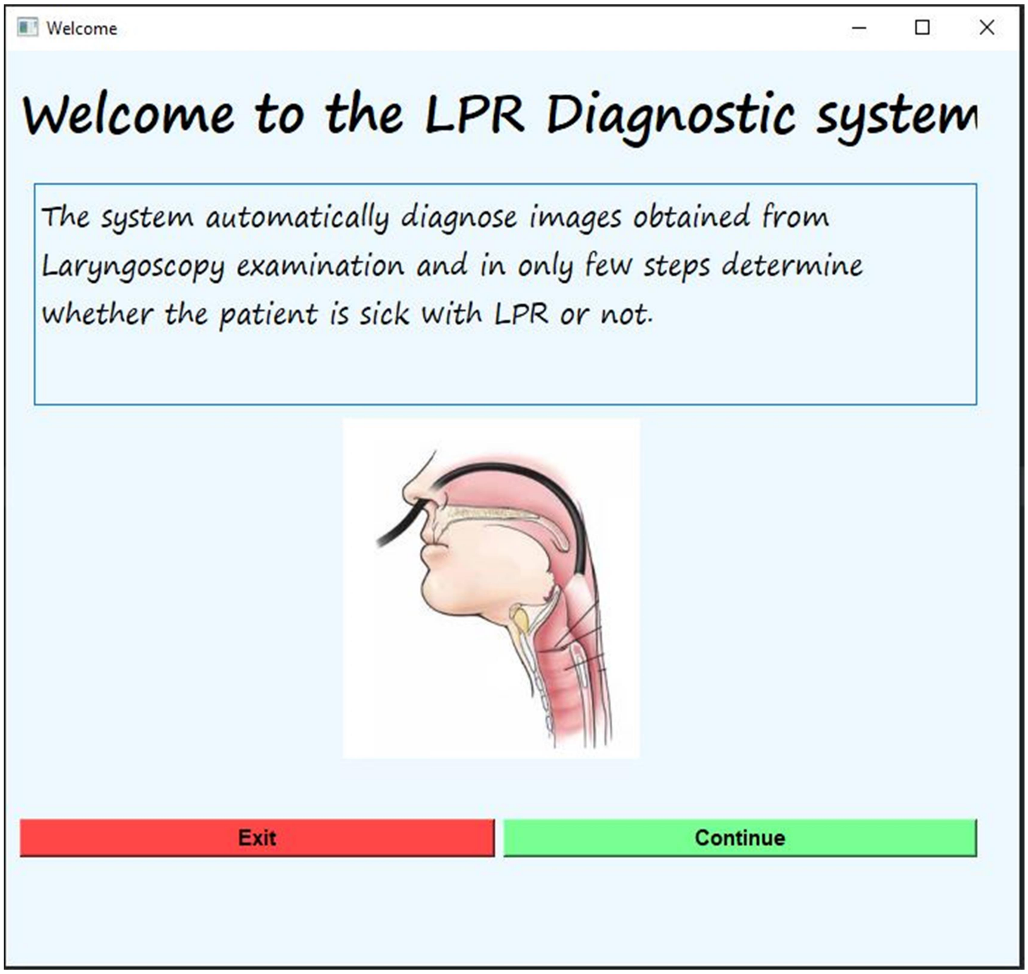


Fig. 32: welcome screen.

After clicking the "Continue" button the doctor will be moved to the "Make Diagnosis" screen as shown in Figure 33. If the user will click on the "Help" button, the system will provide information on the screen and how to use it as shown in Figure 34. The doctor will select the image and the corresponding mask image. The system will display an example to the images that the user will need to upload. To upload image, the user will press the "Upload Larynx Image" button, then the file explorer will be open, and the user will select the image. To upload mask image, the user will press the "Upload Larynx Mask" button and will select the mask using the file explorer.

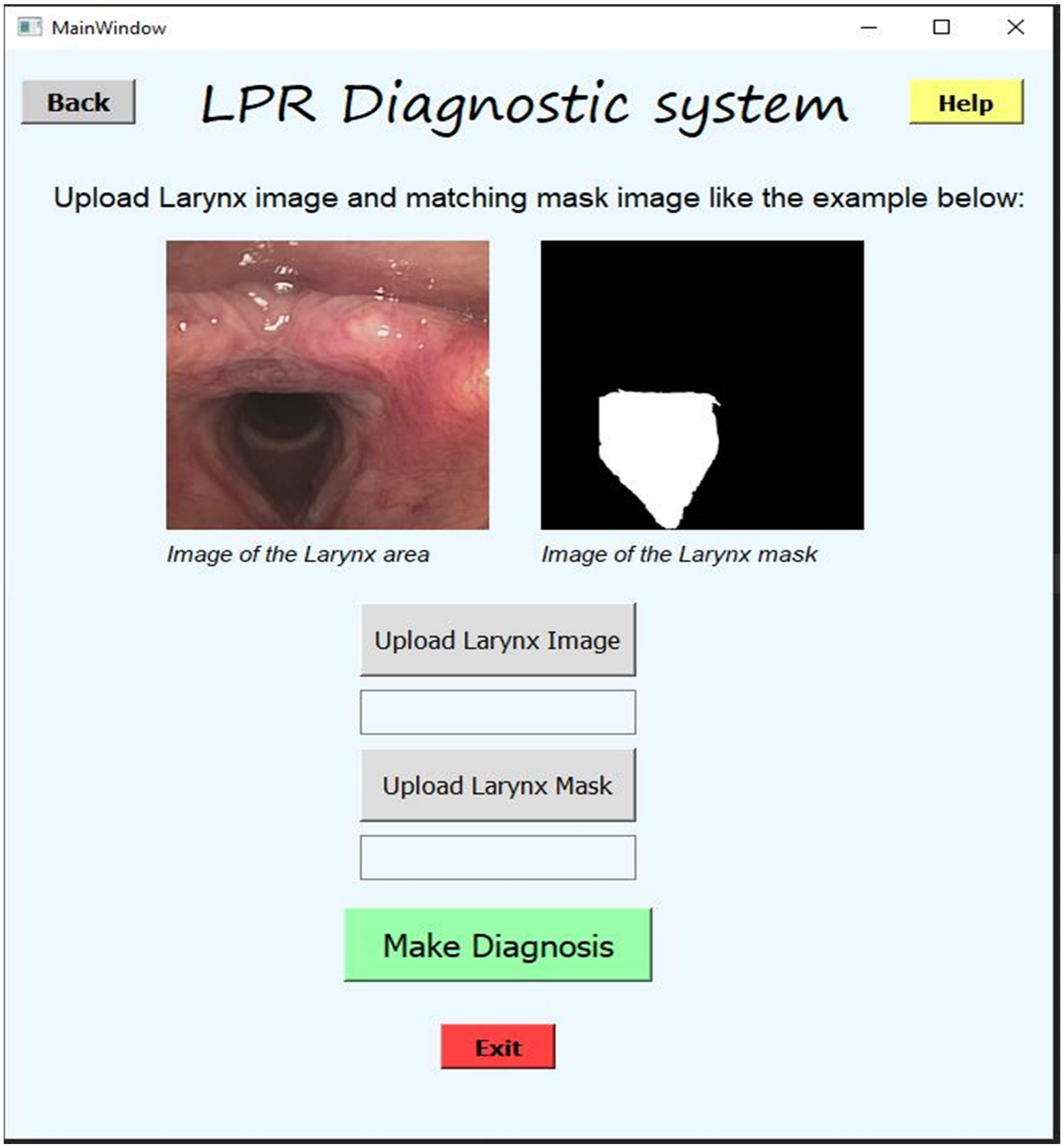


Fig. 33: make diagnosis (upload images) screen.

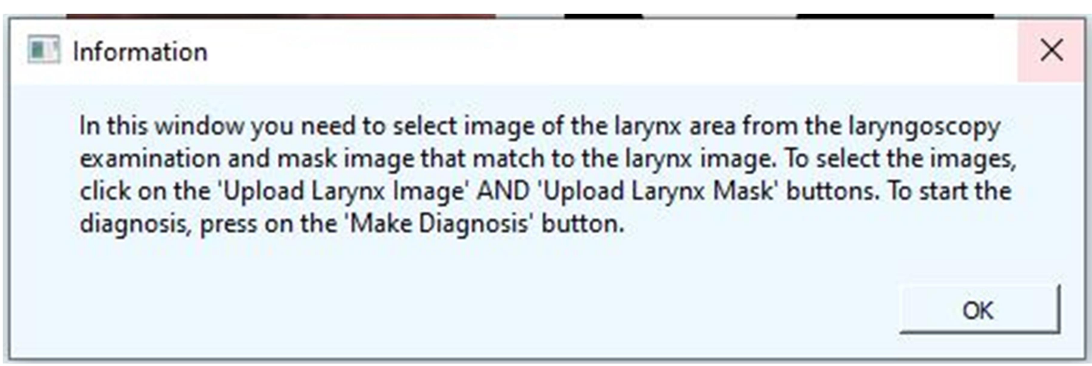


Fig. 34: information screen.

To reduce user errors, the system will only allow him to select png, jpg or jpeg files. If the user will try to press the "Make Diagnosis" button without upload image or mask, the system will show error message, as shown in Figure 35 and Figure 36.

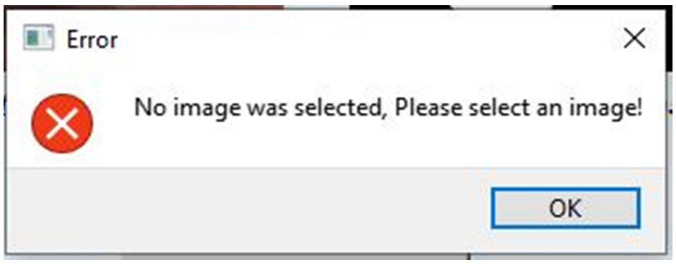


Fig. 35: no image selected error message.

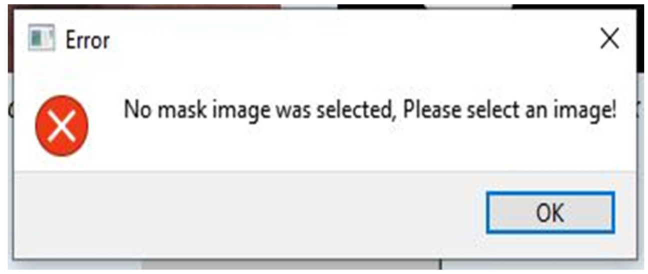


Fig. 36: no mask image selected error message.

Finally, after uploading the images and pressing the "Make Diagnosis" button, the "build model" screen will open as shown in Figure 37. The screen will explain to the user that the operation has begun, and it will take a few seconds. During this time, the system will detect and cut the area of each symptom and load the relevant weights for each symptom into the network. Once the process is complete, the "View Result" button will appear, as shown in Figure 38, and the user will be able to view the results.

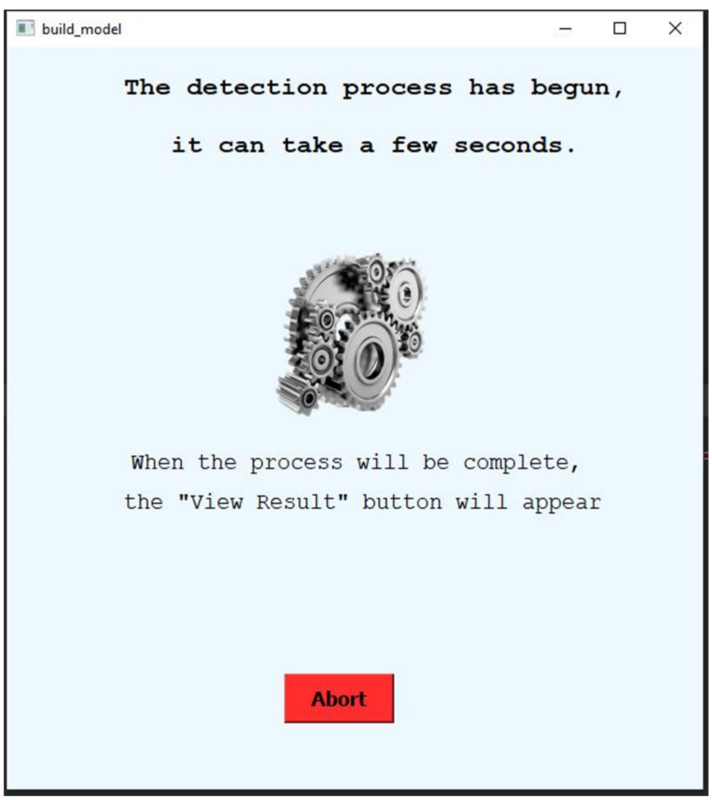


Fig. 37: detection process screen.

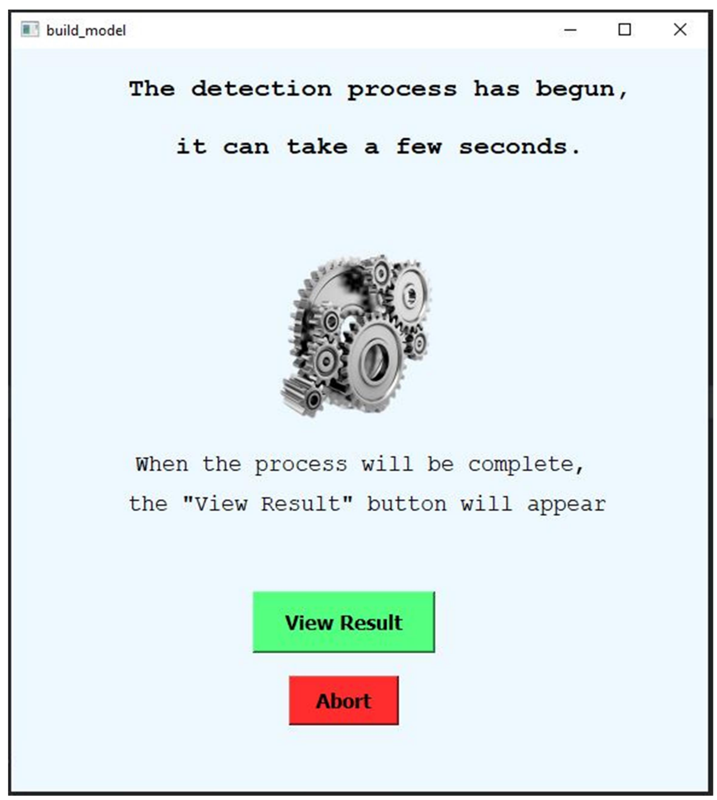


Fig. 38: view result button in the screen.

After clicking the “View Result” button, the results screen will be open as shown in Figure 39. The result screen shows to the user the score of every symptom. If the symptom received score of 2, it means that the system recognized that the symptom exists, if he received the score 0, the symptom was not found. The “Back” button will return the user to the “Make Diagnosis” screen. The “Help” button will open the information on the screen. To view the images of the symptoms the user will click on the

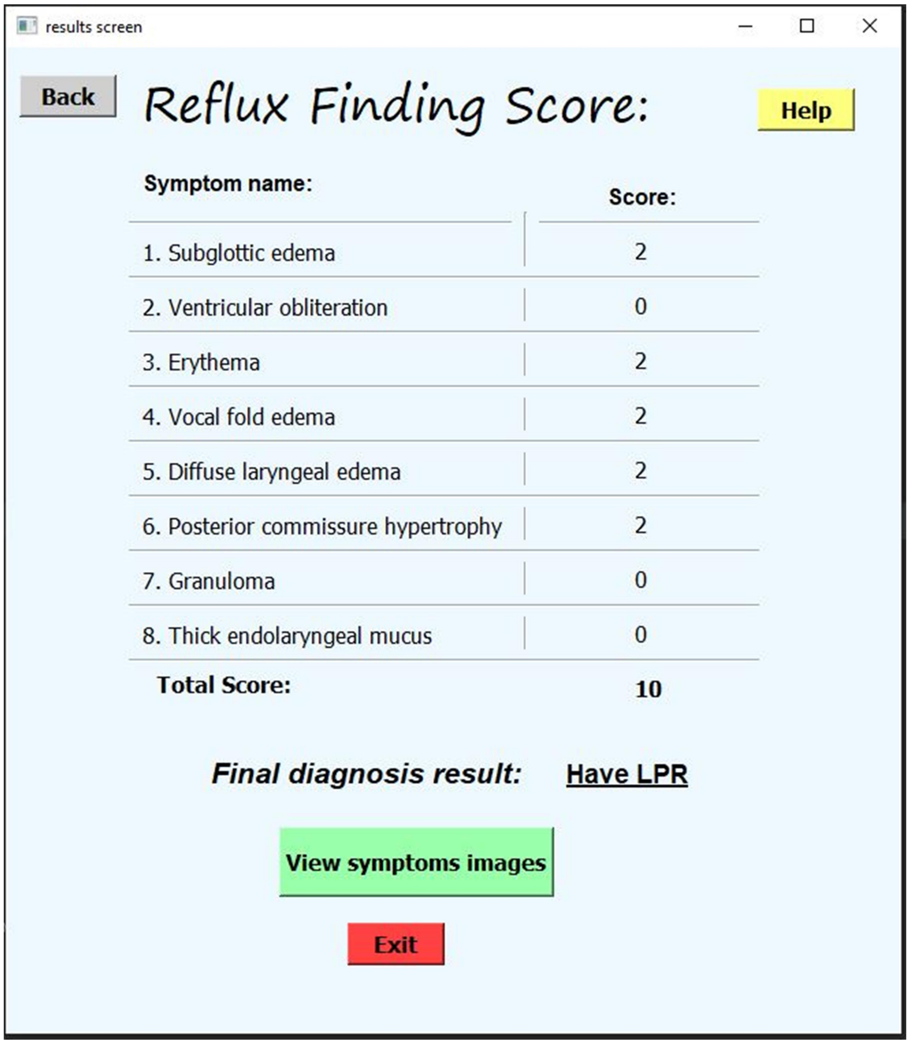


Fig. 39: result screen.

“View symptom images”.

In the “result images” screen the user will be able to view the images of the symptoms by region. In the top of the screen the system will show the name of the symptom, followed by the symptom image, the detection result, and the confidence score. If the system were unable to identify the area (it did not identify a healthy or sick area) at the bottom of the image the result label would say "Unknown" and the score would be -1. To switch between the symptom images, the user will click on the button with the symptom number from the bottom of the screen.

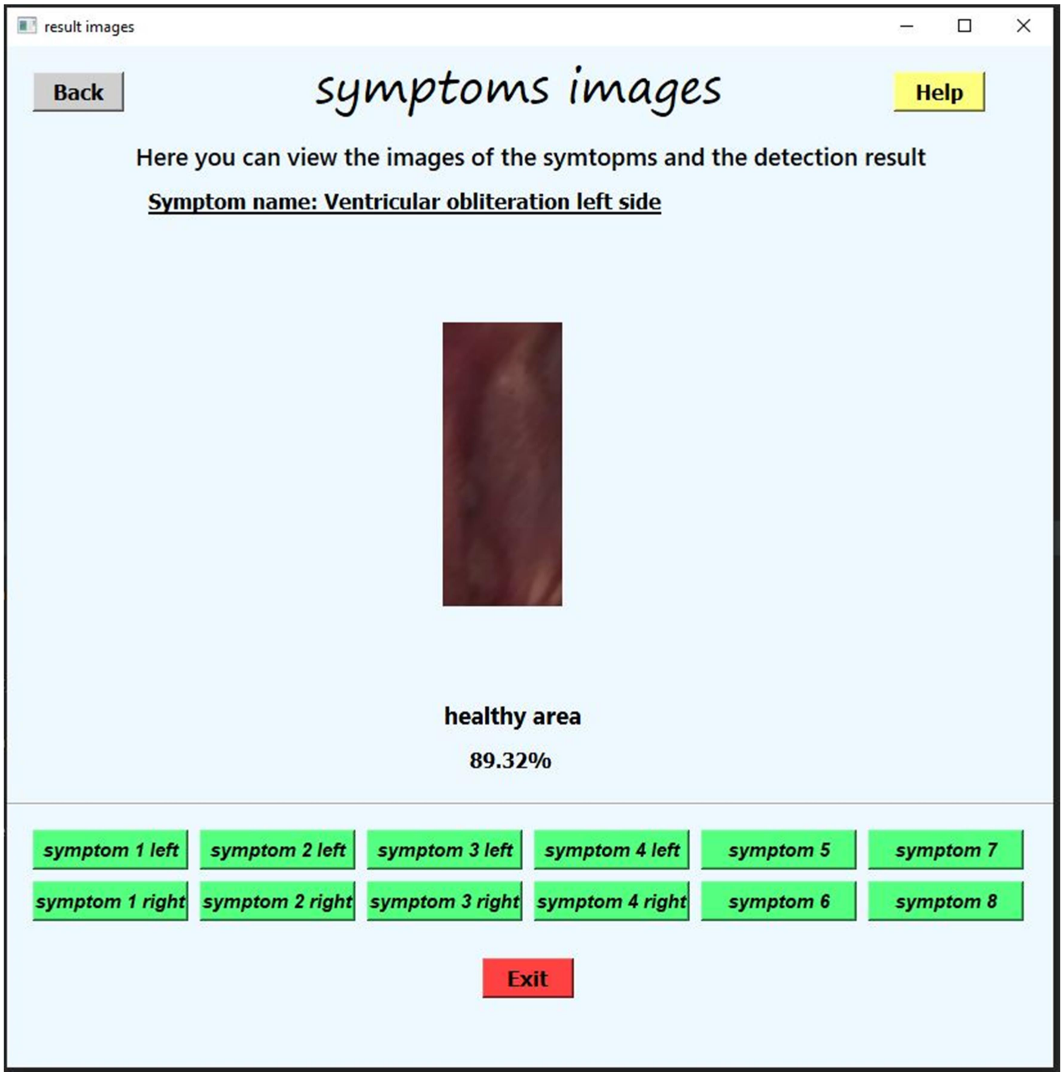


Fig. 40: result images screen.

3.2.2 Class Diagram

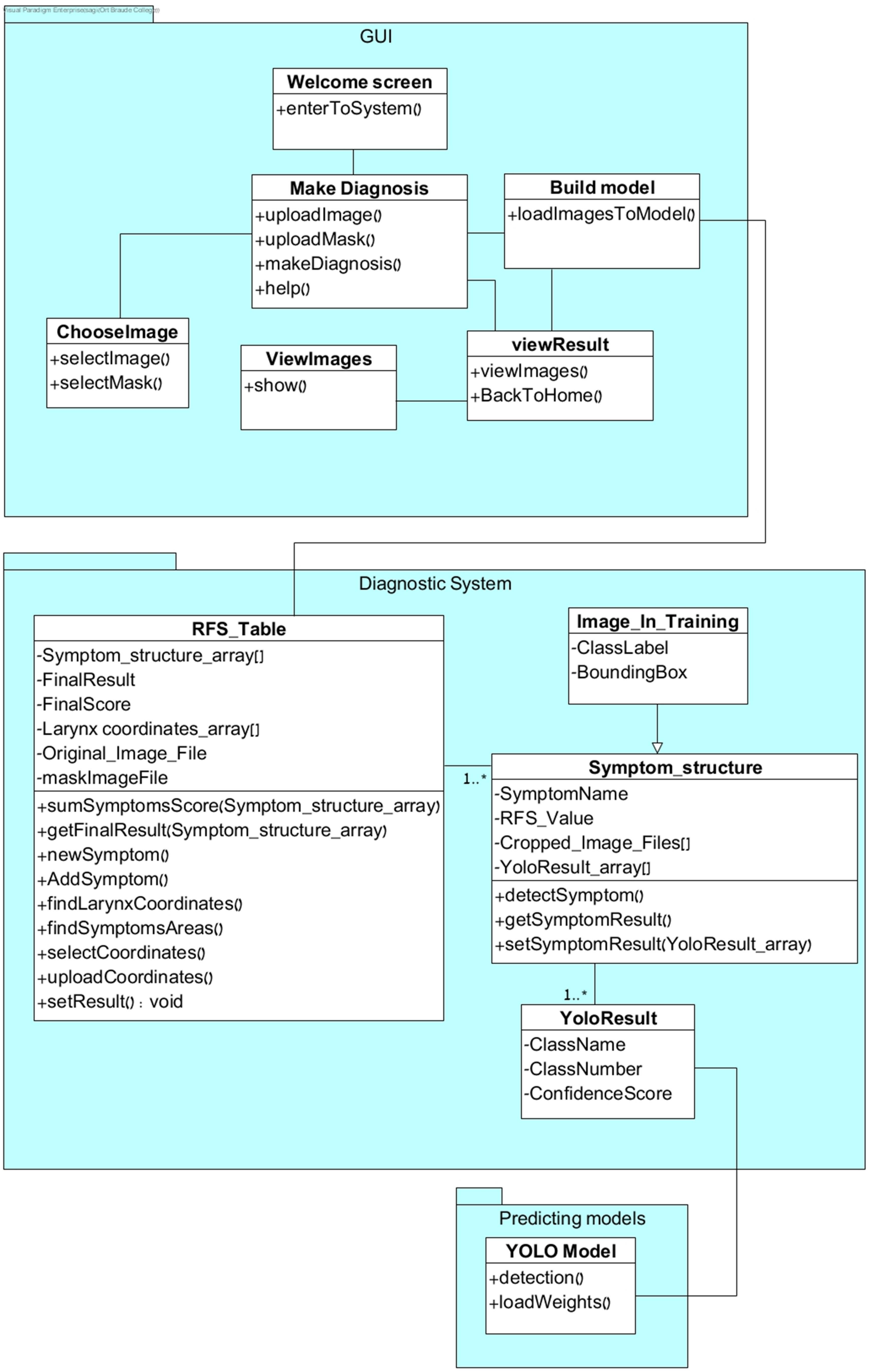


Fig. 41: class diagram.

3.3 Testing

3.3.1 Functional testing

Test Number	Test Subject	Expected Result	Result
1	Try to upload wrong format file.	The system will not display files that are not in the correct format in the file explorer.	Pass
2	Try to press on “Make Diagnosis” button without uploading an endoscopic image.	Error message appear: “No image was selected, please select an image!”.	Pass
3	Try to press on “Make Diagnosis” button without uploading a mask image.	Error message appear. “No mask image was selected, please select an image!”.	Pass
4	Press “Make Diagnosis” button after upload the images.	“build model” screen will be open.	Pass
5	On the result screen, press on “View symptom images” button.	The “result images” screen will be open.	Pass

3.3.2 Algorithm testing

Test Number	Test Subject	Result
1	Upload image of healthy larynx	The result of the detection was negative (healthy larynx) and RFS sum was under 7.
2	Upload image with only “Granuloma” symptom.	The sum of RFS table was equal to 2. The result of the detection was negative.
3	Upload an image of LPR patient.	The result of the detection was positive and RFS sum was above 7.
4	Try to select relevant areas for LPR symptoms from endoscopic image.	Visual inspection that the area is selected correctly.
5	Try to align the larynx image using the rotation transformation algorithms described in section 2.4	Visual inspection that the image is aligned correctly.

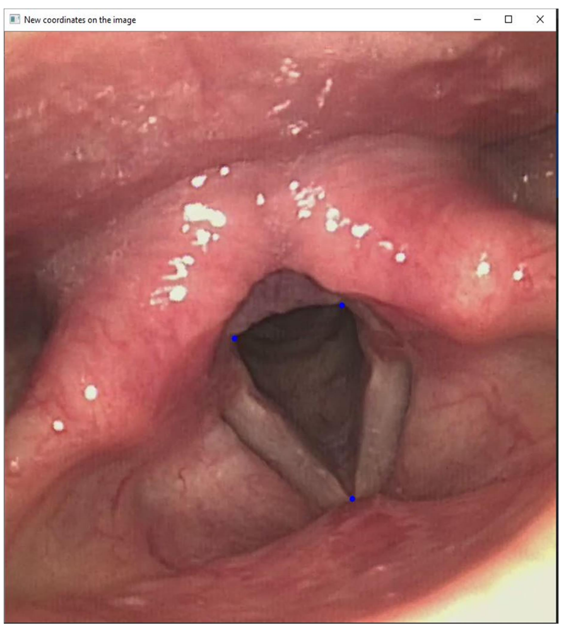
4. RESULTS AND CONCLUSIONS

In this section we describe the experiments we performed during the project, the results, and the conclusions. The full description of our system operations, inputs, outputs, and our algorithms are described in sections 2.5 and 2.6.

4.1 Results

4.1.1 Symptom area selection experiments

To adjust the symptom selection algorithms to cut each symptom area with maximum accuracy, for each algorithm we loaded and cut about 200 images. We tried to select images as varied as possible in terms of anatomy, image angles, and different scales. After applying each algorithm to the set of images, we view the result and we made repairs and adjustments accordingly. We adapted the algorithms to consider as much as possible in the anatomy in each image by setting the different parameters according to larynx scale. For example, we calculated the width of the area for cropping based on the distances between the larynx right vertex and the left vertex. To improve the accuracy, because each image is taken from a different angle, before each region is selected the system aligns the image so that the larynx area is straight, as shown in Figure 42.

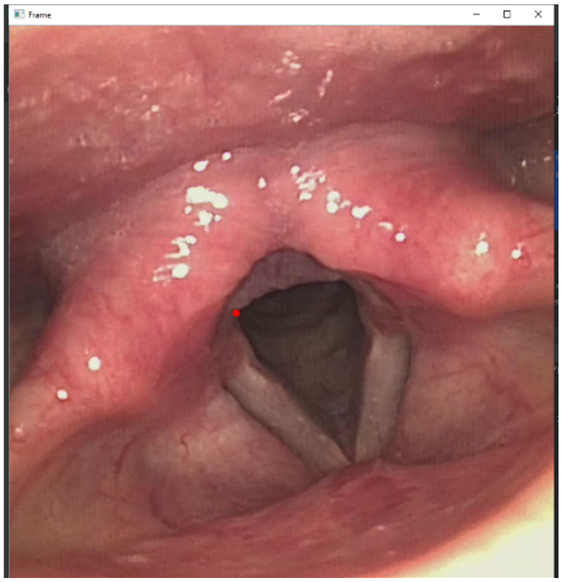


(a) Before

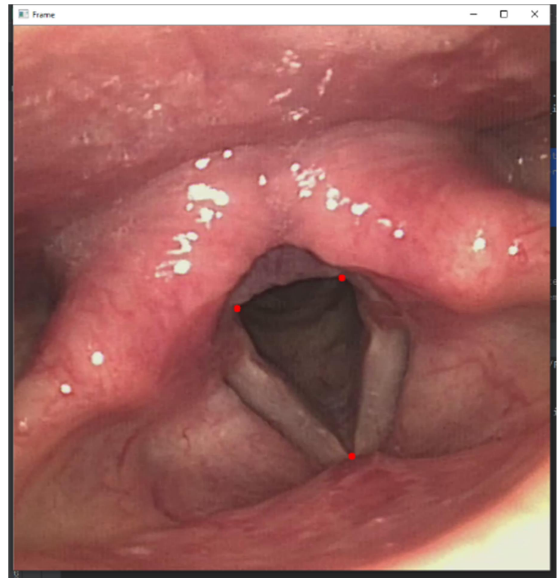
(b) After

Fig. 42: rotating the image so that the larynx area will be straight. The larynx coordinates marked with blue dots.

Another problem we were dealing with at this point is that not all images had a mask, so it was not possible to locate the larynx vertices. To solve this problem, we have written a function that allows the user to manually select the vertices as shown in Figure 43. The system loads the image and displays it to the user, the user clicks with the mouse on the image to select coordinate position. The system marks the selected coordinates with red point and saves the coordinates in .txt file so that they could be loaded and used at each stage.



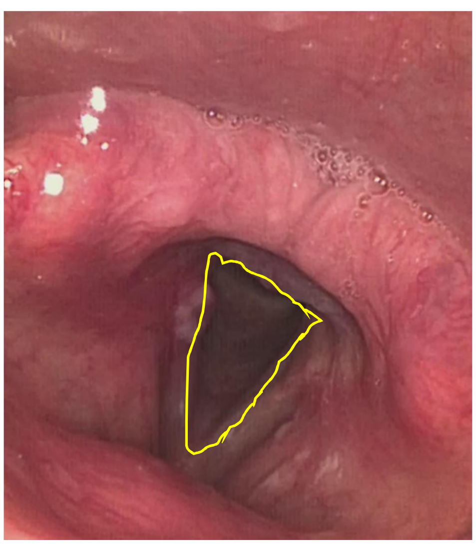
(a) Selecting the first coordinate. The system marked the selected coordinate with red dot.



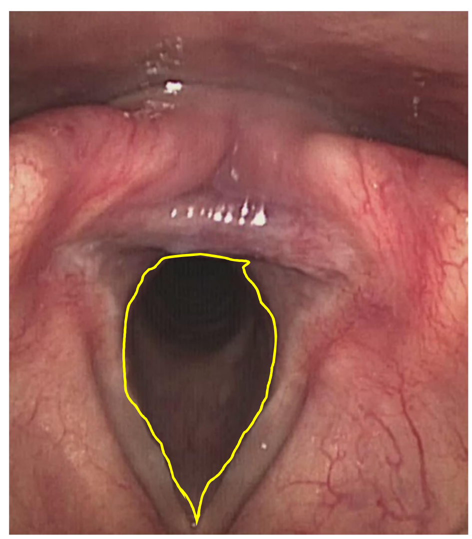
(b) After selecting all 3 coordinates.

Fig. 43: Manual selection of the vertices.

Overall, we have been able to reach a high level of accuracy in cases where the shape of the larynx area is triangular. Cases where the accuracy of our algorithms is less good, are cases where the larynx shape is more circular, the reason is that in the case of the more circular larynx shape, the upper vertices will be more close to each other, as shown in Figure 44.



(a) Triangle larynx shape.



(b) Circular larynx shape.

Fig. 44: example for different larynx shapes.

4.1.2 Training of the network

To train the network, we created a separate training set for each symptom. Each training set consists from array of endoscopic images that contain only the relevant area for that symptom. To train the network to distinguish between an image with symptom and an image without symptom, each training set contains images with a symptom as well as images of healthy relevant area. In each training set we defined two classes. Each image in the training set was labeled according to its class. $Class_1$ was representing image of healthy area, without the symptom, and $Class_2$ was representing image of the relevant area with the symptom. The sizes of the different training sets are different from symptom to symptom, depending on the number of images that was available to us. To increase the number of images, we used different techniques of data augmentation such as rotation and flipping [13].

In cases where we had a few images of a certain class, we tried to take as many images as possible of the other class in order to teach the network more about the other class, in an attempt to teach the system what this type of area looks like and reduce the chance of mistakes. For example, in symptom number 8 we only had 42 images of sick area, but 120 images of healthy area. The full training sets sizes is described in Table 2.

Training prameters:	symptom 1	symptom 2	symptom 3	symptom 4	symptom 5	symptom 6	symptom 7	symptom 8
Total images in training set	410	212	224	322	167	113	179	162
Total sick images in training set	204	126	108	160	122	78	54	42
Total healthy images in training set	206	86	116	162	45	35	125	120

Table 2: training sets sizes.

we trained all sets with the following parameter:

$batch = 64$

$subdivisions = 16$

$max\_batches\ =\ 4000$

$learning\_rate = 0.001$

We also trained our network using Google Colab [14] service. Google Colab is a free cloud service that allows to run python code and train convolutional neural networks on GPU. Google Colab provides a single 12GB NVIDIA Tesla K80 GPU that can be used up to 12 hours continuously.

Once the training for a symptom was complete, we test the results with images of the symptom area. The testing set included both images of the symptom and images without the symptom. We have seen that for symptoms characterized as texture of the skin, like in the case of the posterior commissure hypertrophy symptom (symptom 6), or in cases were the symptom appears as an object, like in the case of the granuloma symptom (symptom 7), the network accurately diagnoses the symptom. In the case of symptom number 3 (erythema), were the symptom is characterized by redness of the skin, the system has difficulty in accurately identifying the symptom. The full testing result are shown in Table 3.

Testing prameters:	symptom 1	symptom 2	symptom 3	symptom 4	symptom 5	symptom 6	symptom 7	symptom 8
Average loss	0.093	0.04	0.14	0.117	0.019	0.055	0.035	0.063
True Positive (sick images that was detected as sick)	62%	100%	47%	45%	100%	90%	92%	83%
False Negitive (sick but detected as healthy)	14%	0%	47%	40%	0%	10%	0%	17%
True Negitive (healthy images that was detected as healthy)	80%	83%	0%	45%	80%	100%	100%	100%
False Positive (healthy but detected as sick)	20%	17%	58%	50%	20%	0%	0%	0%
Cases where Yolo could not identify any object from sick images	24%	0%	7%	15%	0%	0%	8%	0%
Cases where Yolo could not identify any object from healthy images	0%	0%	42%	5%	0%	0%	0%	0%
Total correct answers	68%	96%	26%	45%	93%	93%	96%	96%

Table 3: testing results and average loss that received after training

4.1.3 Execution examples

After we completed all training, we test the results of our system in comparison with doctors' results. We selected images of 11 different patients. For each patient, the system performed full diagnosis for all the symptoms. Then we checked which symptoms our system was found in comparison with the symptoms found by the doctors. In Table 4 we present the full results. we marked with green color places where our system agrees with doctors' opinion, and with red color places where it does not. If the score of some symptom is one, the symptom exists. If the score is zero, the symptom does not exist in that image.

	Doctors								RFS Diagnosis System							
Patient Number	Sign1	Sign2	Sign3	Sign4	Sign5	Sign6	Sign7	Sign8	Sign1	Sign2	Sign3	Sign4	Sign5	Sign6	Sign7	Sign8
156	1	0	1	1	1	0	0	0	1	0	1	1	1	1	0	0
163	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
165	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0
166	1	1	1	1	1	1	0	0	1	0	1	1	1	1	0	0
214	0	1	1	0	0	1	0	1	0	1	1	0	1	1	0	1
164	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0
173	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0
201	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
202	1	0	0	0	1	1	0	0	1	1	0	1	1	1	0	0
212	0	1	1	0	0	1	0	0	0	1	0	0	1	1	0	0
217	0	1	1	0	1	1	0	0	1	1	0	1	1	1	0	0

Table 4: Comparison between system diagnosis and doctors' diagnosis

4.2 Conclusions

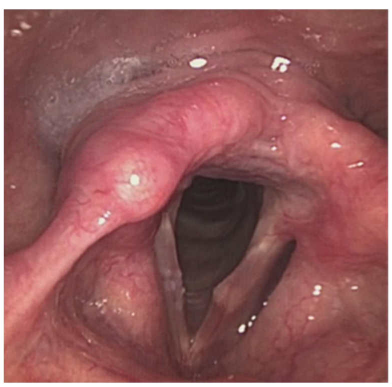
4.2.1 Conclusions from symptom area selection experiments

During the experiments, we saw that alignment of the image according to the upper boundary of the larynx significantly improves the ability to select the relevant area and, in most cases, we were able to reach high accuracy levels.

Although we have tried to adjust our algorithms to consider different larynx anatomies, our assumption is that the larynx area is in triangular shape. As a result of this, in cases where the shape of the larynx is not triangular, but rounder, as we explained in section 4.1.1 (Figure 44), the area selection is less precise. Also, the algorithm for symptom area number 3 was not accurate enough because this area is drastically different from patient to patient.

To improve those algorithms in the future, we recommend trying to find a way to understand the shape of the larynx area in the image, then to cut the image according to the region's anatomy.

In addition, we also saw that the quality of the masks, which we received from a previous project, also affects the accuracy of cutting and finding the larynx coordinates. In many cases, as demonstrated in Figure 45, the masks were not precise enough. Because of that, finding the larynx coordinates was less accurate and affected the cutting accuracy. We think that it is possible to train other convolutional neural network to perform segmentation on the larynx area and produce in that method better and more accurate masks, thereby greatly improving the symptom finding.



(a) Larynx original image.

(b) The corresponding mask for the image.

Fig. 45: example for inaccurate mask.

4.2.2 Conclusions from training of the network

During the training we were able to reach good accuracy in most cases as we demonstrated in section 4.1.2 (Table 3). We have seen that when a symptom appears in an area in the form of an object, edema, or a texture of the skin, the network manages to identify it in good accuracy.

For symptom number 3 - erythema, the results were not good enough. This symptom is characterized by color and not as an object or edema. We think that this is the reason that the network has not been able to diagnose it well. We think that currently, convolutional neural network is not good enough solution for the tasks of diagnose this symptom and more research needs to be done.

We will recommend using our system as a decision support system for LPR diagnosis, which gives doctors an emphasis on whether the patient have symptoms of LPR or not.

REFERENCES

[1] C. N. Ford, “Evaluation and Management of Laryngopharyngeal Reflux”, *JAMA* 294 (12): 1534-1540, 2015.

[2] "Laryngopharyngeal Reflux (LPR)", 2018.

https://my.clevelandclinic.org/health/diseases/15024-laryngopharyngeal-reflux-lpr

[3] S. W. Lee, C. S. Bang, Y. S. Kim, G. H. Baik, D. K. Kim, Y. D Kim, K. H. Han, S. J. Lee, J, "Interrater Reliability among Endoscopists: Diagnosis of Laryngopharyngeal Reflux Based on the Reflux Finding Score Determined by Upper Endoscopy", *The Korean Journal of Helicobacter and Upper Gastrointestinal Research* 17(1): 26-32, 2017.

[4] A. M. Campagnolo, J. Priston, R. H. Thoen, T. Medeiros, A. R. Assunção, "Laryngopharyngeal Reflux: Diagnosis, Treatment, and Latest Research", *Int Arch Otorhinolaryngol* 18: 184-191, 2014.

[5] K. O'shea and R. Nash, "An Introduction to Convolutional Neural Networks", *CoRR abs/1511.08458*, 2015.

[6] V. Luhaniwal, "Forward propagation in neural networks — Simplified math and code version", https://towardsdatascience.com/forward-propagation-in-neural-networks-simplified-math-and-code-version-bbcfef6f9250, 2019

[7] F. A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte, "Breast Cancer Histopathological Image Classification using Convolutional Neural Networks", *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016.

[8] “Fully Connected Layers in Convolutional Neural Networks: The Complete Guide”, https://missinglink.ai/guides/convolutional-neural-networks/fully-connected-layers-convolutional-neural-networks-complete-guide/.

[9] M. Chablani, "YOLO — You only look once, real time object detection explained", https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006, 2017.

[10] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement”, *arXiv preprint arXiv:1804.02767*, 2018.

[11] A. Kathuria, "What’s new in YOLO v3?”, https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b, 2018.

[12] "Rotation Transformation", 2018.

https://www.onlinemathlearning.com/rotation-transformation.html

[13] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning", *arXiv:1712.04621*, 2017.

[14] T. Carneiro, R. V. M. Da Nobrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, P. P. R. Filho, “Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications”, *IEEE Access, 1–1.* doi:10.1109/access.2018.2874767, 2018.