בס"ד



Software Engineering Department  
ORT Braude College

Capstone Project Phase A – 61998

**Automatic selection of suitable images from laryngoscopy videos**

24-1-R-18

Yovel Cohen yovel131998@gmail.com

Eyal Koubi [eyal4845@gmail.com](mailto:eyal4845@gmail.com)

Supervisor: Dr. Elena Kleiman

**Table of Contents**

Abstract .......................................................................................... 3

1. Introduction .................................................................................... 4
2. Related Work .................................................................................. 5
3. Background .................................................................................... 6

3.1 Grayscale and Imaging ............................................................. 6

3.2 Digital Image Histogram .......................................................... 8

3.3 Region Image Segmentation Using Area Growing .................. 9

3.4 Block Matching Algorithm ....................................................... 12

3.5 Autofocus Functions ................................................................. 14

1. Expected Achievements .................................................................. 17
2. Research / Engineering Process ...................................................... 18

5.1 Process ...................................................................................... 18

5.2 Product ...................................................................................... 19

5.2.1 Histogram Creation and Otsu's Thresholding ............ 19

5.2.2 Region Segmentation Using Area Growing ............... 21

5.2.3 Triangle Detection ...................................................... 26

5.2.4 Using Block Matching for Detect the Next Frame …. 28

5.3 Product Diagram and GUI …………………………............…. 33

5.3.1 Use Case Diagram ………………………...………... 33

5.3.2 GUI Design ……………………………………….... 34

5.3.3 Class Diagram ……………………………………… 39

1. Evaluation / Verification Plan ......................................................... 39

6.1 Functional Testing ..................................................................... 39

3.2 GUI Testing …………………………………………………... 40

1. References ....................................................................................... 41

**Abstract**

Our project introduces a system designed for the automatic selection of optimal frames from laryngoscopic videos. This step is critical in addressing the diagnostic challenges associated with Laryngopharyngeal Reflux (LPR) disease, a condition characterized by a wide range of non-specific symptoms that complicate accurate diagnosis. By harnessing various image processing techniques—such as segmentation, clustering, and shape recognition, alongside focus measures—we aim to autonomously identify frames that most effectively reveal the condition of the larynx. This system propose is to enhance the precision and objectivity of LPR diagnosis and also set a foundation for the development of a decision-support tool for medical professionals. Additionally, our project explores the use of advanced autofocus functions to ensure that selected frames meet the standards of clarity and focus, further facilitating the early detection and accurate diagnosis of LPR. Through rigorous experimental research and adaptation of these methodologies to the specific challenges of laryngoscopy video analysis, our project aspires to significantly improve the diagnostic process, offering a pathway towards more reliable and standardized medical assessments.

**Keywords:** LPR, image segmentation, shape recognition, endoscopy, clustering, medical image processing.

**1. Introduction**

In the realm of medical diagnostics, the precision and reliability of identifying conditions early and accurately cannot be overstated. This project specifically targets the diagnostic challenges associated with laryngopharyngeal reflux (LPR) disease, a condition marked by a spectrum of non-specific symptoms that significantly complicate the task of making an accurate diagnosis.

Laryngoscopy (see Figure 1), a procedure essential for the visual examination of the larynx (see Figure 2), provides invaluable insights into the condition of the throat and voice box. However, the current practice in analyzing laryngoscopy videos heavily depends on the subjective judgment of medical professionals. This subjectivity introduces variability in diagnosis outcomes, underlining the necessity for a computer system for diagnosis.

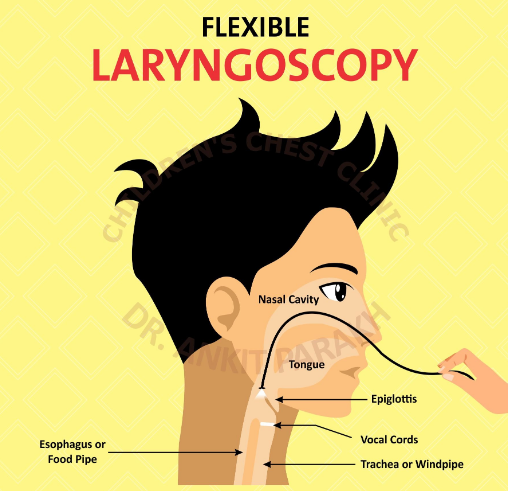


Figure 1: The Laryngoscopy task

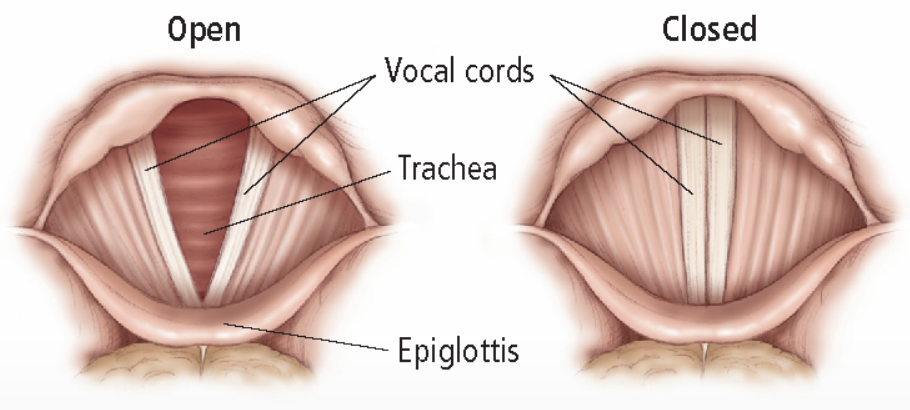


Figure 2: The Larynx itself (open and close)

The core ambition of our initiative is to enhance the diagnostic process by developing an advanced algorithm capable of autonomously selecting the most informative frames from laryngoscopy videos, a critical step towards a more accurate and objective diagnosis of LPR.

Our project aims to mitigate these challenges by introducing an algorithm that utilizes state-of-the-art image processing techniques, including segmentation, clustering, and shape recognition, focus measures. These techniques are applied to meticulously analyze laryngoscopy videos and automatically identify frames that are most indicative of LPR, thereby offering an important in the standardization and objectivity of diagnostic processes.

Our research does not solely rest on the application of these techniques but extends to our systematic approach to refining and adapting these methodologies to the nuanced requirements of laryngoscopy video analysis. We aim to adapt and enhance existing knowledge, addressing specific challenges. Experimental research will play a pivotal role in our project, allowing us to test various parameters and methodologies for frame selection, ensuring that our algorithm will be a basis for a system that exceeds current diagnostic standards.

By integrating image processing technologies with a focused approach to diagnostic challenges, this project aims to provide important strides in improving LPR diagnosis, enhancing the efficacy and precision of medical diagnostics for better patient care.

**2. Related Work**

In the dynamic realm of medical diagnostics, the analysis of laryngopharyngeal reflux (LPR) emerges as a sector ripe for innovation, particularly at the juncture of technology and healthcare. Historically, the domain has leaned heavily on machine learning and computer vision to tackle diagnostic challenges, setting the stage for groundbreaking methods in early disease detection and screening. Among these, enhanced image processing technologies have markedly improved the diagnosis of conditions like LPR, which traditionally depended on the subjective interpretation of visual cues.

However, this landscape reveals a conspicuous void in the realm of open-source algorithmic solutions, specifically designed for the intricate analysis required by laryngoscopy video examinations. This shortfall is not due to a lack of existing methodologies but rather their proprietary nature or specialized focus, which curtails their widespread use and adaptability to the distinct obstacles inherent in LPR diagnosis. This scenario accentuates an urgent call for research and the creation of open, versatile algorithms. Such innovations would not only cater to the immediate needs of LPR diagnosis but also foster a culture of knowledge sharing and collective advancement within both the medical and research communities.

Our endeavor aims to fill this void by proposing a groundbreaking, open-source algorithmic strategy focused on selecting optimal frames from laryngoscopy videos. This initiative distinguishes itself by not merely adapting existing solutions but pioneering a framework built on adaptability, precision, and the integration of cutting-edge image processing techniques. Through experimentation and methodical development, our project is designed to tailor these methodologies to the unique challenges of LPR diagnosis, thereby redefining the benchmarks for accuracy, objectivity, and accessibility in the field of medical diagnostics.

**3. Background**

**3.1** **Grayscale and Imaging**

The RGB color model is a foundational concept in digital imaging, representing images in terms of three primary colors: Red, Green, and Blue. This model operates on the principle that these three colors, when combined in various proportions, can produce a wide spectrum of colors, including the full range of visible light perceived by the human eye. By adjusting the intensity of each color from 0 to 255, it allows for the creation of over 16 million distinct colors, enabling rich and vibrant color images (see Figure 3).



Figure 3: Red – green – blue scale images

A grayscale image is a digital representation where each pixel denotes an intensity value devoid of color [1]. These pixels, the smallest unit of an image, carry information about the light intensity but not about hue or saturation. Ranging from 0 to 255, this scale allows for 256 distinct shades, where 0 represents the absence of light (black), 255 signifies the highest intensity (white), and values in between depict various shades of gray [2] (see Figure 4). This simplicity in representation significantly reduces the data complexity and storage requirements compared to full-color images, making grayscale particularly valuable in applications where color information is less critical.

3.1.1 Grayscale Color Model

The grayscale color model is characterized by its unique ability to represent an image using solely shades of gray, varying from black to white [3]. Unlike one-bit bi-tonal images, which use just two colors (black and white) to represent information, grayscale images incorporate multiple shades of gray to convey more detailed intensity information. This range of gray shades enhances the depth and detail of the image, allowing for a nuanced representation of the visual information.

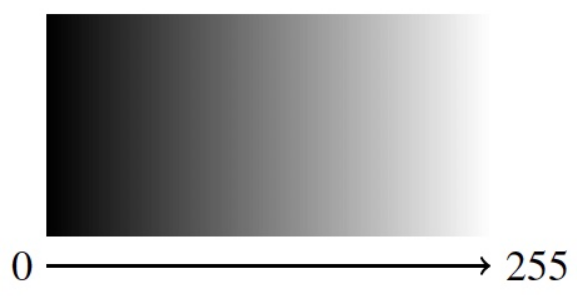


Figure 4: Pixel values ​​in grayscale

Grayscale images are often produced through a process that measures the intensity of light at each pixel according to a weighted combination of the light's frequencies or wavelengths [1]. In scenarios where the image captures a single frequency, it is considered truly monochromatic. However, most grayscale images in digital imaging result from converting color images using specific formulas that reflect the human eye's sensitivity to different colors. The most prevalent of these is the luminosity method, which applies a weighted average to the RGB (Red, Green, Blue) values of each pixel, emphasizing green due to its midrange wavelength and soothing effect on human vision [2]. The formula:

New Grayscale Image = (0.3×R) + (0.59×G) + (0.11×B)

assigns weights of 30% to red, 59% to green, and 11% to blue, acknowledging the varying impact of each color on perceived brightness (see Figure 5). This weighted approach ensures that the converted grayscale image closely mimics the original scene's visual and emotional impact [3].



Figure 5: Original RGB Image and Grayscale Image

By employing this method, the transformation from a color image to grayscale not only preserves the essential visual details but also adjusts the image to better suit applications where color differentiation is less important. Through this nuanced process, grayscale imaging provides a bridge between the multi-colored world and the binary clarity of black and white, capturing the complexity of light and shadow in a format that is both versatile and informative.

**3.2** **Digital Image Histogram**

3.2.1 Introduction to Histograms in Digital Imaging

A histogram in the context of digital imaging represents a graphical representation that shows the distribution of tones in an image. It plots the frequency of each pixel intensity level, providing a visual summary of the tonal range within a digital photograph or image [4]. The x-axis of the histogram displays the intensity levels, typically ranging from 0 (black) to 255 (white) for an 8-bit grayscale image, whereas the y-axis represents the number of pixels at each intensity level (see Figure 6).

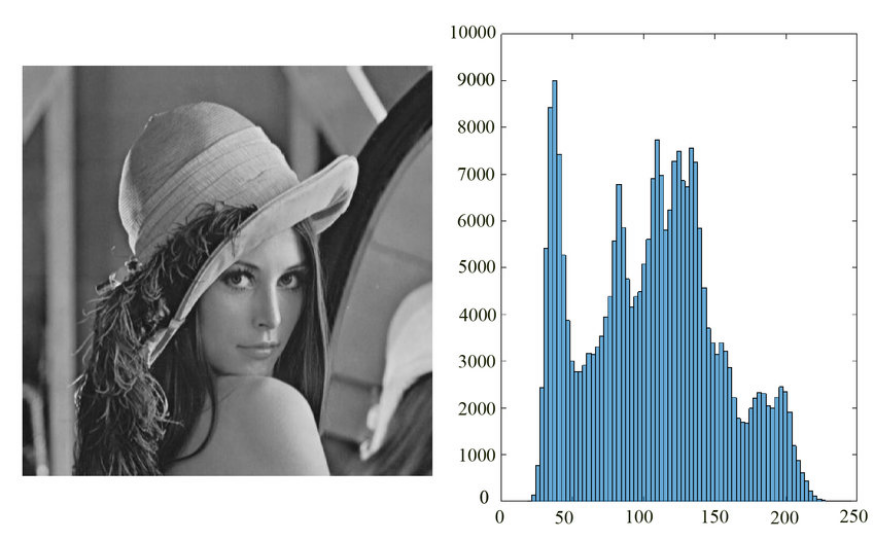


Figure 6: Histogram contrast levels

3.2.2 Importance of Histograms

Histograms serve as a critical tool for photographers, graphic designers, and image processing experts, offering insights into an image's exposure, contrast, dynamic range, and overall tonal quality [4]. They aid in identifying images that are underexposed (most data skewed to the left), overexposed (data skewed to the right), or well-balanced (even distribution across the histogram). Furthermore, histograms are instrumental in diagnosing and correcting issues related to image acquisition, such as exposure errors and contrast adjustments.

3.2.3 Histogram and Image Processing

In the realm of image processing, histograms play a pivotal role in various tasks, including histogram equalization and histogram stretching, which are techniques used to enhance image contrast and detail [5]. Specifically, histogram stretching, as detailed in the project, involves adjusting the intensity range of an image to span a desired range of values, thereby improving the visibility of details in the image.

3.2.4 Digital Image Histogram

A grayscale image's histogram typically comprises entries equal to the number of possible intensity levels (k), where each entry h(i) represents the count of pixels with intensity i [5]. Such histograms play a crucial role in identifying issues related to image acquisition and guiding corrective measures.

One notable technique outlined in the literature is histogram stretching, aimed at enhancing the contrast of an image. This process involves identifying the minimum (a) and maximum (b) intensity values present in the image. Subsequently, these values are scaled to a new range (A to B), thereby improving the image's clarity and making details more discernible. The transformation formula provided, [(I - a) / (b - a)] • (B - A) + A, mathematically represents how each pixel's intensity is adjusted to achieve the desired stretching effect [6].

**3.3** **Region Image Segmentation Using Area Growing**

Region image segmentation is a pivotal process in digital image analysis, aiming to divide an image into multiple segments (regions) based on predetermined criteria. Among the various techniques, the area growing method stands out for its simplicity and effectiveness in delineating homogeneous areas within images. This approach begins with a seed point and expands the region by incorporating neighboring pixels that exhibit similar attributes [7].

3.3.1 Fundamentals of Area Growing Method

Central to the area growing method is the concept of seed selection, which initiates the segmentation process. These seeds are the nuclei from which regions begin to grow, with adjacent pixels being evaluated and added to the region if they align with a predefined similarity criterion. This iterative process continues until no further pixels satisfy the criteria for inclusion [8]. A generalized representation looks like this:

Where measures the similarity between a candidate pixel and the current region , with as the similarity threshold [7].

3.3.2 Seed Selection

The efficacy of area growing segmentation heavily relies on the strategic selection of seed points. These seeds can be selected based on prior knowledge, user input, or through automated methods that determine optimal starting positions within the image, considering factors such as intensity, texture, or other relevant image features [7] (see Figure 7).



Figure 7: The Seed that selected

Seed selection is based on factors such as intensity, texture, or other relevant image features, possibly guided by:

indicating that a seed point is chosen if it satisfies a function against a threshold ​[7].

3.3.3 Similarity Criteria

The criteria for appending pixels to a growing region are fundamental to the area growing methodology. These criteria might include thresholds based on pixel intensity, color, texture, or combinations thereof, influencing the homogeneity of the resulting regions. The meticulous selection of similarity criteria is crucial to achieving successful segmentation outcomes [8] (see Figure 8).

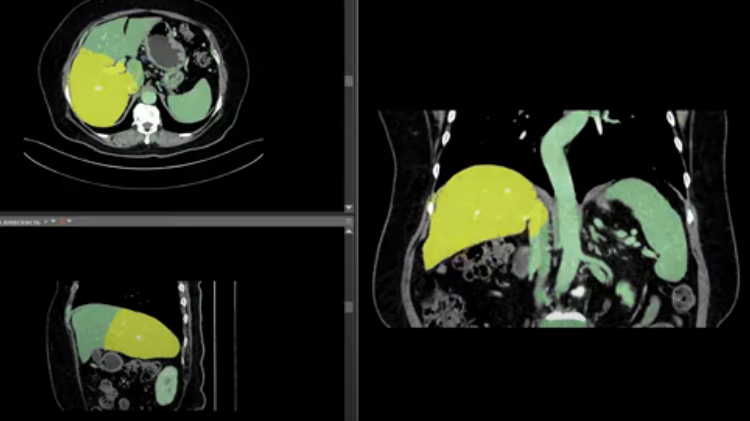


Figure 8: Seed growing depend on the criteria

The similarity criterion for incorporating a pixel into a region considers attributes like pixel intensity, color, texture, or combinations thereof:

Here, is the similarity between pixel and the region , is the attribute of , and is the mean attribute of . A pixel is added to if , with as the predefined threshold [8].

3.3.4 Medical Imaging

The area growing method has found substantial application in the field of medical imaging, where it facilitates the segmentation of anatomical structures and pathological regions. Its utility spans from brain MRI analysis to tumor delineation in CT scans, underscoring its critical role in diagnostic and treatment planning processes [9].

3.3.5 Otso Method for Image Segmentation

The Otso method, also recognized as Otsu's thresholding, is a globally acclaimed algorithm designed for the purpose of segmenting an image by transitioning it from grayscale to binary format. This sophisticated approach identifies the optimal threshold that effectively separates the foreground from the background by minimizing the within-class variance, thereby crystallizing the delineation between distinct regions [10]. Here's a step-by-step breakdown of how the Otso method unfolds:

**1. Compute the Grayscale Histogram (Step 1):**

The first step involves calculating the grayscale histogram of an image , representing the distribution of pixel intensities:

where and are the image's dimensions, and the function evaluates to 1 if pixel has intensity , else 0 [10].

**2. Compute the Cumulative Distribution Function (CDF) (Step 2)**:

CDF, calculated from the grayscale histogram, represents the cumulative probability of pixel intensities up to level :

This formula sums histogram values to , normalized by the image size [10].

**3. Compute the Mean Grayscale Intensity Value of the Image (Step 3):**

The mean grayscale intensity, , is the average of all pixel intensities:

This average is crucial for computing between-class variance in Otsu's method [10].

**4. Compute the Between-Class Variance for Each Possible Threshold Value (Step 4):**

The between-class variance quantifies the separation between foreground and background:

with and as class probabilities, and , their mean intensities [10].

**5. Find the Threshold Value That Maximizes the Between-Class Variance (Step 5):**

Identifying the optimal threshold ​ involves maximizing :

The Otso method is celebrated for its simplicity and effectiveness, providing a robust solution for image segmentation challenges. By automating the thresholding process, it eliminates subjective bias, ensuring consistent and objective segmentation outcomes. This methodological precision is particularly beneficial in applications requiring refined image analysis, such as medical diagnostics, object recognition, and beyond [10].

**Block Matching Algorithm3.4**

3.4.1 Understanding the Basics

Imagine watching a video. What you're essentially seeing is a rapid sequence of still images (frames) that create the illusion of movement. Each object in these frames—be it a person walking, a car driving, or leaves blowing in the wind—is represented by patterns of pixels. The Block Matching Algorithm (BMA) is a sophisticated technique used primarily in video processing to track how these objects move from one frame to the next [11]. This tracking is what we call motion estimation (see Figure 9).

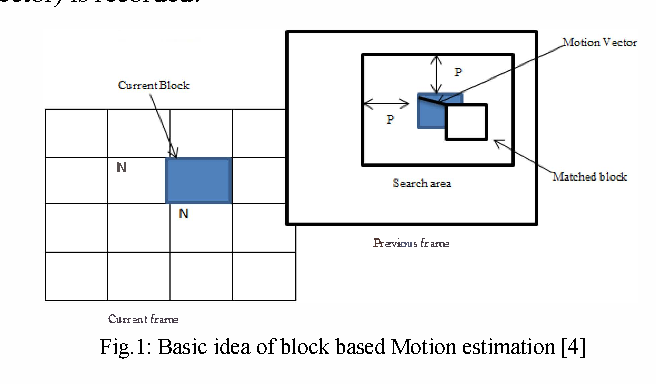


Figure 9: Basic idea of block based motion estimation

3.4.2 How Does It Work?

**Dividing Frames into Blocks:** BMA starts by breaking down a video frame into a grid of smaller, manageable blocks [12]. Think of it as dividing a picture into a puzzle where each piece is a block of the image.

**Finding the Matching Blocks:** The core of BMA involves comparing each block in the current frame with a block in the next frame. However, it doesn't just look for an exact match in the next frame's corresponding position. Instead, it also considers the blocks nearby [13]. This approach helps in accurately tracking the movement, as objects in videos don't just teleport; they move smoothly from one point to another.

**Creating Motion Vectors:** Once a matching block is found, the algorithm defines a motion vector [14] (see Figure 10). This vector represents the direction and distance a block has moved from one frame to the next. By calculating motion vectors for all the blocks in a frame, we get a comprehensive view of how every object has moved in the video.

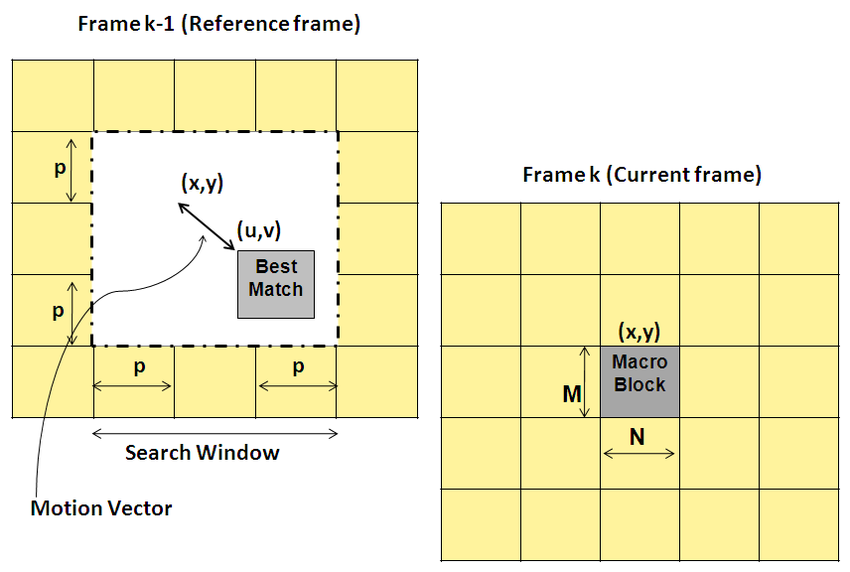


Figure 10: Motion vector

**Search Parameters:** To make this search efficient, BMA limits how far it looks for matching blocks to a specific range, defined by a parameter [15]. This prevents the algorithm from wasting time searching the entire frame for a match, speeding up the process.

**Cost Functions:** The decision on which block matches best is made using cost functions. These functions calculate how similar two blocks are, with the goal of finding the closest match. The Mean Absolute Difference (MAD) and Mean Squared Error (MSE) are two such functions, providing a balance between accuracy and computational efficiency [16].

MAD measures the average absolute difference between the pixels of two blocks:

**Parameters:**

* f(i,j) - The brightness of a pixel in the current block.image..
* N - The size of the block (width and height in pixels).
* - The brightness of a pixel in the reference block, offset by to account for motion.
* - Takes the absolute difference between pixels, ignoring whether one is brighter or darker than the other.

MSE calculates the average of the squares of the differences between the pixels of two blocks:

**Parameters:**

* f(i,j) - The brightness of a pixel in the current block.image..
* N - The size of the block (width and height in pixels).
* - The brightness of a pixel in the reference block, offset by to account for motion.
* - means that larger discrepancies between pixels are given more weight, emphasizing accuracy.

By using these functions, BMA efficiently identifies the best match for each block, enabling accurate motion estimation [17].**ראש הטופס**

**3.5 Autofocus Functions**

**Introduction**

The focus of this section is on the assessment and selection of images based on their focus quality, utilizing focus metrics to gauge the sharpness of each image. These metrics provide a quantitative basis for determining the extent of focus in an image, facilitating the identification and selection of those with optimal clarity. By employing these focus metrics, the methodology prioritizes images that exhibit the highest degree of focus, independent of their overall image quality. This approach ensures that the selection process is guided purely by focus sharpness, allowing for the precise analysis and utilization of images based on their focus levels.

**Focus Measures**

1. **Squared Gradient**

The Squared Gradient measure calculates the sum of squared differences between adjacent pixel intensities, emphasizing edges and textures within an image.

This focus measure is expressed as:

**Parameters:**

* f (x, y) - This represents the intensity of the pixel located at coordinates (x,y) within the image. Pixel intensity can range from 0 to 255 for an 8-bit grayscale image.
* M - This is the number of rows in the image, which equates to the image's height.
* N - This is the number of columns in the image, corresponding to the image's width.
* f (x, y+1) – f (x, y) - This specific part of the formula calculates the difference in pixel intensity between a pixel and its immediate neighbor directly below it (since images are processed as 2D matrices, 'below' refers to an increment in the y-coordinate).
* The squared differences are summed over the entire image (excluding the last row due to the y+1 term) to measure the cumulative amount of edge information, which correlates with the image's perceived focus or sharpness.

This method is beneficial for capturing detailed features in images, making it suitable for diverse applications, including those requiring precise image analysis [18].

2. **Brenner's Gradient** (Horizontal Emphasis)

Brenner's Gradient is another focus measure based on the difference in intensity between pixels, but with a greater distance between compared pixels:

**Parameters:**

* f(x,y) - This represents the intensity of the pixel located at coordinates (x,y) within the image. Pixel intensity can range from 0 to 255 for an 8-bit grayscale M - Represents the total number of rows in the image, essentially its height in pixels. This parameter dictates the range over which the x coordinate iterates, ensuring the operation does not exceed the image boundaries.
* N - Stands for the total number of columns in the image, or its width in pixels. It defines the limit for the y coordinate iteration within the formula.
* - This component calculates the squared difference in intensity between a pixel and its neighbor two pixels further down the same column. By skipping a pixel in between, Brenner's Gradient emphasizes more significant changes in pixel intensity, which are typically indicative of edges or sharp transitions in the image content. The squaring of the difference aims to accentuate larger discrepancies, contributing more heavily to the focus measure, and to ensure that the measure is always positive.

Brenner's Gradient is particularly sensitive to vertical edges due to its method of comparing pixel intensities in the vertical direction. It's suitable for images where the focus is determined by the clarity of vertical structures or edges. The simplicity of this measure lies in its straightforward calculation, requiring no complex operations like convolutions or Fourier transforms, making it computationally efficient and easy to implement.

This method is known for its ability to detect significant changes in intensity, useful in identifying clear, focused areas in an image [18].

3. **Variance of Intensities**

The "Variance of Intensities" measure quantifies the focus of an image by evaluating the variance in pixel intensities [19]. This statistical approach does not involve partial derivatives and is essential for identifying the sharpness of an image. The variance is calculated as follows:

n this formula, denotes the variance of pixel intensities at a given position , is the intensity of the pixel located at coordinates , represents the mean intensity of all pixels in the image, and and are the dimensions of the image. A higher variance suggests a sharper image, as it indicates a wider range of intensity levels, signifying more detailed content. This measure is particularly effective for distinguishing focused images from unfocused ones by highlighting the contrast and clarity present in the image details.

4. **Vollath's F4 Autocorrelation Measure**

Vollath's F4 measure is based on the product of pixel intensities between adjacent rows, emphasizing the vertical correlation of intensities. It then subtracts the product of intensities between every two rows, which can help reduce the influence of noise. This measure can be particularly useful for images with a clear vertical orientation of features or structures [19].

**Parameters:**

* f (x, y) - This represents the intensity of the pixel located at coordinates (x,y) within the image. Pixel intensity can range from 0 to 255 for an 8-bit grayscale image.
* M - This is the number of rows in the image, which equates to the image's height.
* N - This is the number of columns in the image, corresponding to the image's width.
* f (x+1, y)- The pixel intensity at the next row but the same column, providing a vertical comparison.
* f (x+2, y)- The pixel intensity two rows down from the current pixel in the same column, used in the second term of the formula for a wider vertical comparison.

**Conclusion**

The four autofocus functions reviewed offer distinct approaches to assessing image sharpness, each with its strengths in detecting focus in digital images. These methods, characterized by their computational simplicity and effectiveness, are invaluable tools in applications where image detail is paramount.

**4. Expected Achievements**

Our primary aim is to develop an algorithm capable of automatically selecting optimal frames from laryngoscopy videos for diagnosis. This tool, designed to be part of a decision-support system for medical professionals, is essential due to the limited availability of data suitable for neural network applications. We must work with real videos, considering the scarcity of available data.

Our focus is on addressing the challenge of identifying the most suitable frames from videos of varying quality, particularly in detecting and recognizing the open larynx shape. To achieve this, we leverage the unique characteristics of laryngoscopy videos, such as their consistency in image similarity and the presence of the V shape in the throat.

Throughout the project, we explore and adapt image processing methods, including segmentation, clustering, block matching, and sharpness functions, tailored to our specific task. Experimentation is vital in determining the most effective approaches and parameter settings given the constraints of limited data availability.

Key milestones for this semester include gaining a deep understanding of the medical problem and symptom analysis, selecting and adapting suitable image processing methods and algorithms, and designing a comprehensive test plan for post-implementation validation. Concurrently, we document our progress in the project book.

Practically, we plan to utilize VLC software for frame cutting and Java for development in subsequent phases. By semester's end, we aim to have a well-defined algorithmic approach to address the problem, accompanied by a detailed test plan for experimentation. This groundwork will facilitate the successful development and implementation of our diagnostic tool within the larger decision-support system for medical professionals.

**5. Research / Engineering Process**

**5.1 Process**

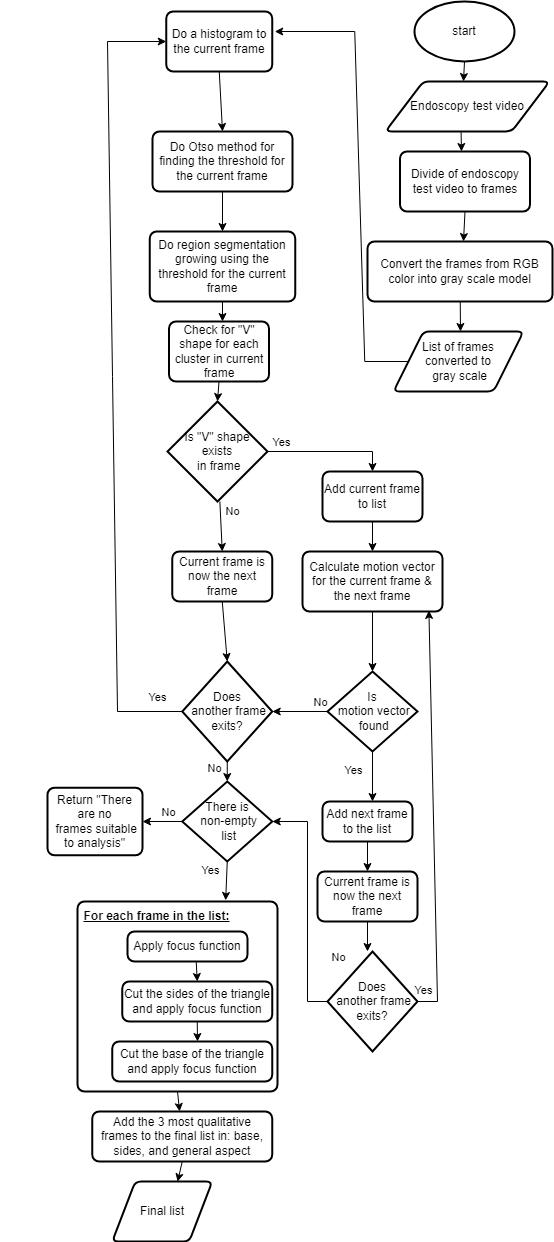


Figure 11: Generic algorithm flow chart

The flow diagram provided presents a structured approach to frame selection within laryngoscopy videos for medical analysis. The process begins by dividing an endoscopy test video into individual frames. Each frame is then converted from the RGB color model to a grayscale model to simplify the intensity analysis. For each grayscale frame, a histogram is generated, and Otsu's method is applied to determine an optimal threshold for binary segmentation. This threshold is used in a region segmentation process, where a search for a specific 'V' shape - indicative of the laryngeal structure - is conducted within the frame.

If the 'V' shape is identified, the frame is added to a list of potential candidates. The process then continues to calculate motion vectors between consecutive frames, Frames with detected motion vectors are also added to the list. Once all frames are processed, a focus function is employed to select the most qualitative frame from different aspects of the laryngeal structure - such as the 'sides' and 'base' of the triangle representing the laryngeal view.

In summary, the flow diagram illustrates a complex yet systematic procedure for identifying the most informative frames for medical evaluation, aiming to enhance the accuracy and efficiency of laryngeal assessments in patients potentially suffering from conditions like LPR.  
And now, we will detail each of the steps explicitly.

**5.2 Product**

**5.2.1 Histogram Creation and Otsu's Thresholding**

5.2.1.1 Histogram Creation  
To begin selecting pertinent images from endoscopy videos for diagnostic evaluation, we first process the video frames by converting them to grayscale. This conversion simplifies the data and focuses on the intensity of light in each pixel, disregarding its color. Once in grayscale, we analyze each frame to construct a histogram that maps the distribution of pixel intensities.

In this histogram, the x-axis represents the possible intensity values that can range from 0 (completely black) to 255 (completely white) for 8-bit images, and the y-axis denotes the frequency of each intensity level within the frame. To construct this histogram, we iterate through each pixel in the image, determine its intensity, and increment the corresponding value in the histogram array.

This histogram serves as a critical tool for understanding the variations in pixel intensities across the frame. It is essential for the application of Otsu's thresholding method, which relies on this distribution to automatically find an optimal threshold value. This value helps in distinguishing between the foreground (areas of diagnostic interest) and the background, enabling the selection of frames that are likely to contain diagnostically relevant information.

ראש הטופס

5.2.1.2 Otsu's Thresholding

Otsu's method is a fundamental algorithm in image processing, designed to automatically determine the optimal threshold for distinguishing the foreground from the background in an image. This technique is particularly effective for images where there is a clear distinction between the foreground and background intensities.

The process begins by calculating the total weighted sum of the histogram frequencies to understand the overall intensity distribution. Then, for each possible threshold value (ranging from 0 to 255 in an 8-bit grayscale image), the algorithm calculates the weight (frequency) of the pixels considered as the background (wB) and the foreground (wF), as well as their mean intensities (mB and mF). It assesses the between-class variance, which is the squared difference between the mean background and foreground intensities, multiplied by the weight of the background and foreground pixels. The essence of Otsu's method lies in maximizing this variance, which signifies the optimal separation of the foreground and background. The threshold that yields the highest between-class variance is chosen as the optimal threshold.

The **otsusThreshold** Function: A Step-by-Step Guide

Key Functions and Variables in Otsu's Thresholding

* **computeTotalSum(histogram):** Directly computes the total weighted sum of the histogram, facilitating the calculation of the mean foreground intensity.

**Variables Explained**

* **sumB:** Accumulates the sum of the product of each intensity level and its frequency for the background. It's crucial for calculating the mean background intensity (**mB**).
* **wB:** The weight or total frequency of background pixels up to the current threshold, vital for determining the proportion of the image classified as background.
* **maximum:** Tracks the highest between-class variance found during the threshold search, indicating the optimal separation point.
* **totalSum:** Represents the total weighted sum of the histogram, used to calculate the mean foreground intensity (**mF**) efficiently.

This approach to Otsu's Thresholding leverages direct computations to optimize performance while ensuring accurate segmentation of the image into foreground and background components.

**5.2.2 Region Segmentation Using Area Growing**

**Introduction**

In the realm of digital image processing, especially within the context of medical imaging, accurately segmenting regions of interest from the background is paramount. Traditional segmentation techniques often struggle with inconsistencies in lighting, which can significantly hinder the analysis. To address this challenge, we present an advanced region segmentation algorithm that utilizes area growing methods, enhanced by a unique tolerance mechanism for light variation. This method is particularly adept at distinguishing the laryngeal structure in laryngoscopy frames, where precise segmentation can dramatically impact diagnostic outcomes.

**Methodology**

To segment regions of interest from grayscale images, our algorithm begins by identifying the darkest pixel in each image, using this pixel as the initial 'seed' for region growth. This process involves examining the grayscale intensity of every pixel, selecting the one with the lowest intensity value as the starting point.

From this seed, the algorithm expands outward, incorporating adjacent pixels into the region based on specific criteria. These criteria include both the intensity threshold derived from Otsu's method and a tolerance for light variation. This approach allows the algorithm to navigate around minor inconsistencies in pixel intensity, such as those caused by light flashes, ensuring that relevant pixels are not excluded from the region.

Neighbors of each pixel are examined, both directly adjacent and diagonally, ensuring that all potential connections within the image boundaries are considered. A pixel is added to the region if its intensity is below the Otsu threshold or if it falls within a defined tolerance of light variation. This process continues, exploring further neighbors of newly added pixels, thereby growing the region in a manner that captures the target structure effectively.

Once no more pixels meet the criteria for inclusion from the current seed, the algorithm searches the remaining pixels outside the grown region for a new seed, starting a new region if such a pixel exists. This cycle repeats until no suitable new seeds are found, resulting in a comprehensive segmentation of all relevant regions within the image.

**Variables Explanation:**

* **grownRegions:** A list that stores the clusters of pixels, or regions, that have been identified and expanded from the seeds. Each region is a collection of pixels that are grouped based on their intensity and spatial proximity according to the algorithm's criteria.
* **darkestPixel:** Represents the pixel with the lowest intensity value in the image, serving as the initial seed for region growth. This selection is based on the assumption that the area of interest (e.g., the laryngeal structure) is darker relative to its surroundings.
* **region:** Represents a single growing region initiated from a seed. It's a collection of pixels that have been deemed to belong together based on the algorithm's logic.
* **queue:** Utilized for a breadth-first search (BFS) to explore neighboring pixels around a current pixel for potential inclusion in the region.
* **Tolerance:** A variable that represents the number of pixels that we are willing to absorb within pixels of a flash of light in the image.

**Functions Explanation:**

* **GrowRegionFromSeed:** This is the main function where the region-growing process is orchestrated. It iteratively expands regions from seeds, checking against intensity criteria and tolerance for light variation.
* **FindDarkestPixel:** Scans the entire image to identify the pixel with the lowest intensity value. This pixel serves as the first seed, initiating the region-growing process.
* **GetNeighbors:** Given a pixel, this function returns a list of its adjacent pixels, considering both direct and diagonal neighbors. It ensures that returned neighbors are within the image's boundaries to prevent indexing errors.
* **IsValidInclusion:** Determines whether a pixel should be added to the current region. It checks if the pixel's intensity is below the Otsu threshold or within an acceptable tolerance of brighter pixels, facilitating growth despite minor light flashes.
* **IsWithinForgivenessPath:** Evaluates whether a pixel, slightly brighter than the Otsu threshold, can be included in the region based on the surrounding context and a predefined tolerance. This function is key to handling variations in lighting within the region of interest.
* **FindNewSeed:** Identifies potential new seed within the image that is not part of any grown region yet but meet the intensity criteria. This ensures that the algorithm continues to explore and expand regions until all relevant pixels are accounted for.

This detailed pseudocode illustrates the methodological underpinnings of our algorithm, showcasing the step-by-step process from the initiation of region growth at the darkest pixel, through to the dynamic expansion of the region. By judiciously selecting adjacent pixels and incorporating a tolerance for light variations, the algorithm ensures comprehensive and accurate region segmentation.

**Outcome and Implications**

Post-implementation, the algorithm exhibits a remarkable ability to accurately delineate the laryngeal structure, even in the presence of challenging lighting conditions. This precision is critical in medical diagnostics, where the clarity of segmented regions can influence the interpretation of laryngoscopic images. Furthermore, the flexibility and adaptability of our approach make it a valuable tool for a wide array of applications beyond medical imaging, wherever accurate region segmentation is crucial.

By integrating the 'forgiveness path' into the area growing technique, we provide a nuanced solution to the perennial problem of light variation in image segmentation. This advancement not only enhances the robustness of the segmentation process but also opens new avenues for research and application in digital image processing.

**5.2.3 Triangle Detection**

**Introduction**

We have adopted the least squares fitting method to robustly detect triangular shapes within these images, which is vital for delineating the V-shaped structure of the larynx, enhancing the accuracy of diagnosing conditions such as Laryngopharyngeal Reflux (LPR).

**Methodology  
Our process involves the following refined steps:**

* **Point Selection:** Recognizing that the most extreme points might not always form the most anatomically accurate triangle, we select points based on their proximity to these extremes:
  + **Lowest Point:** Identify the lowest point and consider a small cluster of points around this base to find the optimal base vertex.
  + **Most Up-Right and Most Up-Left Points:** Similarly, identify the most up-right and most up-left points and select from a cluster of nearby points to finalize the upper vertices. This selection is guided by looking for points that maintain a balance between being anatomically representative and ensuring the triangle remains open and distinct.
* **Formulating the Least Squares Problem:**
  + **Hypothesis Formation:** Assume a model of a triangle with vertices at points **(x₁, y₁), (x₂, y₂), (x₃, y₃).**
  + **Distance Calculations:** Calculate the perpendicular distance from each edge point to each hypothesized triangle side using the formula:
  + **Optimization Problem:** Formulate a function to minimize the sum of the squares of these distances. The objective function to minimize is:

Where is the perpendicular distance from the -th point to the nearest side of the triangle.

* **Evaluation:** Assess the goodness of fit using metrics like the sum of squared residuals (RSS). A smaller RSS indicates a better fit, suggesting a higher likelihood of accurately representing the triangular structure (see Figure 12).

****

Figure 12: Larynx left and right linear lines

**Explanation:**

**The fitTriangle** function constructs a triangle from points carefully selected from clusters near the anatomical extremes, ensuring a realistic and accurate representation. **The computeError** function calculates the sum of squared perpendicular distances from each point in the dataset to the nearest side of the triangle, ensuring that the triangle optimally fits the given points.

**Applications:**

This method is pivotal in our system, enhancing automated detection and analysis of the laryngeal structures in laryngoscopy images. By more accurately capturing the triangular configuration of the larynx, our project significantly advances diagnostic capabilities, contributing to earlier and more precise detection of critical conditions.

**5.2.4 Using Block Matching for Detect the Next Frame**

**Introduction**

Once we detect a V shape in a frame, we examine the next frame to see if it also contains the V shape. This helps us decide if the next frame is suitable or if there's a significant deviation, indicating a problem like poor quality or unclear focus. We focus on the triangle's boundary rather than its interior and select a subset of blocks for analysis to reduce computational load. By finding the best matching block in the next frame and assessing motion vectors, we determine if there's consistent motion between frames.

**Methodology**

Our implementation of the Block Matching Algorithm is tailored to scrutinize motion specifically within the V-shaped area of interest, optimizing our analysis of consecutive frames. Here's how we adapt the BMA to our system:

1. **Defining Boundary Blocks:** Instead of analyzing the entire frame, we focus solely on the boundary of the V shape, where motion is most prominent. This approach allows us to minimize computational overhead by concentrating on relevant areas of interest..
2. **Identifying Matching Blocks:** For each block along the V shape's boundary in the current frame, we search for the most suitable matching block in the subsequent frame. Our search extends beyond exact spatial correspondence, considering nearby regions to account for potential shifts in position.
3. **Calculating Motion Vectors:** Upon identifying a matching block, we compute the motion vector representing the displacement of the block from its original position in the current frame to its corresponding position in the next frame. These motion vectors provide valuable insights into the dynamics of the larynx's movement.
4. **Assessing Motion Consistency:** To ensure reliable motion tracking, we evaluate the consistency of motion across all boundary blocks. This assessment involves calculating the average distance between motion vectors and comparing it against a predefined threshold. If the average distance falls below the threshold, we consider the motion pattern consistent and proceed accordingly.

**Explanation:**

* **findMatchingBlock:** This function identifies the block in the subsequent frame that most closely resembles the current block, based on predefined criteria such as the sum of absolute differences (SAD) or mean squared error (MSE). It compares each block from the current frame's V-shaped boundary with blocks in the next frame's boundary to find the best match.
* **calculateMotionVector:** Computes the vector representing the directional and magnitude changes from the original position of the block in the current frame to its matched position in the next frame. This vector is crucial for understanding the motion dynamics between frames.

**Applications:**

By accurately identifying and analyzing the V-shaped structure of the larynx across frames, it supports the automatic addition of suitable subsequent frames when consistent motion is detected. This reduces the need for frame-by-frame analysis, enhancing efficiency in diagnostic workflows that rely on tracking anatomical changes over time.

**5.2.5 Frame Analysis for Optimal Focus Detection**

**Introduction**

Our project's effectiveness hinges on accurately identifying the sharpest frames from laryngoscopy videos, focusing on specific anatomical details. To facilitate this, we implement segmentation techniques to isolate key areas such as the 'sides' and 'base' of the larynx and evaluate their focus separately along with the overall frame focus.

**Methodology**

We segment each frame and measure its focus, ensuring that both general and specific anatomical features are evaluated for optimal clarity.

**Segmentation and Focus Measurement:**

1. **Frame Segmentation**:
   * **General Frame**: The entire frame is considered for general focus evaluation.

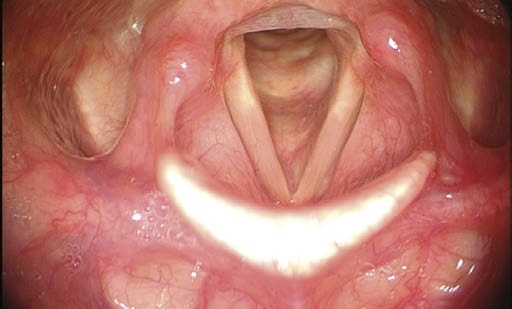


Figure 13: General picture in high quality focus of Larynx

* + **Sides Segmentation**: Specific algorithms segment the lateral parts of the larynx the vocal folds area, typically where crucial muscular structures are visible.

* + **Base Segmentation**: The base is the back side of larynx, often showing subglottic areas, is isolated for precise focus assessment.

תמונה שמכילה תקריב, עור, אורגן, אדם

התיאור נוצר באופן אוטומטי

1. **Focus Measurement**:

* We apply focus metrics such as the Squared Gradient, Brenner's Gradient, Variance of Intensities, and Vollath’s F4 Autocorrelation to quantify the sharpness and clarity of these segments. These measures are chosen for their sensitivity to detailed variations and their ability to capture essential features in the images.

**Explanation:**

* **measureFocus:** The function now indicates that Brenner's Gradient is used as an example, suggesting that other focus measures such as Squared Gradient, Variance of Intensities, and Vollath's F4 Autocorrelation could also be utilized based on the specific needs.

**Applications:** This method is designed to ensure that the most focused frames are selected for each specified aspect ('general', 'sides', 'base'), which aids in more precise diagnostic analyses. By potentially utilizing a variety of focus measures, our approach can adapt to the unique characteristics of different laryngoscopy images, providing a nuanced understanding necessary for accurate medical assessments.

**5.3 Product Diagram and GUI**

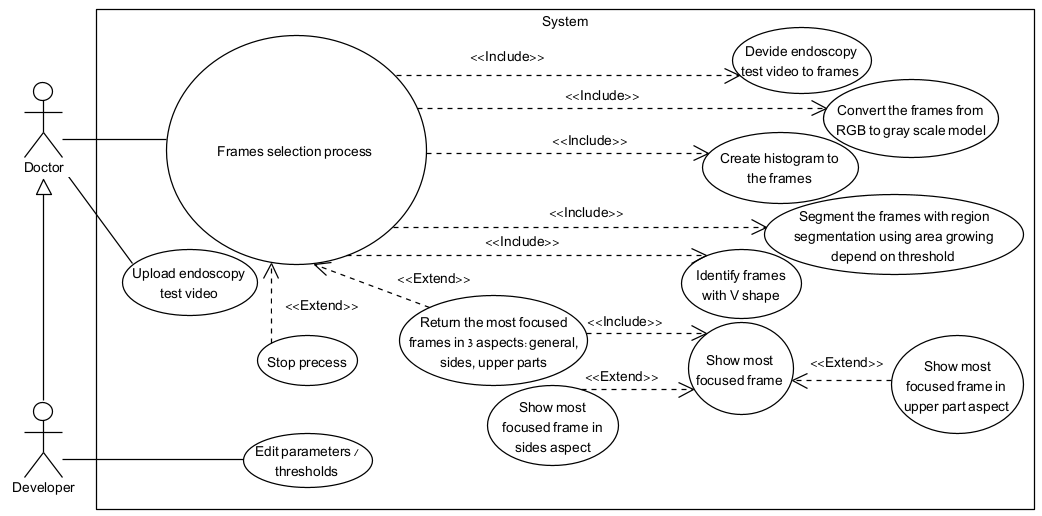
**5.3.1 Use Case Diagram**

Figure 14: Use Case diagram

**5.3.2 GUI Design**

The interface of our software engineering project begins with a user-friendly welcome screen, tailored for both doctors and developers to upload laryngoscopy videos with ease. Following a successful upload, the frame selection process is initiated, utilizing advanced focus functions to dynamically analyze and display the progress. This setup allows for any necessary interruptions. For developers, a dedicated settings screen enables the adjustment of system parameters to refine analysis procedures further. Upon completion of the analysis, the system automatically presents the highest quality image, determined through precise focus functions. Additionally, doctors have the option to view and assess the best side and base views of the larynx, each selected as the highest quality from all frames within the video. This comprehensive functionality supports a thorough diagnostic review from multiple perspectives, ensuring clarity and focus in the final assessment.

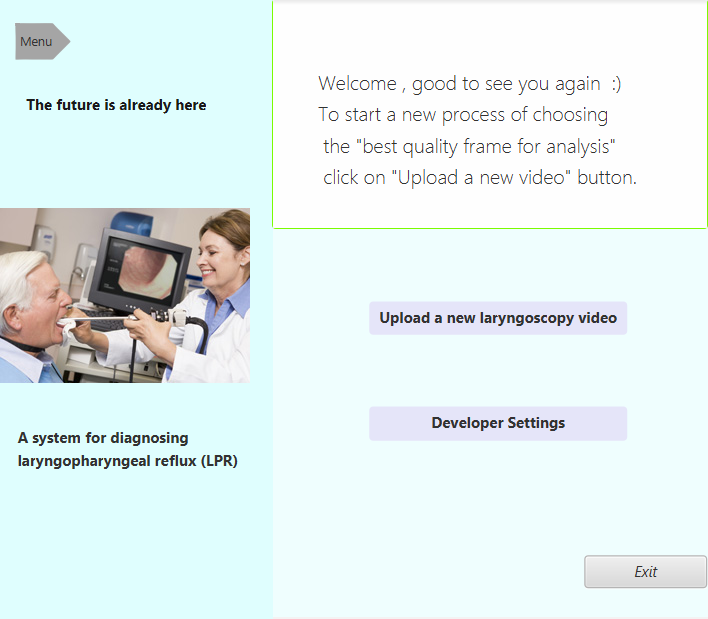
****

Figure 15: GUI Design: Main Screen

The initial interface presents a clear and straightforward layout. The screen includes two main elements: menu options and an interactive area. It features a button for all users, "Upload new laryngoscopy video," which allows users to upload video files to the system. Additionally, there is a "Developer Settings" button, accessible only to developers, which leads to the possibility of adjusting parameters and overall levels within the system's algorithms.

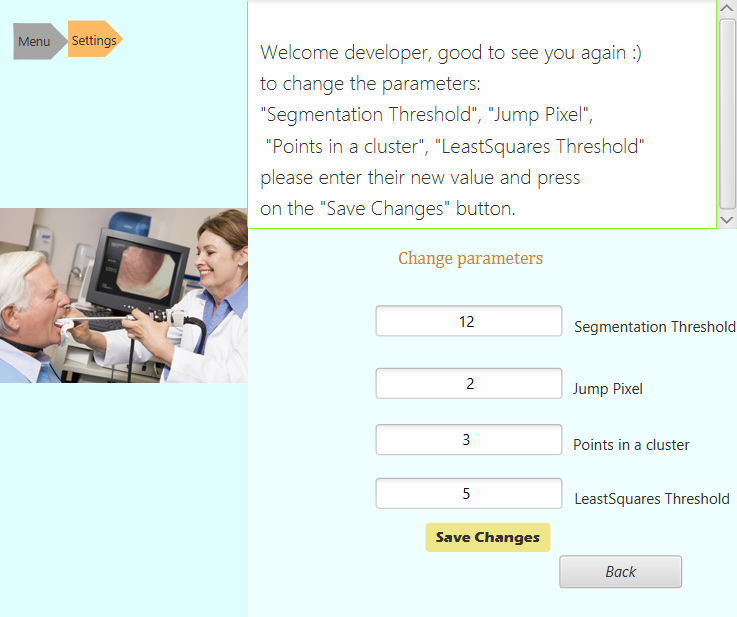


Figure 16: GUI Design: Developer Settings Screen

This screen is accessible only to developers and allows for the modification of key parameters that impact the processing algorithms of the system. Developers can adjust settings such as " Segmentation Threshold", "Jump Pixel", "Points in a cluster" and " LeastSquares Threshold", which are integral to how the system analyzes laryngoscopy videos. By entering new values and clicking the "Save Changes" button, developers can ensure that the system's behavior is aligned with specific project requirements or testing conditions. The "Back" button is provided to return to the main menu or previous screens, allowing for seamless navigation between settings.

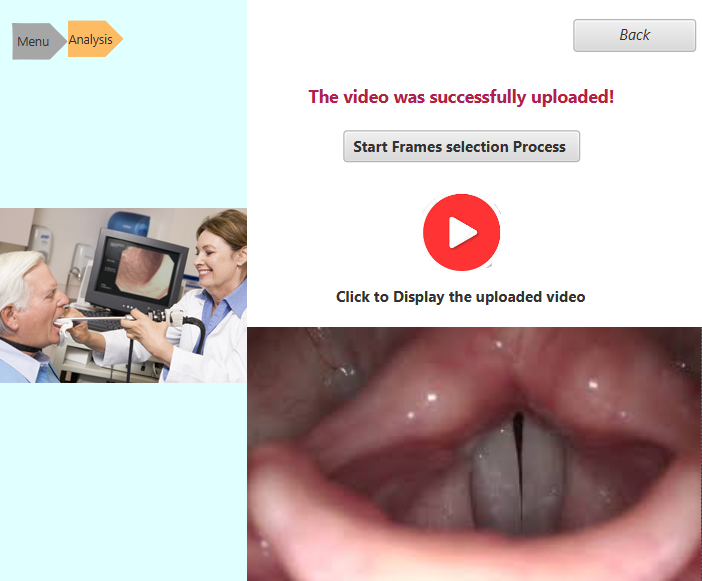


Figure 17: GUI Design: Video Analysis Screen

This screen appears after a video has been successfully uploaded to the system. It confirms the successful upload with a notification message and provides a "Start Frames selection Process" button, which initiates the frame analysis. Additionally, there is a playback button that, when clicked, displays the uploaded video below, allowing users to review the video prior to analysis. The "Back" button located at the top allows users to return to the previous screen if needed.

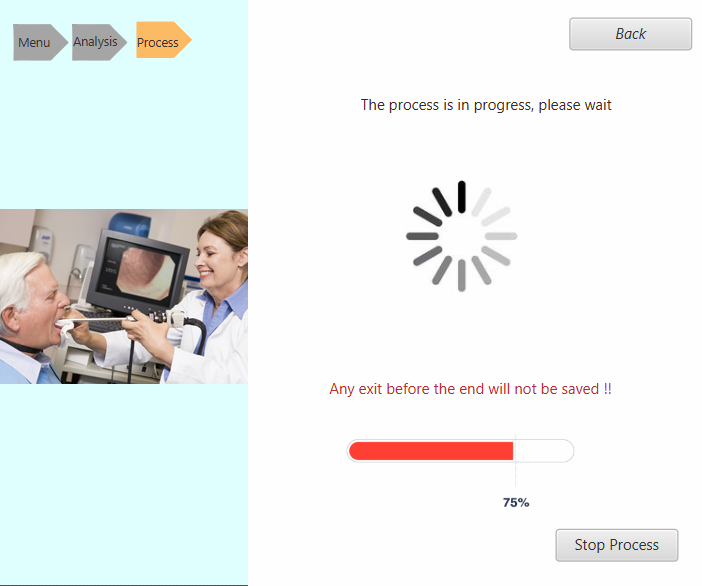


Figure 18: GUI Design: Process Screen

This screen is displayed when the frame selection process is actively running. It features a progress indicator and a message stating "The process is in progress, please wait" to inform users that the analysis is underway. A loading animation is shown to visually represent the ongoing process. Below, there is a progress bar indicating the percentage of completion, and a warning message alerts users that any exit before the process is fully completed will result in unsaved changes. Additionally, a "Stop Process" button is provided, allowing users to terminate the process if necessary. The "Back" button at the top enables navigation back to the previous screen.

תמונה שמכילה טקסט, פני אדם, צילום מסך, אדם

התיאור נוצר באופן אוטומטי

Figure 19: GUI Design: Result Screen

This screen marks the completion of the frame selection process and initially displays the most focused and high-quality image featuring a V shape from the endoscopy video. Additional options are provided to further explore the video's content: "Show Best Sides View" and "Show Best Base View" buttons. These allow users to view alternative images that best represent the sides and the base of the larynx, respectively, from all frames in the video that include a V shape. The "Back" button is available for users to navigate to previous screens, enabling further analysis or adjustments if necessary. This setup ensures a comprehensive evaluation of key anatomical features from multiple perspectives within the video.

**5.3.3 Class Diagram**

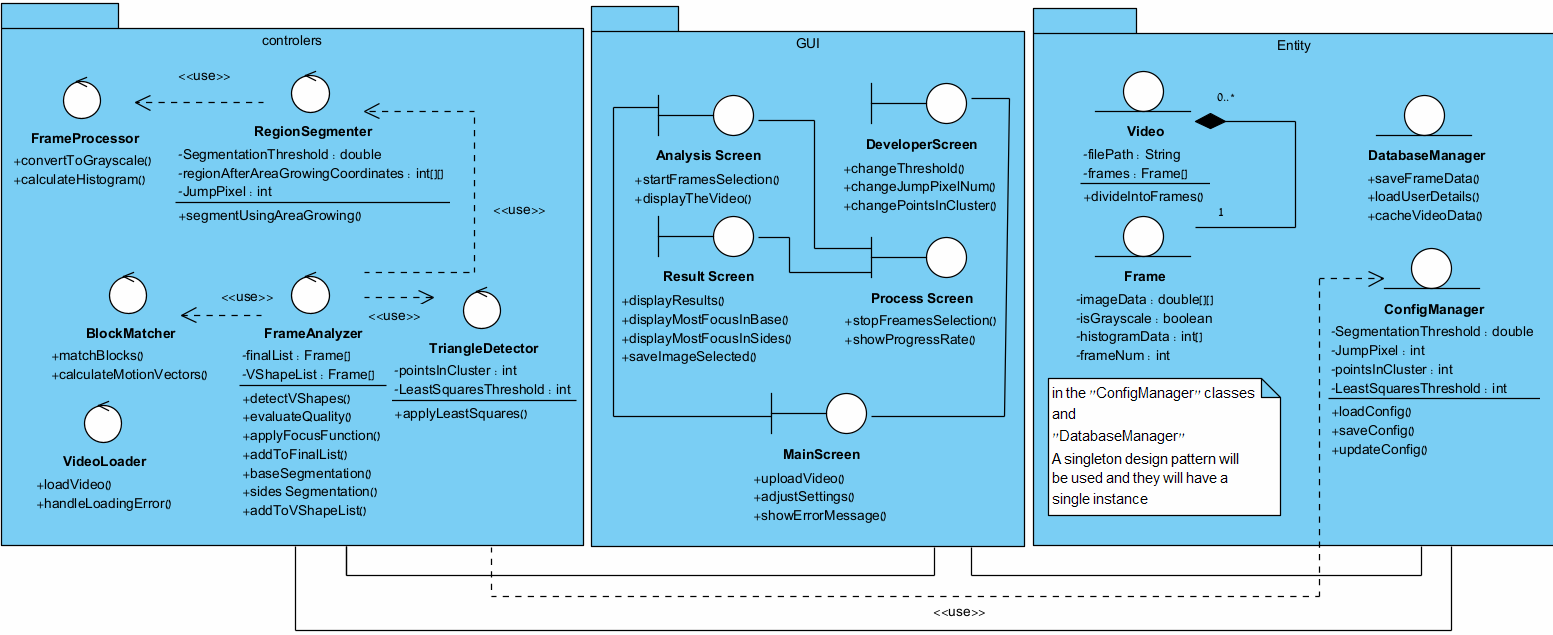


Figure 19: Class Diagram

**6. Evaluation / Verification Plan**

This evaluation plan is designed to ensure our system functions effectively and meets the requirements for identifying and analyzing laryngoscopy video frames.

**6.1 Functional Testing**

Functional testing is integrated to verify that all system components work seamlessly together and achieve the intended outcomes:

| **Test No.** | **Test Subject** | **Testing Action** | **Expected Result** |
| --- | --- | --- | --- |
| 1. | RGB to Grayscale Conversion | Verify the algorithm's output against standard library functions. | System should correctly convert RGB images to grayscale, matching standard outputs. |
| 2. | Frame Segmentation | Test if the algorithm correctly isolates the 'sides' and the 'upper part' of the larynx. | Segmented regions should accurately isolate the 'sides' and 'upper part' of the larynx. |
| 3. | V-Shape Determination | Validate the shape recognition algorithm for detecting V-shaped structures. | Algorithm should correctly identify V-shaped structures in various frames. |
| 4. | Absence of Open Larynx in Frames | Verify system response when no open larynx frames are present. | System should notify the user that no suitable images were detected. |
| 5. | Focus Measurement Algorithms | Assess each focus measurement algorithm against predefined focus benchmarks. | Each algorithm should accurately score the focus based on predefined benchmarks. |
| 6. | Frame Analysis and Selection | Ensure the system identifies and selects the top-quality frame based on sharpness and clarity. | The most qualitatively superior frame should be selected from a set of video frames. |
| 7. | Block Matching Algorithm | Test the algorithm by inserting two subsequent images to check if a motion vector can be built. | System should successfully build a motion vector, demonstrating the algorithm's effectiveness. |
| 8. | Region-Growing Segmentation | Assess the accuracy of the region-growing algorithm by verifying segmented regions' relevance. | Segmented regions should coherently correspond to predefined anatomical features. |

Table 1. Functional Testing Plan

**6.2 GUI Testing**

Acceptance testing will focus solely on the graphical user interface (GUI):

| **Test No.** | **Test Subject & Screen Interaction** | **Expected Result** |
| --- | --- | --- |
| 1. | Application Launch: Initial GUI Load | The GUI should load without errors, displaying the home screen with accessible options for video upload and parameter adjustments. |
| 2 | Upload Inappropriate Video Format | System should display an error message indicating that the video format is not supported. |
| 3 | Start Frame Selection without Video Input | System should display an error message indicating that video input is required. |
| 3. | Parameter Configuration: Adjusting thresholds | The user should be able to change thresholds or parameters via sliders or input fields, and the system should confirm changes. |
| 4. | Video Upload Interaction: Clicking 'Upload Video' | Clicking the 'Upload Video' button should open a file dialog, allowing the user to upload a video and see a confirmation message. |
| 5. | Process Initiation: Starting the analysis | The user should be able to start the frame analysis process with a 'Start' button, triggering backend processes and showing progress. |
| 6. | Display of Focused Frame Options: Selecting best quality photo | The GUI should offer a selection (e.g., radio buttons) for 'General', 'Sides', and 'Upper Part' focused frames and display the choice. |
| 7. | Error Message Handling: Incorrect actions or inputs | Incorrect actions or inputs (e.g., wrong file type upload) should trigger clear, instructive error messages in the GUI. |
| 8. | Successful Operation Confirmation: Completion messages | Upon successful completion of any process, the GUI should display a message or visual cue to inform the user of the success. |
| 9. | Interruption Option: Stopping the process | The GUI should provide an option to interrupt or stop the process, prompting the user to confirm or cancel the interruption. |

Table 2. Gui Testing Plan

**7. References**

]1] **Grayscale**: Gonzalez, R.C., & Woods, R.E. (2002). Digital Image Processing. Prentice Hall.

[2] **Grayscale**: Parker, J.R. (2010). Algorithms for Image Processing and Computer Vision. Wiley-Interscience.

[3] **Grayscale:** Smith, S.M., & Brady, J.M. (1997). SUSAN—A New Approach to Low-Level Image Processing. International Journal of Computer Vision, 23(1), 45-78.

[4] **Digital Image Histogram**: Raju, P. D. R., & Neelima, G. (2012). Image segmentation by using histogram thresholding. International Journal of Computer Science Engineering and Technology, 2(1), 776-779..‏

[5] **Digital Image Histogram**: Pratt, W.K. (2007). *Digital Image Processing: PIKS Inside, Third Edition*. Wiley-Interscience.

[6] **Digital Image Histogram**: Sonka, M., Hlavac, V., & Boyle, R. (2013). *Image Processing, Analysis, and Machine Vision*. Cengage Learning.

[7] **Region Image Segmentation Using Area Growing:**

Adams, R., & Bischof, L. (1994). Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), 641-647. This seminal paper introduces the seeded region growing algorithm, elaborating on the process and its applications in image segmentation.

[8] **Region Image Segmentation Using Area Growing:**

Tremeau, A., & Colantoni, P. (2000). Regions adjacency graph applied to color image segmentation. IEEE Transactions on Image Processing, 9(4), 735-744. Discusses the importance of seed selection and similarity criteria in the context of color image segmentation using region growing methods.

[9] **Region Image Segmentation Using Area Growing**:

Haralick, R.M., & Shapiro, L.G. (1985). Image segmentation techniques. Computer Vision, Graphics, and Image Processing, 29(1), 100-132. Provides a comprehensive overview of image segmentation techniques, including area growing, with a focus on applications and advancements in medical imaging.

[10] **Region Image Segmentation Using Area Growing**: https://www.baeldung.com/cs/otsu-segmentation

[11]**Block Matching Algorithm**: Jain, J.R., and Jain, A.K., "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, vol. COM-29, no. 12, December 1981.

[12] **Block Matching Algorithm**: Ghanbari, M., "The cross-search algorithm for motion estimation," IEEE Transactions on Communications, vol. 38, no. 7, July 1990.

[13] **Block Matching Algorithm**: Tekalp, A.M., "Digital Video Processing," Prentice Hall, 1995.

[14] **Block Matching Algorithm**: Po, L.M., and Ma, W.C., "A novel four-step search algorithm for fast block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, June 1996.

[15] **Block Matching Algorithm**: Zhu, S., and Ma, K.K., "A new diamond search algorithm for fast block-matching motion estimation," IEEE Transactions on Image Processing, vol. 9, no. 2, February 2000.

[16] **Block Matching Algorithm**: : "Block Matching Algorithms For Motion Estimation", DIP 6620 Spring 2004 Final Project Paper, Aroh Barjatya, Student Member, IEEE.

[17] **Block Matching Algorithm**: Srinivasan, R., and Rao, K.R., "Predictive coding based on efficient motion estimation," IEEE Transactions on Communications, vol. 33, no. 8, August 1985.

[18] **Autofocus Functions:** Mir, H., Xu, P., & van Beek, P. (2014). An extensive empirical evaluation of focus measures for digital photography. *Cheriton School of Computer Science, University of Waterloo, Canada*.

[19] **Autofocus Functions:** Santos, A., Ortiz de Solorzano, C., Vaquero, J.J., Pena, J.M., Malpica, N., & del Pozo, F. (1997). Evaluation of autofocus functions in molecular cytogenetic analysis. *Journal of Microscopy*, 188(3), 264-272.