**CHAPTER-1**

**INTRODUCTION**

**1.1 OVERVIEW**

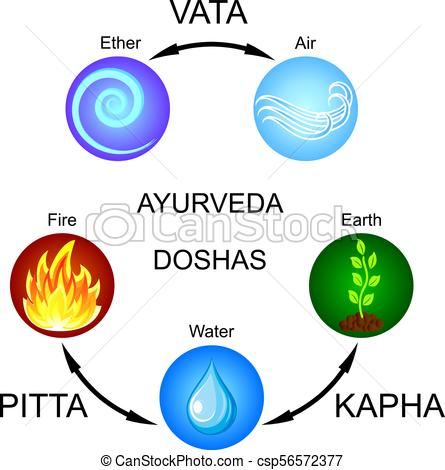
Ayurveda medical system is one of the popular medical systems followed in India and other South Asian countries along with Allopathy, Homeopathy and Naturopathy. Ayurveda was recorded more than 5,000 years ago in Sanskrit, in the four sacred texts called the Vedas: the Rig Veda (3000-2500 BCE), Yajur Veda, Sama Veda and Atharva Veda (1200-1000 BCE). The word 'Ayurveda' is Sanskrit, meaning knowledge of life and longevity. Literally, the word Ayurveda is a combination of two words: **Ayuh** + **Veda**. **Ayu** means life and **Veda** means the science. It is a science of health based on universal principles and profound insights into the connection between mind and body and the laws of nature which structure all progress in life.

Ayurveda therapies have varied and evolved over more than two millennia. Therapies are typically based on complex herbal compounds, minerals and metal substances. It has an accumulated knowledge of disease diagnosis and treatment formulated by several Rishis (knowledgeable persons) through thousands of years. As with other medicine systems, it has its own understanding of body functions and the imbalances which cause diseases. Physical observation of body parts of the patient and changes in them during the course of treatment is an important way of diagnosing the disease and health condition of the patient.

Ayurveda names 7 basic tissues (dhatu) which are plasma(rasa), blood(rakta), muscles(mamsa), fat(meda), bone(asthi), marrow(majja) and semen***(***shukra). Ayurvedic philosophy maintains that people are born with a specific constitution, which is called the **prakruti**. The prakruti, established at conception, is viewed as a unique combination of physical and psychological characteristics that affect the way each person functions. Throughout life, an individual's underlying prakruti remains the same. However, one's prakruti is constantly influenced by various internal, external and environmental factors like day and night, seasonal changes, diet, lifestyle choices, and more.

Like the medicine of classical antiquity, Ayurveda has historically divided bodily substances into five classical elements (Sanskrit) *panchamahabhuta,* i.e. earth, water, fire, air and space. These elements combine in the body and form the three energies called as doshas.

These three energies are responsible to maintain the physical and mental health. Three energies are vata (wind), pitta (fire) and kapha (earth+ water). These energies are important to maintain the good health. Imbalance of any of three energies results in the health complications. Ayurveda nidan (diagnosis) methods study about physical, physiological, psychic, and behavioral aspect of the patient. Each dosha has both positive and negative qualities and is affected by food, environment, lifestyle and movement practices.



**Fig 1.1: Five Elements**

**1.2 DIAGNOSIS**

In the West, the term diagnosis generally refers to identification of the disease after it has manifested. However, in Ayurveda, the concept of diagnosis implies a moment-to-moment monitoring of the interactions between order (health) and disorder (disease) in the body. Ayurveda has eight ways to diagnose illness, called Nadi(pulse), Mootra(urine),

Mala(stool), Jihva(tongue), Shabda(speech), Sparsha(touch), Druk(vision) and Aakruti(appearance).

**Fig 1.2: Ayurvedic Diagnosis**

# 1.2.1 PULSE DIAGNOSIS (NADI)

One of the important diagnostic techniques of the ancient science of Ayurveda is 'Nadi vijñan' or **Pulse diagnosis**. Nadi or pulse is that vital flow of energy or life that courses through as a subtle channel all over the body, and enables the vaidya to feel the way the blood spurts from the heart. The position of the index finger denotes the vata dosha. The middle finger denotes the pulse corresponding to the Pitta dosha. When the throbbing of the pulse under the ring finger is most noticeable, it is a sign of Kapha constitution.



**Fig 1.3: Pulse Diagnosis**

**1.2.2 LIP DIAGNOSIS (OSTHA)**

****As with the other features of the body (e.g., tongue, nails, face, eyes), the lips too reflect the health or disease of the various physical organs. One should observe the size, shape, surface, colour and contour of the lips. If they are dry and rough this indicates dehydration or a vata de-arrangement.

**Fig 1.4: Lip Diagnosis**

**1.2.3 NAIL DIAGNOSIS**

According to Ayurveda, the nails are a waste product of the bones. Look at the size, shape, surface and contour of your nails. If the nails are dry, crooked, rough and break easily, vata pitta predominates. When the nails are thick, strong, soft and very shiny with a uniform predominates in the body. If the nails are soft, pink, tender, easily bent and slightly glistening, contour then kapha diagnosis.

****

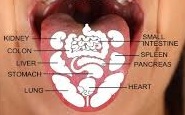
**Fig 1.5: Nail Diagnosis**

**1.2.4 EYE DIAGNOSIS**

Eyes that are small and blink frequently show a predominance of vata in the body. Big, beautiful and attractive eyes indicate a kapha constitution. Pitta eyes are lustrous and sensitive to light, with reddened whites and have a tendency to be nearsighted. According to Ayurveda, the eyes derive their energy from the basic fire element.

**Fig 1.6: Eye Diagnosis **

**1.2.5 TONGUE DIAGNOSIS (JIHVA)**

A discoloration and/or sensitivity of a particular area of the tongue indicates a disorder in the organ corresponding to that area (see diagram). A whitish tongue indicates Kapha derangement and mucus accumulation; a red or yellow-green tongue indicates Pitta derangement; and a black to brown coloration indicates Vata derangement. A dehydrated tongue is symptomatic of a decrease in the dhatu Rasa (plasma), while a pale tongue indicates a decrease in the dhatu Rakta (red blood cells).

**Fig 1.7: Tongue Diagnosis**

**1.3 TONGUE DIAGNOSTIC METHODS AND SIGNIFICANCE IN AYURVEDA**

The tongue is one of the important body parts that can be analyzed to assess the exact nature of digestion, metabolism, affected parts of the body and type of disease. Changes in the tongue can take weeks to years, giving the practitioner information about the depth and length of the illness. The tri-dosha are vata, pitta and kapha. In a patient’s health they are responsible for the function and structure of body. However, if they become excessive or deficient either individually, or in combination, then they cause illness in the patient. Size, shape, contour, surface, margins and colour are the characteristics one can observe on the tongue.

A healthy tongue should look like a children's tongue: symmetric and evenly pink, should be supple, free of cracks and not quiver or tremble. It should be slightly moist. It is neither too thick nor too thin, and oval in shape. When the tongue is sticking out, it's naturally straight rather than curving to one side. It should remain still, not trembling, flaccid, and flat-tipped or stiff. It should have a thin, transparent coating, coloring the tongue pink. All the taste buds are flat, orderly and free from strawberry-looking bumps, deep cuts, lines, cracks and patches. It should not have foam, hair, fur, be too dry or too wet or have a foul odor or taste; a healthy tongue should have some coating.

**Fig 1.8: Healthy Tongue Images**

**1.4 DOSHA’S IN AYURVEDA**

A discoloration and/or sensitivity of a particular area of the tongue indicate a disorder in the organ corresponding to that area. A whitish tongue indicates Kapha derangement and mucus accumulation; a red or yellow-green tongue indicates Pitta derangement; and a black to brown coloration indicates Vata derangement.



**Fig 1.9: Tongue Images with Vata, Pitta and Kapha dosha respectively**

Ether, Air, Fire, Water and Earth, the five basic elements, manifest in the human body as three basic principles, or humors, known as the tridosha. From the Ether and Air elements, the bodily air principal called vata is manifested. (In Sanskrit terminology, this principle is called vata dosha.} The Fire and Water elements manifest together in the body as the fire principle called pitta. The Earth and Water elements manifest as the bodily water humor known as kapha. These three elements vata, pitta, kapha govern all the biological, psychological and physiopathological functions of the body, mind and consciousness. They act as basic constituents and protective barriers for the body in its normal physiological condition; when out of balance, they contribute to disease processes.

A dehydrated tongue is symptomatic of a decrease in the dhatu Rasa (plasma), while a pale tongue indicates a decrease in the dhatu Rakta (red blood cells). The front one-third of the tongue relates to the lungs, heart, chest and neck. Froth in the middle of this area is often a sign of low lung energy with damp and cold lungs and may translate as a cold, bronchitis, asthma, or respiratory allergy. If there is a small depression in the heart area, it often relates to intense grief, sadness, or depression. The central third of the tongue relates to the liver, spleen, stomach, and pancreas. Small ulcers on the tongue in the stomach area may reflect gastritis or peptic ulcer. The rear one-third area of the tongue relates to the lower abdominal organs, such as the small intestine and colon. If this posterior part of the tongue is covered with coating, that is ama or toxins in the colon, indicating low colon energy.

**1.5 SUMMARY**

The main objective of this work is to apply advanced techniques and algorithms of digital image processing and Machine learning to quantify and verify clinical knowledge of tongue colour identification by characterizing variations in tongue features. Tongue images are captured from good quality camera with sufficient lighting conditions, and collected about 260 tongue images. Segmentation algorithm based on edge information is applied on input image to segment the tongue area, and then apply clustering technique using K-means. Clustering approach is applied to separate tongue-body and coating area. The data of the experimental results reveal that our method produces significant useful result i.e it classifies the image into vata, pitta, kapha. We are providing scientific proof for Ayurveda diagnosis by developing an automated diagnosis system for identification of Tri-Dosha.

The region of the tongue image plays a vital role in further classification which makes the segmentation a crucial process. The data acquired are segmented using simple thresholding segmentation method with the average of obtained mean pixel value and maximum repeated intensity value. They are further segmented with ROI (region of interest) segmentation to obtain the exact portion of the tongue. Features are extracted on the segmented image for further classifying into Vata, Pitta, Kapha.

In Chapter 2, we will talk more related to the papers referred to the project. These papers give an insight to the view of diagnosis of various methods which were used for the process of classification. Every paper gave an overview from a different perspective which is considered for the project.

In chapter 3, the methodology of the project is shown which explains an insight into the various techniques used for classification of the model.

Chapter 4 talks about the Software and Hardware requirements which were considered for the development of the project.

Concluding in Chapter 6 with the future work which can enhance the model for betterment of accuracy.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 RECENT TRENDS**

Ayurvedic diagnosis is an ancient technique for the treatment of a patient based on certain features of the human body. There are several features which can be used to namely tongue diagnosis, eye diagnosis, nail diagnosis, lip diagnosis, pulse diagnosis. Each of the diagnosis method has the aim of performing correct treatment method for the patient. The diagnosis method of our interest is tongue. The reference for the same are provided below which gives an insight into various methods which were previously used. These provide a glimpse of how the diagnosis can be performed and how to improve the existing method.

**In[1] Dr Vasant et.al.** has discussed about understanding of tri-dosha with their functionalities. The elements of tri-dosha namely Vata( air + ether), Pitta(fire + water) and Kapha(earth + water) determine the attributes of the dosha’s. Diagnosis, generally referred to identification of diseases can be based on various parts of the human body. Pulse diagnosis, lip diagnosis, facial diagnosis, tongue diagnosis, nail diagnosis and finally the eye diagnosis. For each of the above diagnosis, classification is categorized broadly into the following spectrum of vata, pitta and kapha. The resemblance of either of the tri-dosha namely either of Vata, Pitta, Kapha are prominently seen in each of the diagnosis method which co-relate with one another.

The tongue is the organ of taste and speech. We perceive taste through the tongue when it is wet; a dry tongue cannot perceive taste. The tongue is also the vital organ of speech, used to convey in words, thoughts, concepts, ideas and feelings. Examination of this important organ reveals the totality of what is happening in the body. The different parts of the tongue are related to different organs in the body. If there are discolorations, depressions or elevations on certain areas of the tongue, the respective organs are defective. A coating covering the tongue indicates toxins in the stomach, small intestine or large intestine. If only the posterior part is coated, toxins are present in the large intestine: if the middle of the tongue is coated, toxins are present in the stomach and small intestine.

Treatment of ailment varies for each diagnostic method. The Ayurvedic technique for dealing with negativity is: observation and release. For numerous ailments such as excess mucus in the chest, bile in the intestines, kapha in the stomach or gas accumulation in the large intestine, physical elimination may be used. For such treatment, Ayurveda suggests pancha karma. These processes are cleansing to the body, mind and emotions. Pancha means five and karma means actions or process. The five basic processes are: vomiting, purgatives or laxatives: medicated enemas, nasal administration of medication and purification of the blood.

**In [2] Farid Melgani et.al** has discussedthe relation of obesity in a human as a reference for tongue predictor. Obese patients of BMI (Body mass index) >24 kg/m^2 of either of the sex within the age group of 18-65 were taken into consideration. . It was a cross-sectional study, and 95 patients were enrolled. Participants included were mostly females (𝑛 = 75). All of the obesity patients were divided into two groups of metabolic syndrome group (MetS) group and non- metabolic syndrome group (nMetS) group. Council on Nutrition Appetite Questionnaire (CNAQ) is an 8-item single-domain questionnaire was used to assess the test group based on the 5-Grade scale from A-E. Analysis was performed with SPSS statistical software package, version SPS 22.0. Significant difference between the groups was analyzed using the chi-square of the Fisher’s Extract Test data and Mann-Whitney U test to analyze the differences. The methodology considers the factor of Traditional Chinese (TCM) way of classification of patients. Results of each individual test member was divided based on Gender, BMI, Systolic Pressure, Diastolic Pressure, Waist circumference and Cholesterol.

Based on the data-set in hand in TCM pattern, there was no difference between the two tongue colour groups. However the tongue coating in nMets group was white. Statistical analysis shows that there was no difference between two groups of either gender on heart rate visibility.

In the study, we found the MetS patients having a well-marked white tongue coating more than thin white coating. Otherwise, the nMetS had white coating and yellow coating. In conclusion, obese MetS patients have lower CNQ score and the HRV (Heart rate variability) is not different from nMetS group significantly. However the tongue appearance showed that MetS group has white coating that is different from the nMetS group with white and yellow coating.

**In [3] Satoshi Yamamot et.al** have discussed about identifying the disease oriented state based on visual tongue inspection by a trained physician. Surface colors of uncoated tongue provide essential features. A hyper spectral imaging system which performs regional image analysis of the tongue colour spectrum was developed to automatically inspect the uncoated tongue. Hyper spectral tongue surface images were acquired by a camera equipped with an integrating sphere. The colour spectrum from 4 tongue areas — uncoated tongue, coated tongue, lip, and perioral areas were quantiﬁed.

The average spectrum of each area was determined and 4 images calculated by focusing on the diﬀerences between respective spectra; highlight, shadow, tongue coat, and lip-weighted images were calculated. The uncoated tongue area was extracted by subtraction. In the methods used up to now, segmentation of the whole tongue was performed with 3-dimensional colour spaces. In those methods, image contrast was enhanced by the Sobel operator and the edge of the tongue was traced using various algorithms. Nine hyper spectral images were randomly selected from 44 images and directed to this analysis. For each hyper spectral image, 20 small areas were picked manually, 5 small areas each from 4 tongue areas uncoated tongue area, coated tongue area, lip area and perioral area. Fifty pixels comprised each small area, and the average spectrum was calculated as the representative spectral reﬂectance of the small area. From 9 persons, a total of 45 small areas were picked from each facial area. As the second step, the shadow area was eliminated. This second step was performed with a 700-nm image. As the third step, the coated tongue area was eliminated together with the perioral area. As the fourth step, the lip area was eliminated. The image at 690 nm was selected as a minuend image and at 635 nm as a subtrahend image. Finally, four binary images were processed and eﬀectively combined to obtain the binary uncoated area, and one pixel of opening and closing was performed to remove the dot noise due to thresholding. Tongue area extraction and segmentation have been previously performed using 3-dimensional colour spaces, such as RGB images. In these methods, the tongue could be extracted clearly, although elimination of the coating was incomplete, and it was diﬃcult to analyze the colour properties of the tongue since the colour space was three-dimensional. Tongue area without coating was properly extracted by subtracting other areas, focusing on the spectral diﬀerences among respective facial areas.

**In [4] Jennifer Rioux et.al** has discussed about an automatic tongue diagnosis framework is proposed to analyze tongue images taken by smartphones. Different from conventional tongue diagnosis systems, our input tongue images are usually in low resolution and taken under unknown lighting conditions. Consequently, existing tongue diagnosis methods cannot be directly applied to give accurate results. To overcome the above difficulty they proposed a lighting condition estimation method based on using the SVM (support vector machine) classifier to predict the colour correction matrix according to the colour difference of images taken with and without flash.

The authors also modify the state-of-the-art work of fur and fissure detection for tongue images by taking hue information into consideration and adding a denoising step. When the user faces the back camera, the system automatically captures one image every 0.2second and detects if there are enough tongue colour pixels in the image. The tongue colour distribution is trained by the kernel density estimation technique. Once the user successfully captures a tongue image with proper size/position, the system will upload the image to the cloud server for further analysis. Tongue image colour correction is an important issue in the field of automatic tongue diagnosis. Existing research literature can be classified into four categories, i.e. methods based on simple image statistics, colour temperature curve calibration, supervised learning, and double exposure theory. The authors roughly extract tongue colour pixels of the current image. To more precisely segment the tongue region, Otsu’s method is applied to the hue values of all tongue colour pixels in the image. The Otsu’s threshold can further classify pixels into two clusters, i.e. the real tongue pixels and the noisy pixels.

The authors focus on measuring the colour/ratio of fur and the quantity of fissures in the tongue region, which are important bases for tongue diagnosis. Tongue fur is a substance covering the tongue and usually appears in white, yellow, dark purple or black. Different colour/ratio of tongue fur implies various symptoms of diseases. A normal tongue is relatively flat across the length of the organ. A fissured tongue is marked by a deep groove or fissure in the middle or other areas of the tongue. People may have one or more fissures of varying sizes and depths. Fissures on the tongue can be intuitively detected by edge detection algorithms, such as Canny edge detection. Further improve this approach by applying a 10x10 de-noising window to the tongue region. Only pixels with intensity values smaller than the mean intensity inside the window will be kept. Different from conventional tongue diagnosis systems, the input tongue images are usually in low resolution and taken under unknown lighting conditions.

**In [5] RatchadapornKanawon et.al** have discusseda support decision tool to weight the relative influences of the geometry features, they classified an image into one of six tongue shapes namely hammer, rectangle, acute triangle, obtuse triangle, square and round. The images captured were colour corrected to eliminate any noise caused by variations of illumination and device dependency. Using this device we form a large tongue image database consisting of 672 samples. This database is composed of 130 healthy and 542 disease samples, divided into 7 classes with at least 19 examples. Every image is segmented with the background removed and tongue foreground remaining from each tongue image consisting of a tip, body, and root. They describe the 13 geometry features extracted from tongue images namely width, length, length-width ratio, smaller half distance, center distance, center distance ratio, area, circle area, circle area ratio, square area, square area ratio, triangle area, triangle area ratio. Taking into account the above geometrical features classification is conducted. All the 5 tongue shapes are derived based on the geometric features. To classify the tongue images into proper shape, a decision tree is constructed. Utilizing a grouping of the features is logical, since not every feature can have a positive contribution to the final result, as is the case here where 13 features used for classification produced poor results. Therefore, an optimization of the features is necessary. SFS (Sequential forward selection) is a feature selection method that begins with an empty set of features. It adds additional features based on maximizing some criterion function 𝐽 and terminates when all features have been added. In our case 𝐽 is the average accuracy of SVM. In the experimental results, coarse classification first showed that tongue classes belong to different shapes. Although more than one class occupies the same shape, in fine level classification they vary are still distinguishable, when employing SFS with SVM. With tongue shape and a persons’ health state now established using computer-based methods, this potentially provides a new painless and efficient way to diagnose patients. A continuation of this work will investigate the fusion of all possible tongue features including colour and texture in order to better determine patient’s state.

**In [6] Yakoub Bazi et.al** have discussed aboutanexperimental study to show the superiority of the generalization capability of the support vector machine (SVM) approach in the automatic classiﬁcation of electrocardiogram (ECG) beats. Second, we propose a novel classiﬁcation system based on particle swarm optimization (PSO) to improve the generalization performance of the SVM classiﬁer. The research in the ﬁeld of automatic ECG classiﬁcation has reached a good level of maturation.

However, in the design of an ECG classiﬁcation system, there are still some open issues, which, if suitably addressed, may lead to the development of more robust and efﬁcient classiﬁers. They present a thorough experimental exploration of the SVM capabilities for ECG classiﬁcation. In a second step, we propose to optimize further the performances of the SVM approach in terms of classiﬁcation accuracy: 1) by automatically detecting the best discriminating features from the whole considered feature space and 2) by solving the model selection issue. SVMs are intrinsically binary classiﬁers. But, the classiﬁcation of ECG signals often involves the simultaneous discrimination of numerous information classes.

The authors applied the SVM classifier directly on the entire original hyperdimentional feature space, which is made up of 303 features. During the training phase, the SVM parameters were selected according to a m-fold cross-validation(CV) procedure, first by randomly splitting the 500 training beats into m mutually exclusive subsets(folds) of equal size, and then, by training m times an SVM classifier model with predefined values. The authors can strongly recommend the use of the SVM approach for classifying ECG signals on account of their superior generalization capability as compared to traditional classification techniques. This capability generally provides them with higher classification accuracies and a lower sensitivity to the cure of dimensionality. This can be used for analysis of the images with SVM (support vector machines).

**In [7] Vrinda Hitendra Kurande et.al** have discussed about thediagnostic method often relies on some degree of subjective interpretation by physicians. If the physicians cannot agree on the interpretation, the results will be of little use. The scope of this paper is limited to inter and intra rater reliability for specific diagnostic methods. Reliability denotes consistency of a measure. Validity and reliability concepts can easily be misunderstood. Validity is analogs to accuracy. A test/instrument is valid when it measures, what it is intended to measure. The test is reliable when it produces same results under identical conditions.

Thus, reliability does not denote validity. There are several types of reliability estimates namely intra-rate reliability, inter-rate reliability, inter-method reliability. Internal consistency reliability is used to assess the consistency of results across items on the same test or questionnaire. These reliability studies are linked to Ayurveda by the subject being exposed to 2 different Ayurvedic practitioners at certain time interval. The so obtained results are compared and concluded. The results of the clinical trials conducted on many herbs and formulations could be improved by incorporating classical principles of Ayurveda diagnosis.

In Ayurveda, prakriti based prescription helps in enhancing the therapeutic effect as well as reduces the unwanted effects of the drug. For better results, it is important to include prakriti assessment in the clinical trial as inclusion/ exclusion criteria. In another study, the agreement between raters clinical rationale was assessed for Ayurveda diagnosis. Overall agreement on diagnoses ranged between 60% and 100% and was higher for vikriti. Ayurvedic pulse diagnosis is the unique and non-invasive diagnostic method that determines the state of dosha; however, this is only justifiable if pulse diagnosis yields a consistent result. A study on tongue diagnosis was conducted by 30 TCM practitioners. It is reported that the low level of reliability was due to inadequate operational definitions of both the tongue characteristics studied and of the inspection regions of the tongue.

**2.2 REMARKS ON RECENT TRENDS**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial Number | Title | Findings | Shortcomings |
| 1 | Ayurveda: The Science of self-healing by Dr Vasant Lad | Overview of tri-dosha | No algorithmic approach |
| 2 | Traditional Chinese Medicine for Metabolic Syndrome via TCM Pattern differentiation | Obesity relation to tongue using TCM | No direct correlation between the intended features |
| 3 | Regional Image Analysis of the Tongue colour Spectrum | Spectral analysis of the tongue features | Spectral values can be improved with better techniques |
| 4 | Significant Geometry Features in Tongue Image Analysis | Analysis of tongue based on the shape | Inclusion of various shapes leading to more error in accuracy due to less number of test examples of a particular shape. |
| 5 | Colour Correction Parameter Estimation on the Smartphone and Its Application for Automatic Tongue Diagnosis | Intensity value dependence on the diagnosis method | The intensity variation may hamper the image |
| 6 | Classiﬁcation of Electrocardiogram Signals With Support Vector Machines and Particle Swarm Optimization | SVM classification of the colour correlation matrix | Particle Swarm may not give the accurate result |
| 7 | Reliability studies of diagnostic methods in Indian traditional Ayurveda medicine: An overview | Indian method of tongue diagnosis overview | No approach to model building and classification |

Paper on the reliable study talks about the insights of the various methods in which the dosha’s are classified and on the features used to classify the same. This provides a way to distinguish the features impacting the feature extraction method. Paper on SVM talks about the color correction matrix and how the particles value varies with electrocardiogram. The paper on shape used a different view-point for model which classifies the values based on the shape of the given image. Traditional Chinese Medicine for Metabolic Syndrome via TCM Pattern differentiation relates human obesity to the change in tongue pattern. This gives an insight which was previously unknown relating the physical characteristics of a person to the tongue diagnosis. The diagnostic methods used of edge detection were gained insights from “Regional Image Analysis of the Tongue colour Spectrum” where the segmentation happens. This method of segmentation was used to effectively remove the unwanted parts from the image and obtain used the tongue region of interest. The latter part of classification is done into 3 classes of vata, pitta and kapha helping the ayurvedic specialist for better treatment.

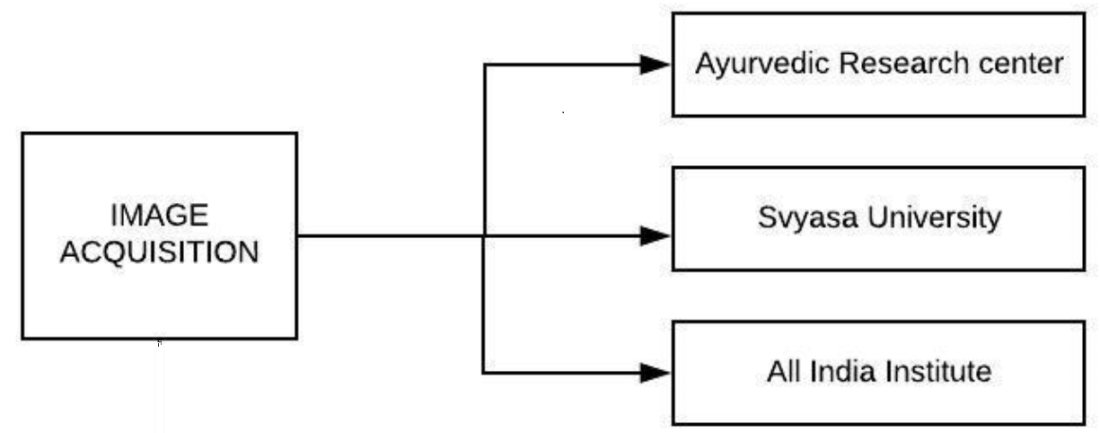
**CHAPTER-3**

**METHODOLOGY**

The section of methodology is defining the work-flow of the project. Any diagnosis method should have a definite workflow which results in the ultimate end of required/desired results. The tongue diagnosis method is initial carried out with the process of image acquisition. Correct images of equal distribution of various dosha’s( namely vata, pitta, kapha) helps in obtaining better accuracy. The dosha are the primary classification categories under which an image falls into. Every successive image would all into one of the 3 dosha’s.

**3.1 IMAGE ACQUISITION**

The process of image acquisition is the primary step of the project. Various Ayurvedic doctor’s were consulted for the process of image processing. A combination of 270 images of various healthy and un-healthy patients was gathered consisting of all the tri-dosha’s namely vata, pitta, kapha. The obtained images were used for the successive modules of the project. The Institutes which helped in the gathering of the images for the processing were SVyasa University, All India Institute of Medical Sciences and Ayurvedic Research Centre. A [flowchart](https://project-management-knowledge.com/definitions/f/flowchart/) is one of the seven basic quality tools used in project management and it displays the actions that are necessary to meet the goals of a particular task in the most practical sequence. Also called as process maps, this type of tool displays a series of steps with branching possibilities that depict one or more inputs and transforms them to outputs.

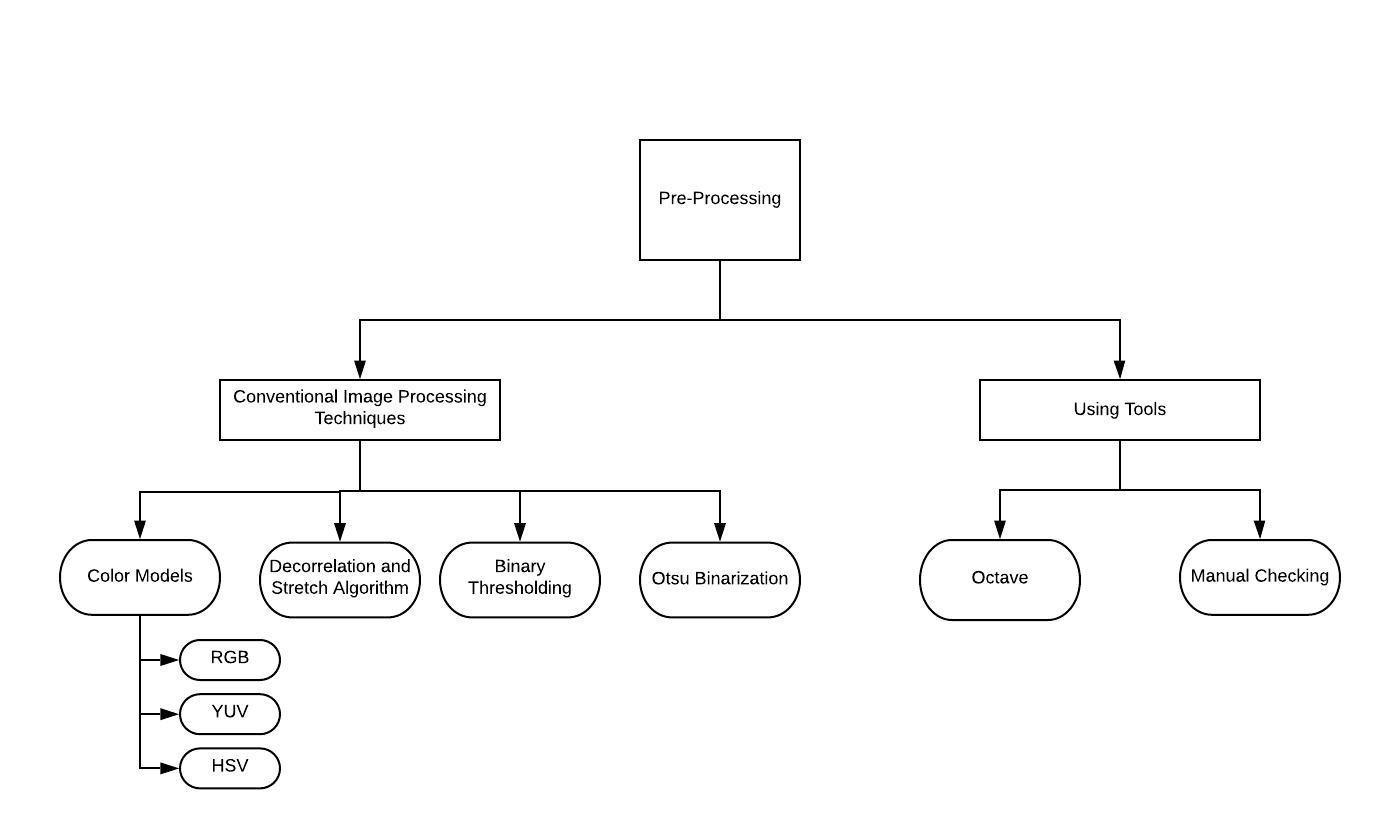
****

**Fig 3.1: Image Acquisition**

**3.2 PRE-PROCESSING**

Pre-processing is carried out using conventional image processing technique and also using tools such Octave. Image processing technique is the way of processing the image to obtain the desired results. This is verified with Octave which is an open-source tool like MatLab.

There were various tried out methods for the method of pre-processing namely colour models, binary thresholding, otsu binarization, de-correlation and stretch algorithm.

****

**Fig 3.2: Pre-Processing**

**3.2.1 COLOUR MODELS**

The colour models were used to differentiate the region of the tongue image based on a single colour. For instance in RGB colour model, we try to differentiate the tongue image into either of the R,G,B colours. There are various colour models which could be used namely RGB, YIQ, HSV.

|  |  |  |  |
| --- | --- | --- | --- |
| SI. No | COLOR SPACES | DESCRIPTION | FORMULAS |
| 1 | RGB | RGB feature represents the normalized intensity value of Red, Green and Blue component respectively. | i = 255 |
| 2 | YIQ | Y refers to the Luminance or Intensity and I/Q channels represents color Information. |  |
| 3 | HSV | HSV stands for Hue, Saturation and Value. This is the Cylindrical system where we separate 3 of the most primary properties of colors and represents them using different channels. |  |

**Fig 3.3: Colour Models**

**3.2.2 DE-CORRELATION AND STRETCH ALGORITHM**

**Input**: the acquired colour image I;

**Output**: three-channel image S.

1. Suppose the height and width of the acquired colour image I are m and n pixels, respectively. The image has three channels: R, G, and B. For each channel, its element is taken column-wise to form a column vector respectively, whose length equals to (m × n). Then this three column vectors are arranged in columns to form (m × n) ×3matrix A.

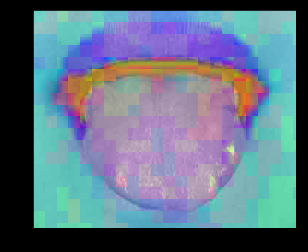
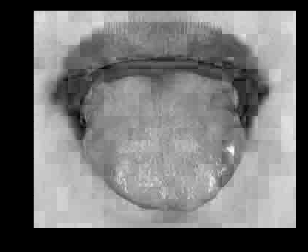
2. Calculate covariance matrix Cov for A and Eigen-value decomposition of Cov will produce matrices of Eigen-values (D) and eigenvectors (V), so that

3. Gain transform matrix T using formula

4. De-correlate matrix A using matrix T by right multiplication

5. Reshape A back to obtain a three-channel image matrix S whose size is the same to the original image I. Each column of A is used to make up one channel of S. To be detailed, elements for each channel of S are fulfilled in columns (to populate the first column firstly) by taking pixels from the corresponding columns of matrix A. In this way, three bands for matrix S are uncorrelated to each other.

6. Enhance contrast for each band of S by adjusting histogram.



**Fig 3.4: De-correlation and Stretch algorithm result**

3.2.3 BINARY THRESHOLDING

From the image obtained using De-Correlation binary thresholding is carried out for the image. It is the simplest thresholding process which separates the region of image corresponding to the object which we want to analyze. The differentiating parameter is the threshold value. A comparison of each pixel value with respect to the threshold value is done classifying into 0(black) or 255 (white) values. Usually background pixels have low values and easily can be removed by applying the heuristic threshold values, θthr which is calculated by the following.

=+)/2

This Binary thresholding operation can be expressed as:

Thresholding Binary inverted operation:

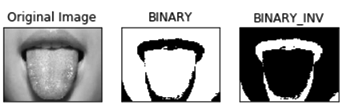


Fig 3.5: **Binary thresholding**

3.2.4 OTSU BINARIZATION

The final step of primary segmentation is Otsu binarization. Otsu binarization, named after Nobuyuki Otsu performs image thresholding. The algorithm uses a single intensity value. It classifies the images into foreground or background based on threshold value, where the sum these values spans a minimal. A white class variance is determined using the weights associated.. Let t to be the threshold value. The threshold subdivides the tongue picture into two classes: c0 and c1. Now calculate the class variance of the segmented classes c0 and c1.

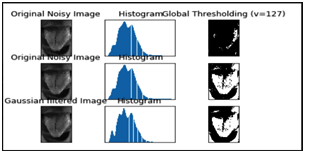


Fig. 3.6 : **Otsu binarization with histogram values**

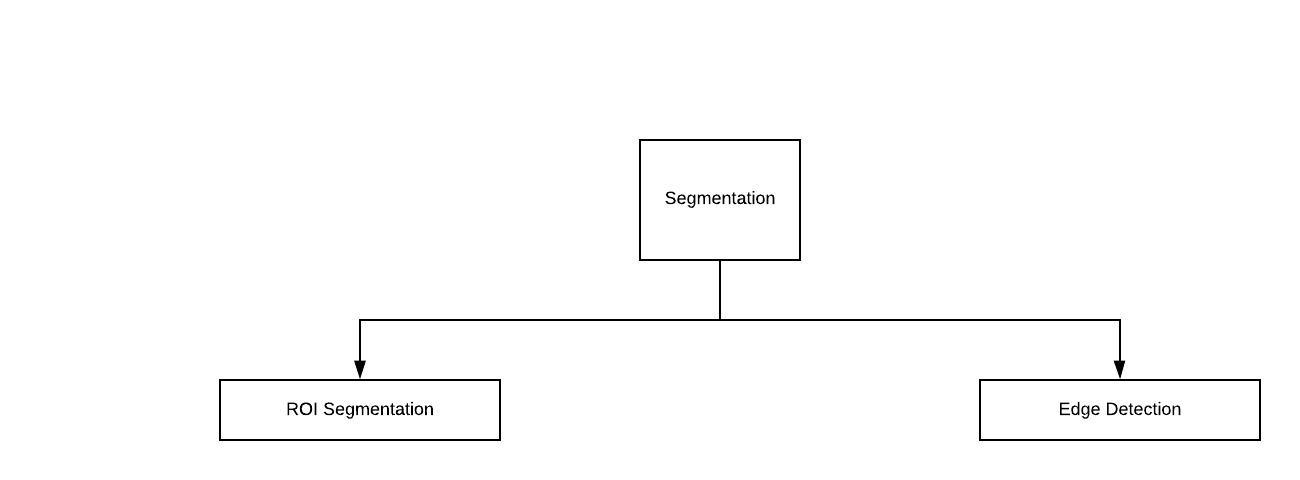
**3.3 SEGMENTATION**

Image acquisition is followed by Primary segmentation where it is divided into 2 categories. Colour clustering is done by converting the image from BGR colour into HSV colour format. The image is vectorized and applying k-means clustering so that the pixels around a colour are consistent and gave same BGR/HSV values K value is checked and was settled for the value K=4. Regenerate the clusters which would have undergone k-means clustering. For the original input image perform canny edge detection to determine the edge region of the tongue. ROI is the region under consideration which gives the region of interest. Edge detection method draws the edge lines of the image. This helps in differentiating the edges of the tongue image from the rest of the image.

The data collection step was accounted with capturing of tongue images of 270 healthy and unhealthy patients. Individual image health record has been taken into account on which the model would be trained and tested. All features of tongue colour, coating, texture, shape is taken as the input to the model. Threshold (binary thresholding, ROI extraction) methods are carried out to extract the region of interest. Classification to be carried out after the segmentation of ROI from tongue image based on required features. The model prediction of Vata Pitta, Kapha is the last step with the test data confirming the grouping of images. In a machine-vision based tongue diagnosis system, background needs to be separate out from tongue image. Usually background pixels have low values and easily can be removed by applying the heuristic threshold values, θ thr which is calculated by the following:

θ thr = (θ MaxRep + θ Median )/2

The acquired tongue image is to output three-channel images. The image has three channels R, G, B to form column vector of length (m × n). Then this three column vectors are arranged in columns of form (m × n) ×3. Calculate the covariance of the matrix. Decorrelate the matrix, reshape back to the three-channel image. Enhance contrast of each band by adjusting histogram.



**Fig 3.7: Segmentation**

**3.3.1** **ROI SEGMENTATION**

ROI segmentation is the primarily technique in secondary segmentation manner. This 'Area of Interest' or ROI generally is decided on the premise of pixel intensity values or person-decided regions (through drawing and subsequent protecting). Segmentation is referred to as manner of isolating objects of interest from uninteresting objects.

Extracting the region of interest(ROI) from original tongue images. The tongue images are normalized with respect to position, orientation, scale, reflection, as follows.

The new invariant coordinates (x, y) of image pixels and the old coordinates (x0,y0) are related by

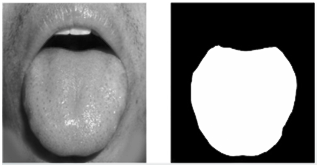
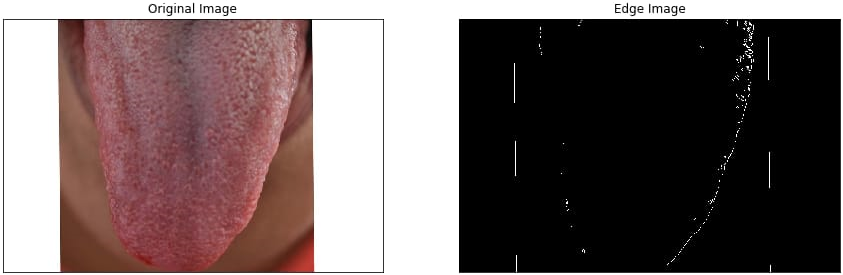


Fig 3.8 : **ROI Segmentation**

A segmented picture based totally on threshold intensity is separated into uninteresting from interesting pixels. Once segmented, the picture pixels may be reassigned the intensity values of either zero or 1. On this operation geometry remains the identical, but the grayscale image is converted right into a binary image along with simplest depth 1 or zero. Figuring out morphometric is a system of counting adjoining pixels with a cost of one and ignoring those with a value of zero. If depth values are required, then the binary photo may be used as a mask**s** to overlay the unique image exposing most effective the roi pixels. As soon as masked, the depth values of the photograph may be obtained.

**3.3.2 EDGE DETECTION**

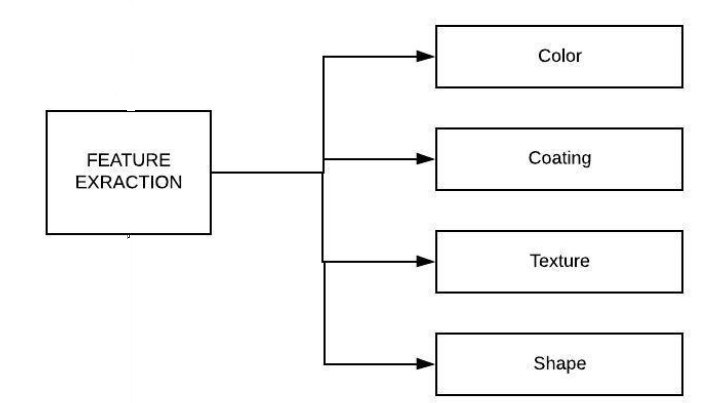
A method that extract beneficial structural records from one-of-a-kind imaginative and prescient objects and dramatically lessen the amount of information to be processed is known as **canny edge detection**. It is been substantially carried out in computer vision systems. Canny has discovered that the requirements for the utility of edge detection on diverse vision structures are exceedingly similar. As a consequence, an edge detection approach to deal with those necessities may be achieved in a extensive kind of situations.



**Fig 3.9: Edge detection result**

**3.4 FEATURE EXTRACTION**

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set. The main features used are: colour, shape, coating and texture.

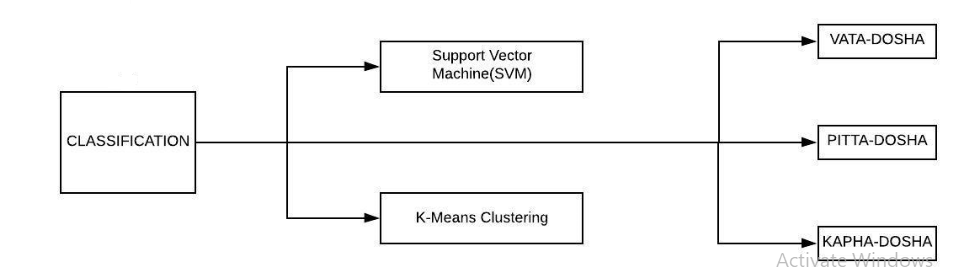
****

**Fig 3.10 : Feature Extraction**

Feature extraction helps in obtaining the correct features for the model using various characteristics such as color, texture, shape and coating. Each of these features taken into account helps in determining the correct classification. Accuracy of the model increases with the increase in the number of features used to obtain the model. Texture of the model helps in identifying the dryness of the tongue. Shape of the tongue helps in determining the classification. These feature extraction help in the better classification of the model.

Colour models refers to a selected manner of organizing colours. A colour model is actually a combination of two things: a colour model and a mapping feature. The motive we want colour models is because it allows us in representing pixel values using tuples. The mapping function maps the colour model to the set of all possible colours that can be represented. There are many different colour models that are useful. RGB, YUV, HSV, Lab are some of the most popular colour models. Different colour models provide exclusive advantages. We simply need to choose the colour model that is right for the given problem. A set of 25 features that span the entire colour models. They can be grouped under eight categories: RGB, HSV, YIQ, YCbCr, XYZ, Lab, CIE Luv, and CMYK

**3.5 CLASSIFICATION**

****

**Fig 3.11: Classification**

**3.5.1 SUPPORT VECTOR MACHINE**

Support Vector Machine (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

**3.5.2 KNN (K NEAREST NEIGHBOUR)**

The k-nearest neighbour is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

* In K-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In K-NN regression, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors.

## The KNN Algorithm

1. Load the data.
2. Initialize K to your chosen number of neighbours.
3. For each example in the data.
4. Calculate the distance between the query example and the current example from the data.
5. Add the distance and the index of the example to an ordered collection.
6. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances.
7. Pick the first K entries from the sorted collection.
8. Get the labels of the selected K entries.
9. If regression, return the mean of the K labels.
10. If classification, return the mode of the K labels.

Classification can be done using f-measure components which includes precision and Recall.

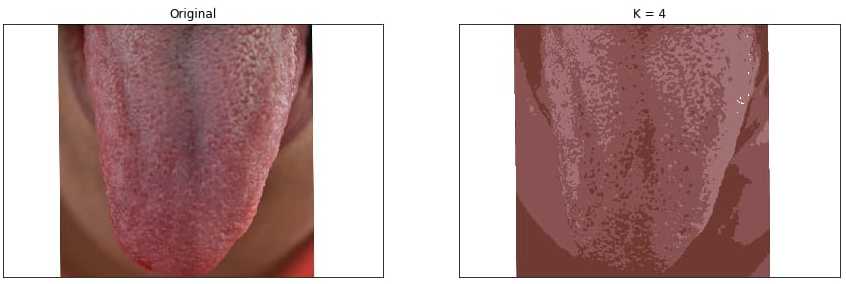
(x, y) is the colour level for the output pixel (x, y) after the contrast stretching process. (x, y) is the colour level input for data the pixel (x, y). (x, y) - is the maximum value for colour level in the input image. (x, y) - is the minimum value for colour level in the input image, γ - constant that defines the shape of the stretching curve.

Further classification is done by using K-Means Clustering Algorithm

**Algorithm determination for K-means:**

1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids ie assignment of data points to clusters isn’t changing.

* Compute the sum of the squared distance between data points and all centroids.
* Assign each data point to the closest cluster (centroid).
* Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.



**Fig 3.12 : Image segmentation using k-means clustering**

**3.6 CHALLENGES FACED**

* Acquisition of images.
* Determination of individual nature of the obtained image namely vata, pitta, kapha during QA (quality assurance).
* Shortlisting features on a broader scale for classification of images.
* Edge detection varied for individual images in python implementation.
* Algorithm for feature extraction.
* Language usage for implementation.

**CHAPTER-4**

**SYSTEM REQUIREMENTS SPECIFICATIONS**

The Objective of this system is to build a detection system platform which help doctors to asses the patient’s medical condition. It based on the ancient ayurvedic technique for analysis. The system build classifies the human tongue into 3 categories. This classification helps the doctor to asses a patient individually based on the nature of vata, pitta and kapha.

The image of analysis is taken from a camera with good light condition. This is then fed into the system where the image undergoes feature extraction and final classification by the model.

**4.1 SOFTWARE REQUIREMENTS**

* **Operating system : Windows 8 / 10**
* **Programming Language : Python, Octave**
* **Framework : Anaconda , MatLab**
* **IDE : Jupyter Notebook, MatLab MathWork**

**4.1.1 PYTHON**

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aims to help programmers write clear, logical code for small and large-scale projects. Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [procedural](https://en.wikipedia.org/wiki/Procedural_programming), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library). Python [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). A global community of programmers develops and maintains [Python](https://en.wikipedia.org/wiki/CPython), an [open source](https://en.wikipedia.org/wiki/Open-source_software) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation). Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by [meta programming](https://en.wikipedia.org/wiki/Metaprogramming) and [meta objects](https://en.wikipedia.org/wiki/Metaobject) (magic methods)). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming). Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing), and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management). It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution. Rather than having all of its functionality built into its core, Python was designed to be highly [extensible](https://en.wikipedia.org/wiki/Extensibility). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Python strives for a simpler, less-cluttered syntax and grammar while

giving developers a choice in their coding methodology. Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases. Python uses [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation, rather than [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) or keywords, to delimit [blocks](https://en.wikipedia.org/wiki/Block_(programming)). An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.

Thus, the program's visual structure accurately represents the program's semantic structure. This feature is also sometimes termed the [off-side rule](https://en.wikipedia.org/wiki/Off-side_rule). Python's large [standard library](https://en.wikipedia.org/wiki/Standard_library), commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as [MIME](https://en.wikipedia.org/wiki/MIME) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) are supported. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary precision decimals, manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing).

**4.1.2 PACKAGES**

**4.1.2.1 NUMPY**

**NumPy is a library for the** [**Python programming language**](https://en.wikipedia.org/wiki/Python_(programming_language))**, adding support for large, multi-dimensional** [**arrays**](https://en.wikipedia.org/wiki/Array_data_structure) **and** [**matrices**](https://en.wikipedia.org/wiki/Matrix_(math))**, along with a large collection of** [**high-level**](https://en.wikipedia.org/wiki/High-level_programming_language)[**mathematical**](https://en.wikipedia.org/wiki/Mathematics)[**functions**](https://en.wikipedia.org/wiki/Function_(mathematics)) **to operate on these arrays. NumPy is** [**open-source software**](https://en.wikipedia.org/wiki/Open-source_software) **and has many contributors. The core functionality of NumPy is its "ndarray", for *n*-dimensional array, data structure. These arrays are** [**stride**](https://en.wikipedia.org/wiki/Stride_of_an_array) **views on memory. In contrast to Python's built-in list data structure (which, despite the name, is a** [**dynamic array**](https://en.wikipedia.org/wiki/Dynamic_array)**), these arrays are homogeneously typed: all elements of a single array must be of the same type. NumPy has built-in support for** [**memory-mapped**](https://en.wikipedia.org/wiki/Memory-mapped_file) **ndarrays. A new package called *Numarray* was written as a more flexible replacement for Numeric. Like Numeric, it is now deprecated. Numarray had faster operations for large arrays, but was slower than Numeric on small ones, so for a time both packages were used for different use cases. We can import numpy as follows:**

**>>> import numpy as np**

**4.1.2.2 SciPy**

SciPy is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) library used for [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing) and technical computing. SciPy contains modules for [optimization](https://en.wikipedia.org/wiki/Optimization_(mathematics)), [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra), [integration](https://en.wikipedia.org/wiki/Integral), [interpolation](https://en.wikipedia.org/wiki/Interpolation), [special functions](https://en.wikipedia.org/wiki/Special_functions), [FFT](https://en.wikipedia.org/wiki/Fast_Fourier_transform), [signal](https://en.wikipedia.org/wiki/Signal_processing) and [image processing](https://en.wikipedia.org/wiki/Image_processing), [ODE](https://en.wikipedia.org/wiki/Ordinary_differential_equation) solvers and other tasks common in science and engineering. SciPy builds on the [NumPy](https://en.wikipedia.org/wiki/NumPy) array object and is part of the NumPy stack which includes tools like [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib), [pandas](https://en.wikipedia.org/wiki/Pandas_(software)) and [SymPy](https://en.wikipedia.org/wiki/SymPy), and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as [MATLAB](https://en.wikipedia.org/wiki/MATLAB), [GNU Octave](https://en.wikipedia.org/wiki/GNU_Octave), and [Scilab](https://en.wikipedia.org/wiki/Scilab). The NumPy stack is also sometimes referred to as the SciPy stack. The basic data structure used by SciPy is a multidimensional [array](https://en.wikipedia.org/wiki/Array_data_structure) provided by the [NumPy](https://en.wikipedia.org/wiki/NumPy) module. NumPy provides some functions for linear algebra, [Fourier transforms](https://en.wikipedia.org/wiki/Fourier_transform), and [random number generation](https://en.wikipedia.org/wiki/Random_number_generation), but not with the generality of the equivalent functions in SciPy. NumPy can also be used as an efficient multidimensional container of data with arbitrary datatypes. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. We import scipy as follows:

**>>> from scipy import stats**

**4.1.2.3 PANDAS**

When you want to use Pandas for data analysis, you’ll usually use it in one of three different ways:

* Convert a Python’s list, dictionary or Numpy array to a Pandas data frame
* Open a local file using Pandas, usually a CSV file, but could also be a delimited text file (like TSV), Excel, etc.
* Open a remote file or database like a CSV or a JSON on a website through a URL or read from a SQL table/database

There are different commands to each of these options.

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary.

The last set of basic Pandas commands are for joining or combining data frames or rows/columns.

**4.1.3MATLAB**  
Millions of engineers and scientists worldwide use MATLAB® to analyze and design the systems and products transforming our world. The matrix-based MATLAB language is the world’s most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

MATLAB helps you take your ideas beyond the desktop. You can run your analyses on larger data sets, and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise, and production systems.

The MATLAB application is built around the MATLAB programming language. Common usage of the MATLAB application involves using the "Command Window" as an interactive mathematical [shell](https://en.wikipedia.org/wiki/Command_line_interface) or executing text files containing MATLAB code.

### Structures

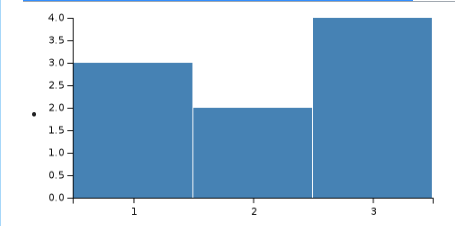
MATLAB supports structure data types.[[28]](https://en.wikipedia.org/wiki/MATLAB#cite_note-28) Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field name.

### Functions

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. Variables and functions are case sensitive.

### Variables

Variables are defined using the assignment operator, =. MATLAB is a [weakly typed](https://en.wikipedia.org/wiki/Strong_and_weak_typing) programming language because types are implicitly converted. It is an inferred typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects and that their type can change. Values can come from [constants](https://en.wikipedia.org/wiki/Constant_(computer_science)), from computation involving values of other variables, or from the output of a function.



## Fig 4.1 : Graphics and graphical user interface programming

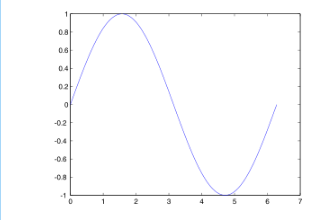
MATLAB has tightly integrated graph-plotting features. For example, the function *plot* can be used to produce a graph from two vectors *x* and *y*. The code:

x = 0:pi/100:2\*pi;

y = sin(x);

plot(x,y)

produces the following figure of the [sine function](https://en.wikipedia.org/wiki/Sine_wave):



MATLAB supports three-dimensional graphics as well:

|  |  |  |
| --- | --- | --- |
| [X,Y] = meshgrid(-10:0.25:10,-10:0.25:10);  f = sinc(sqrt((X/pi).^2+(Y/pi).^2));  mesh(X,Y,f);  axis([-10 10 -10 10 -0.3 1])  xlabel('{\bfx}')  ylabel('{\bfy}')  zlabel('{\bfsinc} ({\bfR})')  hidden off |  | [X,Y] = meshgrid(-10:0.25:10,-10:0.25:10);  f = sinc(sqrt((X/pi).^2+(Y/pi).^2));  surf(X,Y,f);  axis([-10 10 -10 10 -0.3 1])  xlabel('{\bfx}')  ylabel('{\bfy}')  zlabel('{\bfsinc} ({\bfR})') |

## 

## Interfacing with other languages

MATLAB can call functions and subroutines written in the programming languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) or [Fortran](https://en.wikipedia.org/wiki/Fortran). A wrapper function is created allowing MATLAB data types to be passed and returned. [MEX files](https://en.wikipedia.org/wiki/MEX_file) (MATLAB executables) are the dynamically loadable object files created by compiling such functions. Since 2014 increasing two-way interfacing with [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) was being added.

Libraries written in [Perl](https://en.wikipedia.org/wiki/Perl), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [ActiveX](https://en.wikipedia.org/wiki/ActiveX) or [.NET](https://en.wikipedia.org/wiki/.NET_Framework) can be directly called from MATLAB and many MATLAB libraries (for example [XML](https://en.wikipedia.org/wiki/XML) or [SQL](https://en.wikipedia.org/wiki/SQL) support) are implemented as wrappers around Java or ActiveX libraries. Calling MATLAB from Java is more complicated, but can be done with a MATLAB toolbox which is sold separately by [MathWorks](https://en.wikipedia.org/wiki/MathWorks), or using an undocumented mechanism called JMI (Java-to-MATLAB Interface),(which should not be confused with the unrelated [Java Metadata Interface](https://en.wikipedia.org/wiki/Java_Metadata_Interface) that is also called JMI). Official MATLAB API for Java was added in 2016.

As alternatives to the [MuPAD](https://en.wikipedia.org/wiki/MuPAD) based Symbolic Math Toolbox available from MathWorks, MATLAB can be connected to [Maple](https://en.wikipedia.org/wiki/Maple_(software)) or [Mathematica](https://en.wikipedia.org/wiki/Mathematica). Libraries also exist to import and export [MathML](https://en.wikipedia.org/wiki/MathML).

**4.2 HARDWARE REQUIREMENTS**

* **System Processor : Core i3 / i5**
* **Hard Disk : 500 GB.**
* **Ram : 8 GB.**
* **Rom : 1TB.**
* **Any desktop / Laptop system with above configuration or higher level.**

**4.2.1 CAMERA**

|  |  |
| --- | --- |
| Type | Single-lens reflex digital camera |
| Lens mount | Nikon F mount (with AF contacts) |
| Effective angle of view | Approx. 1.5x lens focal length (35 mm format equivalent); Nikon DX format |
| Effective pixels | |
| Effective pixels | 24.2 million |
| Image sensor | |
| Image sensor | 23.2 x 15.4 mm CMOS sensor |
| Total pixels | 24.7 million |
| Dust-reduction System | Image sensor cleaning Image Dust Off reference data (optional Capture NX 2 software required) |
| Storage | |
| Image size (pixels) | 6,016 x 4,000 [L], 4,512 x 3,000 [M], 3,008 x 2,000 [S] |
| File format | * NEF (RAW): 12 bit, compressed * JPEG: JPEG-Baseline compliant with fine (approx. 1:4), normal (approx. 1:8) or basic (approx. 1:16) compression * NEF (RAW)+JPEG: Single photograph recorded in both NEF (RAW) and JPEG formats |
| Picture Control System | Standard, Neutral, Vivid, Monochrome, Portrait, Landscape; selected Picture Control can be modified |

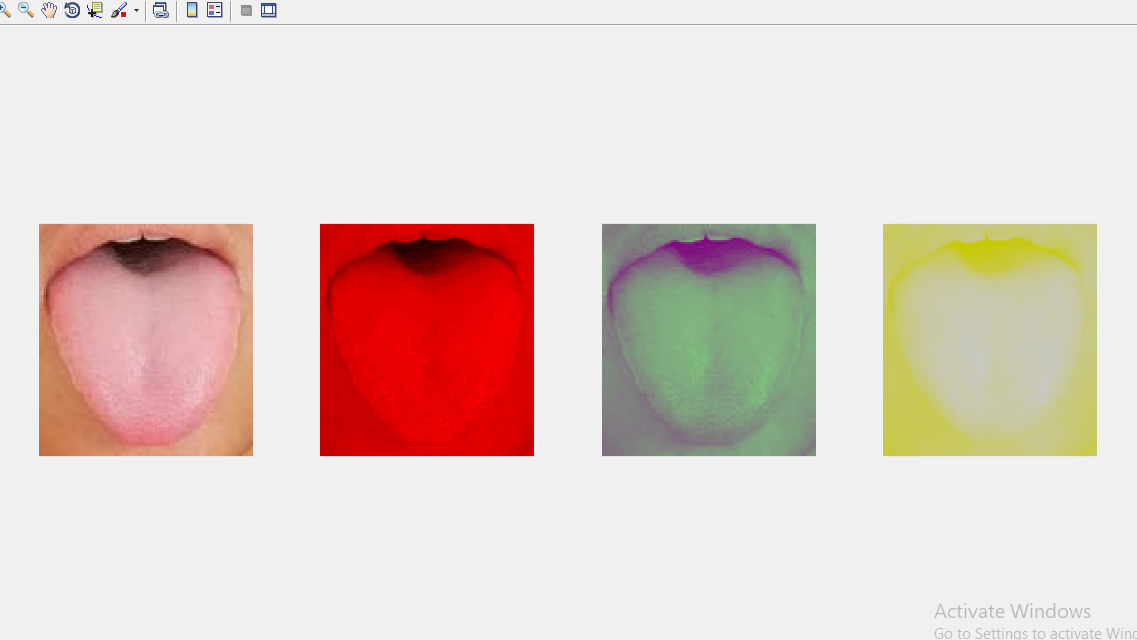
* 1. **FUNCTIONAL REQUIREMENTS**
* The minimum requirement for the doctor is to know how to operate the user interface.
* The image should be captured with a camera of resolution of atleast 15 mega pixels.
* The obtained image should be visible under enough light conditions to obtain a high accuracy of image.
* The image should be before-hand captured for swift access of the system.
* The obtained data should co-relate with the expertise of the ayurvedic doctor.

**4.4 NON-FUNCTIONAL REQUIREMENTS**

* **Accuracy:** The nature of the shown image should peer with the analysis of the practitioner.
* **Acceptability:** The model of the scheme is kept adequately achievable, so that there should not be any issue.
* **Efficiency:** The model should work with all set of valid images provided to obtain broader level of support practitioner.
* **Portability:** The application is developed in Matlab. It is compatible to other programming environments reducing the need for a single development environment to run the applications.

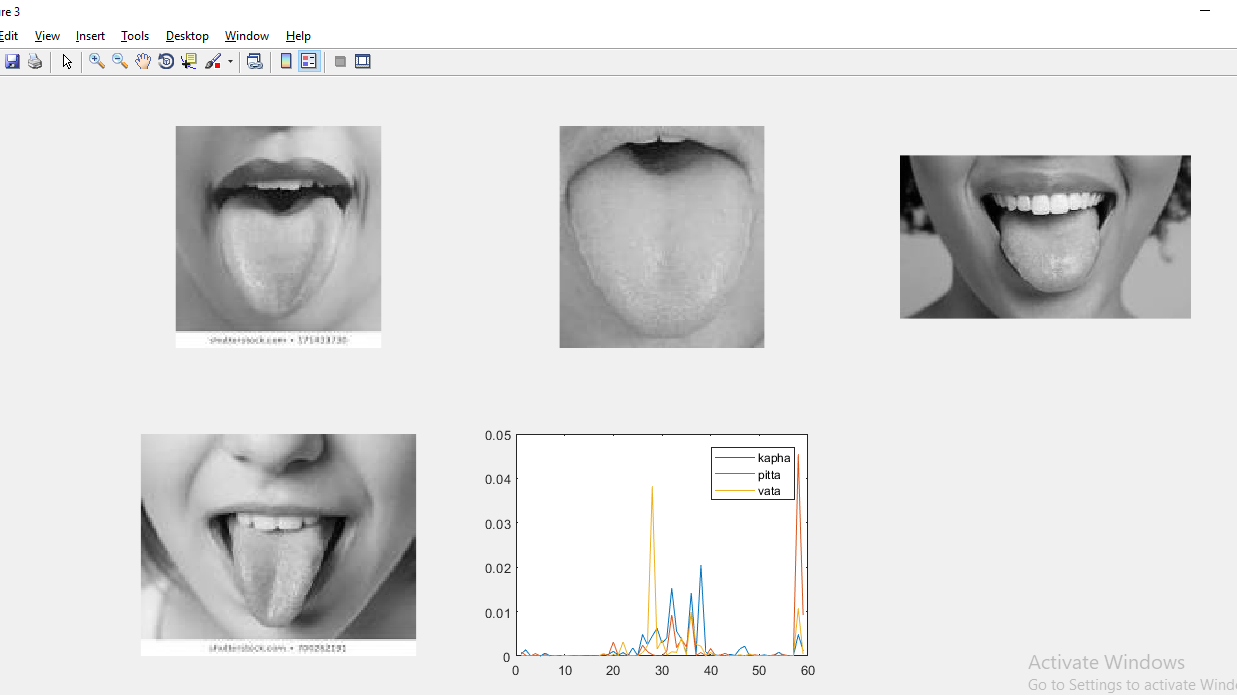
**CHAPTER-5**

**SNAPSHOTS AND RESULTS**

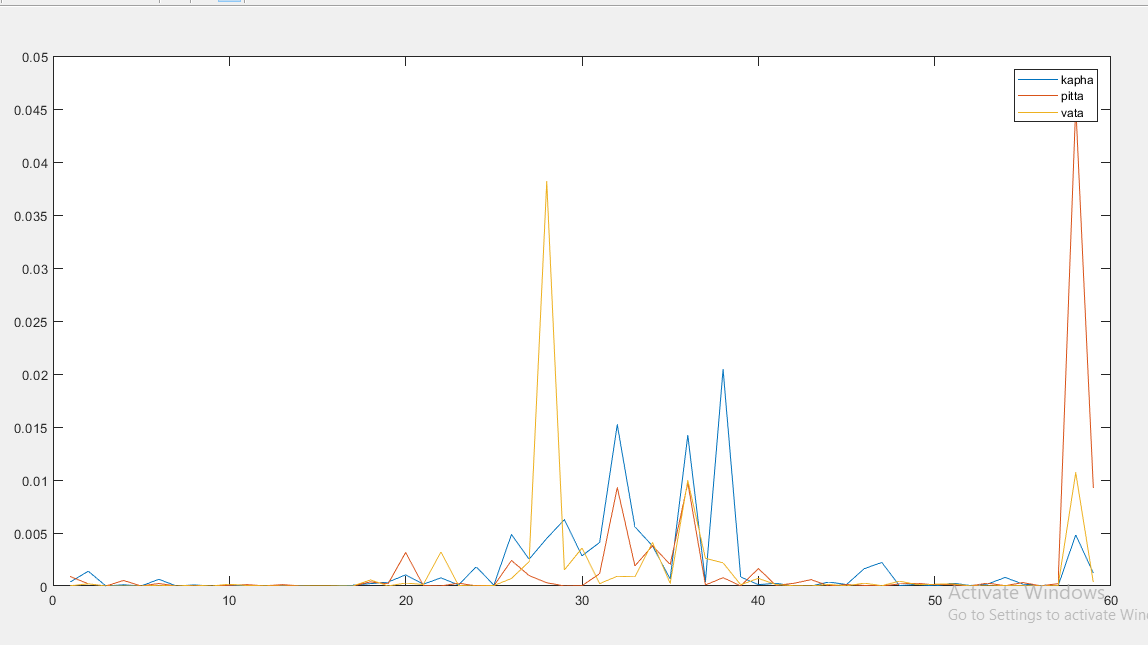
****

**Figure 5.1: Segmented RGB Component of the image**

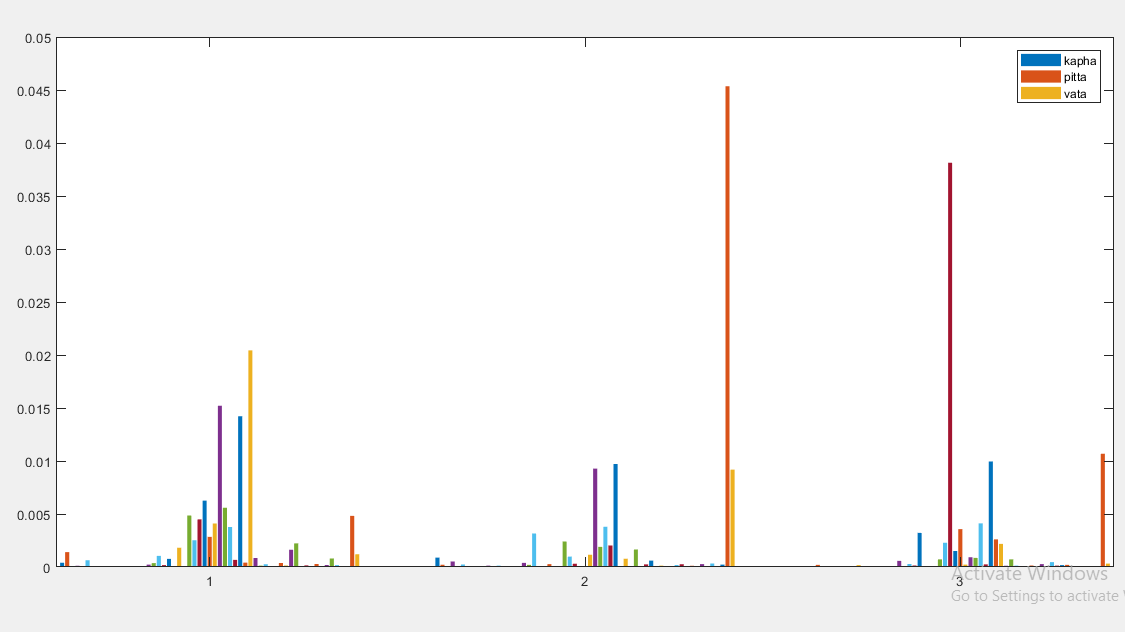
The color composition of RGB color format representation is shown above. It plays a crucial role in extracting the required region of interest for further processing. After initial segmentation, the process of feature extraction is of importance and the results depend on the segmentation.

****

**Figure 5.2: Least Mean Square Error compensated graph**

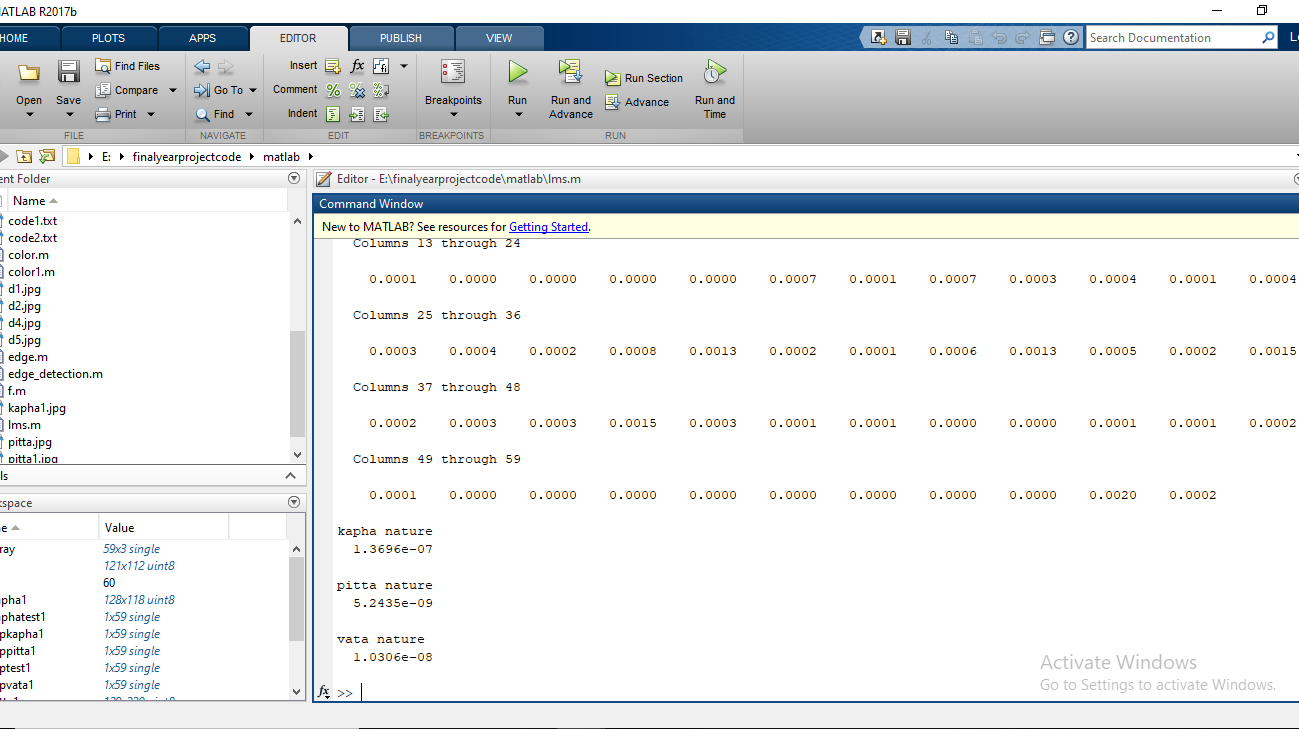
****

**Figure 5.3: Least Mean Square Error Compensated Graph**

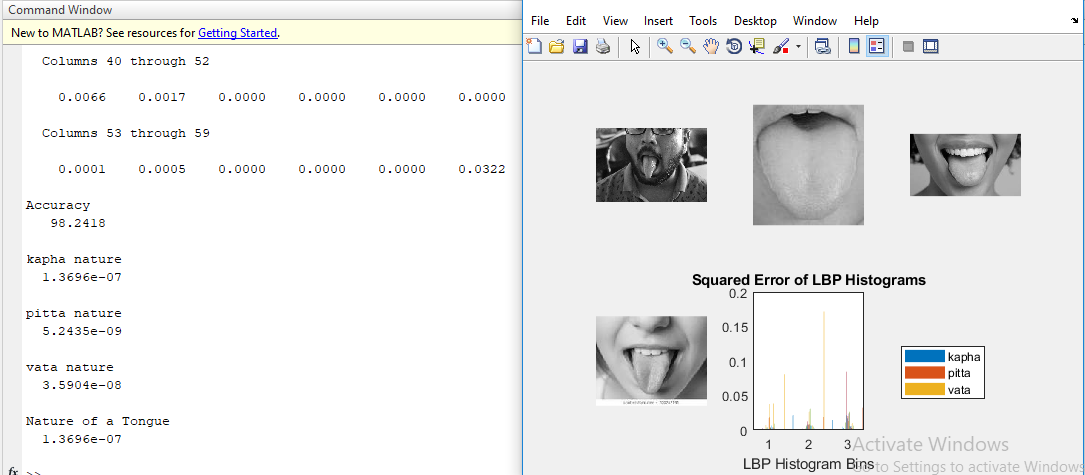
****

**Fig 5.4: Least Mean Square Error of LBP Histograms**

The above graphs, gauge the similarity between the LBP features by computing the squared error between them. The squared error is smaller when images have similar texture.

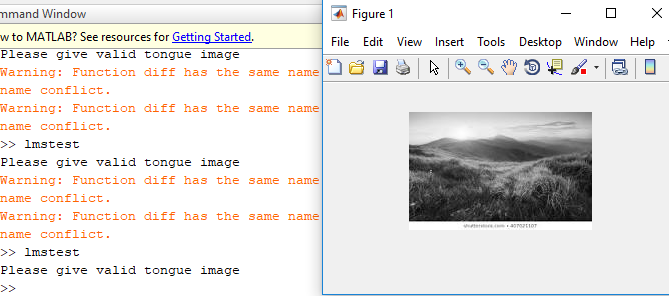
****

**Fig 5.5: Classification Result**

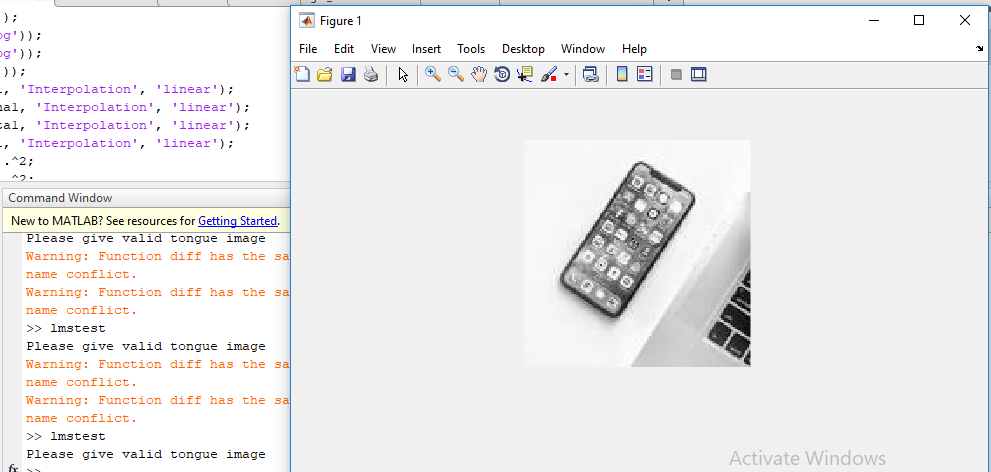
****

**Fig 5.6: Accuracy result**

The image depicts the accuracy based on the input image .The nature of the shown image should peer with the analysis of the practitioner.

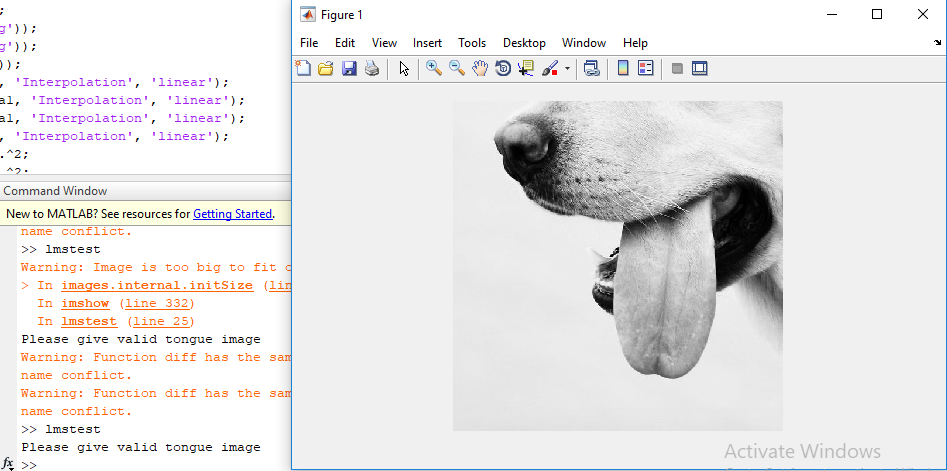
****

**Fig 5.7(a) : Unit Testing**

****

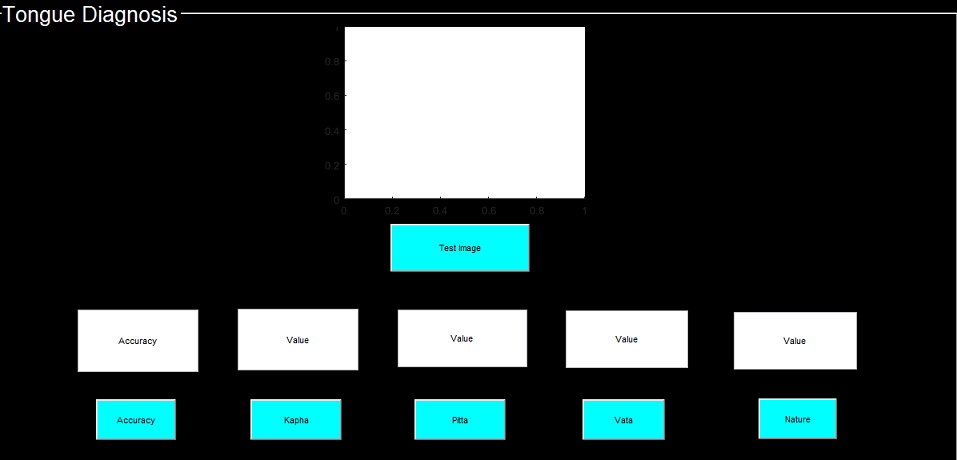
**Fig 5.7(b) : Unit Testing**

The interest of output is restricted to human tongue for classification based on the nature. In the above images phone and nature image is taken as an input. The obtained result signifies an important part of unit testing for anomaly.

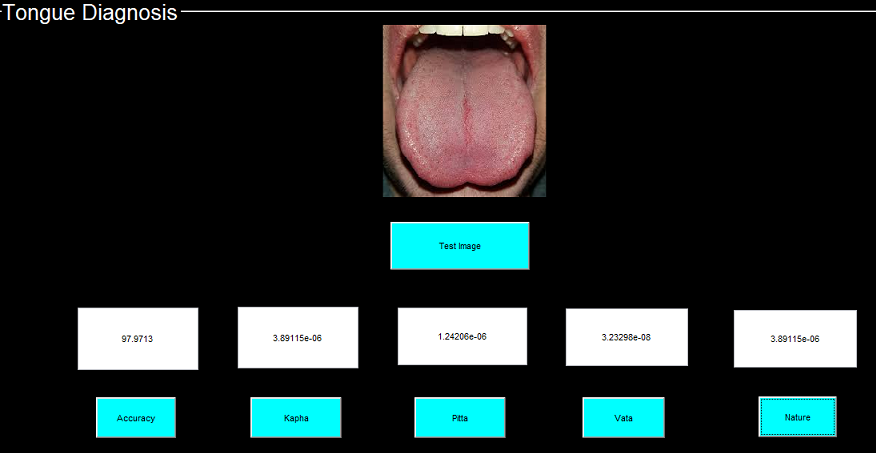
****

**Fig 5.8 : White box testing**

The interest of output is restricted to human tongue for classification based on the nature. In the above image non-human tongue image is taken as an input. The obtained result signifies an important part of white box testing for anomaly.

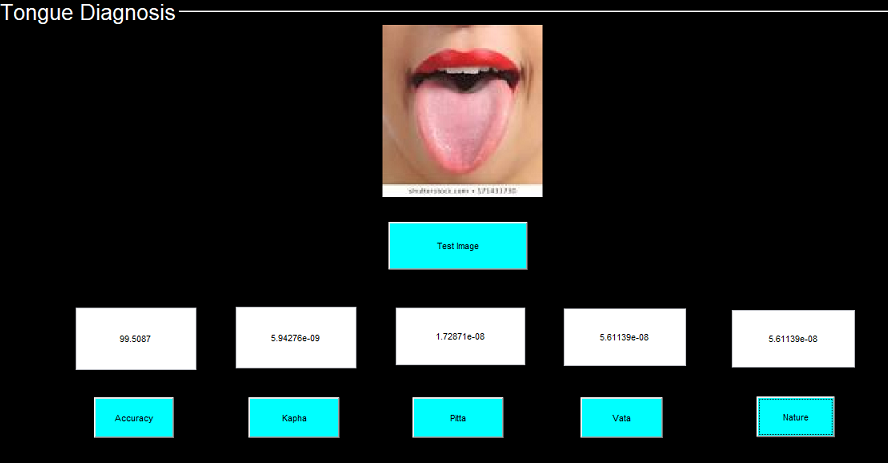


**Fig 5.9: Outline of Graphical User Interface**

****

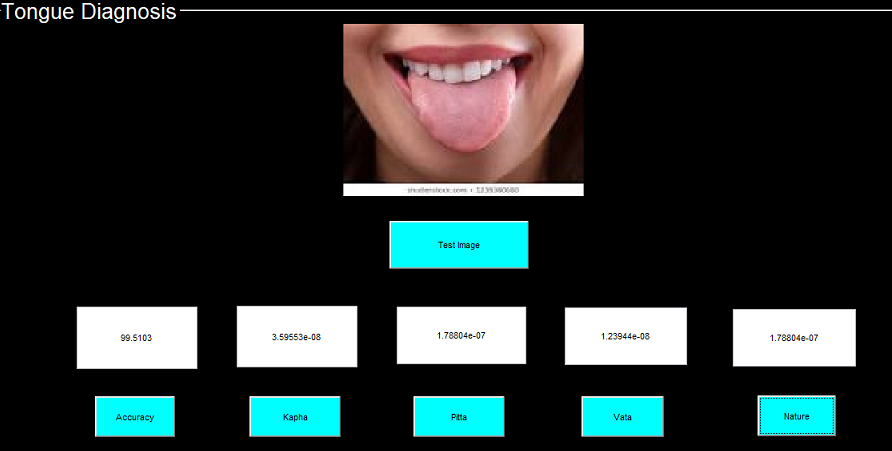
**Fig 5.10(a): Kapha dosha**

The accuracy obtained for the above test image is 97.9% and the nature is Kapha dosha.

****

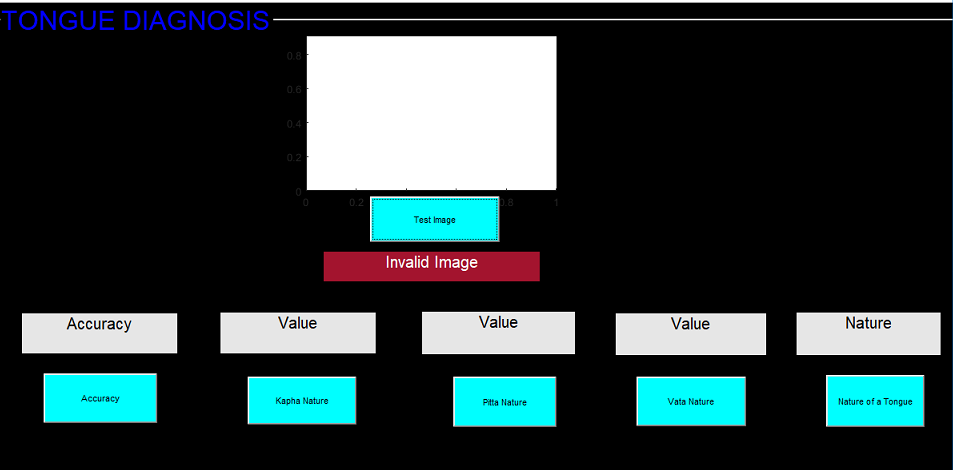
**Fig 5.10(b): Vata dosha**

The accuracy obtained for the above test image is 99.5% and the nature is Vata dosha.

****

**Fig 5.10(c): Pitta dosha**

The accuracy obtained for the above test image is 99.5% and the nature is Pitta dosha.

****

**Fig 5.11:Invalid image**

**CHAPTER-6**

**CONCLUSION**

At the beginning of the project it was decided that we build a model and classify them into tri-dosha’s using image processing models. Further during the course of the project various tools were necessary and was developed using Matlab. Least square mean error was used to distinguish the images into respective classes. We have achieved classification into 3 classes of vata, pitta and kapha.

The final outcome of the research is to blend in with the traditional Ayurvedic methods to the machine learning model in a way it describes the most basic class of inspection in medication.

**REFERENCES**

[1] Dr Vasant Lad” Ayurveda: The science of self-healing”

[2] Jennifer Rioux, Cynthia Thomson, Amy Howerter, “A Pilot Feasibility Study of Whole-systems Ayurvedic Medicine and Yoga therapy for Weight Loss”, USA January 2014

[3] DanMeng, GuitaoCao, YeDuan, MinghuaZhu, LipingTu, DongXu, and JiatuoXu “Tongue Images Classification Based on Constrained High Dispersal Network” Shanghai2012,China.

[4] Farid Melgani, Yakoub Bazi “Classiﬁcation of Electrocardiogram Signals With Support Vector Machines and Particle Swarm Optimization”,2018 IEEE.

[5] BobZhang, HanZhang “Significant Geometry Features in Tongue Image Analysis”, 2015

[6] Satoshi Yamamoto, Norimichi Tsumura, Toshiya Nakaguchi, Takao Namiki, Yuji Kasahara, Katsutoshi Terasawa, Yoichi Miyake “Regional Image Analysis of the Tongue Colour Spectrum”

[7] Vrinda Hitendra Kurande, Rasmus Waagepetersen, Egon Toft, Ramjee Prasad “Reliability studies of diagnostic methods in Indian traditional Ayurveda medicine: An overview”, India, 2015

[8] M. J. Shi, G. Z. Li, F. F. Li, and C. Xu, “A novel tongue segmentation approach utilizing double geodesic flow,” in Proceedings of the 7th International Conference on Computer Science and Education (ICCSE ’12), pp. 21–25, Melbourne, Australia, July 2012.

[9] M. J. Shi, G. Z. Li, and F. F. Li, “C2 G2 FSnake: automatic tongue image segmentation utilizing prior knowledge,” Science China Information Sciences, vol. 56, no. 9, pp. 1–14, 2013.

[10] C. Liang and D. C. Shi, “A prior knowledge-based algorithm for tongue body segmentation,” in Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE ’12), pp. 646–649, Hangzhou, China, March 2012.

[11] X. M. Zhai, H. D. Lu, and L. Z. Zhang, “Application of image segmentation technique in tongue diagnosis,” in Proceedings of the International Forum on Information Technology and Applications (IFITA ’09), pp. 768–771, Chengdu, China, May 2009.

[12] J. F. Ning, D. Zhang, C. K. Wu, and F. Yue, “Automatic tongue image segmentation based on gradient vector flow and region merging,” Neural Computing and Applications, vol. 21, no. 8, pp. 1819–1826, 2012.

[13] S. X. Li, “Automatic segmentation of tongue image based on histogram threshold and snakes model,” Journal of University of Science and Technology Liaoning, vol. 32, no. 6, pp. 585–590, 2009

[14] Z. C. Fu, X. Q. Li, and F. F. Li, “Tongue image segmentation based on snake model and radial edge detection,” Journal of Image and Graphics, vol. 14, no. 4, pp. 688–693, 2009.

[15] N. Paragios, O. Mellina-Gottardo, and V. Ramesh, “Gradient vector flow fast geometric active contours,” IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 3, pp. 402–407, 2004. [22]. N. Paragios, O. Mellina-Gottardo, and V. Ramesh, “Gradient vector flow fast geodesic active contours,” in Proc. of IEEE International Conference on Computer Vision, 2001.

[16]. C. Xu and J. Prince, Handbook of Medical Imaging: Processing and Analysis Management, ch. Gradient Vector Flow Deformable Models. Academic Press, 2000.

[17]. C. Xu, D. Pham, and J. Prince, Handbook of Medical Imaging: Processing and Analysis Management, ch. Image Segmentation Using Deformable Models, pp. 129–174. Academic Press, 2000.