

CLASSIFICATION OF VARIOUS STAGES OF CERVICAL CANCER USING TEXTURE ANALYSIS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in B.Tech Biomedical Engineering

By

POOJAHSRI S (39240029)

PRERANA S (39240030)



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **POOJAHSRI S (39240029)** and **PRERANA S (39240030)** who carried out the final year project entitled **"CLASSIFICATION OF VARIOUS STAGES OF CERVICAL CANCER USING TEXTURE ANALYSIS"** under our supervision from **October 2022** to **April 2023**.

Internal Guide

(Dr . P. GRACE KANMANI PRINCE M.E, Ph.D)

Head of the Department

(Dr. T.SUDHAKAR, M.Sc, Ph.D)

HEAD OF THE DEPARTMENT
DEPARTMENT OF BIOMEDICAL ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Mandayam Nagar, Rajiv Gandhi Salai
Chennai - 600 079

Submitted for Viva voce Examination held on 18th April 2023

Internal Examiner

External Examiner

DECLARATION

We, **POOJAHSRI S (39240029)** and **PRERANA S (39240030)**, hereby declare that the Project Report entitled "**CLASSIFICATION OF VARIOUS STAGES OF CERVICAL CANCER USING TEXTURE ANALYSIS**" done by us under the guidance of **Dr. P. GRACE KANMANI PRINCE** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in B.Tech Biomedical Engineering.

DATE: 18.4.23

1. Poojahsri S

2. Prerana S

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this Project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T . Sudhakar, M.Sc., Ph.D Head of the Department, Department of Biomedical Engineering** for providing us with necessary support and details at the right time during the progressive reviews.

We would also like to express our sincere and deep sense of gratitude to our Project Guide, **Dr. P. Grace Kanmani Prince M.E., Ph.D, Assistant Professor, Department of Biomedical Engineering** for her valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of my project work.

We wish to express our thanks to all **Teaching and Non–Teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways in the completion of the Project.

We would also like to express our gratitude towards our parents for supporting us in the successful completion of our project

ABSTRACT

Cancer is one of the deadliest diseases which threatens human life. Among various types of cancer, Cervical cancer is the leading cause of cancer death in women. Early detection of Cervical cancer is hard to detect because it does not have any external manifestation of the disease. Pap smear is one of the clinical procedures to find cervical cancer but it is not accurate always. Also, it is a very sensitive process and also consumes a lot of time. Various diagnostic techniques like CT, MRI, etc can be used with various imaging techniques to detect cervical cancer. This project aims at analysing MRI images for detection of malignant tissues in cervix. The image processing includes image enhancement and segmentation. The segmented image undergoes textural analysis. The features are extracted based on the textural information and this is fed to a classifier to detect the various stages of cancer.

Keywords— *cervical cancer, deep learning, feature extraction, machine learning, texture analysis*

TABLE OF CONTENTS:

CHAPTER NO	TITLE	PAGE NO
	ABSTARCT	i
	FIGURES	iv
	LIST OF ABBREVIATION	v
1	INTRODUCTION	1
2	LITERATURE REVIEW	4
3	AIM AND SCOPE	10
	3.1 Aim	10
	3.2 Scope	10
4	MATERIALS AND METHODS	11
	4.1 Block Diagram	11
	4.2 Python Packages	11
	4.2.1 Keras	11
	4.2.2 NumPy	12
	4.2.3 Pywt	13
	4.2.4 Cv2	14
	4.2.5 Pickle	15
	4.2.6 Sklearn	16
	4.2.7 SVC	17
	4.3 Methods	18
	4.3.1 Data Collection	18
	4.3.2 Data Augmentation	18
	4.3.3 Image Enhancement	18
	4.3.4 Image Segmentation	19
	4.3.5 Feature Extraction	20
	4.3.6 Classification	22

	4.4 Working	22
5	RESULTS AND DISCUSSION	29
6	CONCLUSION	31
	REFERENCES	32
	PROOF OF ACCEPTANCE	34
	APPENDIX	35

LIST OF FIGURES

FIGURE	DESCRIPTION	PAGE.NO
1.1	Conventional pap smear technique	2
4.1	Block diagram	10
4.2	Adaptive gamma corrected image for Presence data set	23
4.3	Adaptive gamma corrected image for Absence data set	24
4.4	Otsu segmented image from Presence data set	25
4.5	Otsu segmented image from Absence data set	25
4.6	Gabor Filtered image from Presence data set	26
s4.7	Gabor Filtered image from Absence data set	26
4.8	Contourlet Filtered image from Presence data set	27
4.9	Contourlet Filtered image from Absence data set	27
4.10	Obtained Parameters	28
5.1	Classification Report	29
5.2	The Confusion Matrix of the Prediction model	30

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	FULL FORM
1	ANN	Artificial Neural Network
2	CRT	Chemo Radio Therapy
3	CT	Computed Tomography
4	DTCWT	Dual-Tree Complex Wavelet Transform
5	DWT	Discrete Wavelet Transform
6	GLCM	Gray Level Occurrence Matrix
7	GLRLM	Gray Level Run Length Matrix
8	H2 MLP	Hierarchical Hybrid Multilayered Preceptron
9	HPV	Human Papiloma Virus
10	KNN	K-Nearest Neighbors
11	LACC	Locally Advanced Cervical Cancer
12	MRI	Magnetic Resonance Imaging
13	OLHT	Oriented Local Histogram Technique
14	SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

The cervix is the lower, narrow part of the uterus in the female reproductive system. The uterus holds a growing fetus during pregnancy. The cervix connects the lower part of the uterus to the vagina and, with the vagina, forms the birth canal. Cervical cancer begins when healthy cells on the surface of the cervix change or become infected with human papillomavirus (HPV) and grow out of control, forming a mass called a tumor. Long-term infection of HPV on the cervix can result in cancer, leading to a mass or tumor on the cervix. A tumor can be cancerous or benign. A cancerous tumor is malignant, meaning it can spread to other parts of the body. A benign tumor means the tumor will not spread. Cervical cancer can grow from the surface of the cervix seen in the vagina, called the ectocervix, or from the canal going from the vagina to the uterus, called the endocervix. HPV infection is spread through intercourse. Human papillomaviruses encompass more than 120 different types that may infect human skin and mucosa. Only 13–15 of these are found in cervical cancers and other malignancies and are called 'high risk' HPV (HPV-HR). HPV 16 is the most important HPV-HR-type; it is linked to approximately 50 % of cervical cancers worldwide. (Karl et al., 2014). It is usually comparatively easier to screen for and identify than other types of cancer, cervical cancer is crucial for achieving risk expectations because it can be greatly avoided. The cancerous cervical development is a cancerous tumor. Without a controlled mechanism for cell division or death, cervical tissue cells grow and replicate improperly. The cell spreads throughout the body if the tumor develops into a malignant form, causing some bodily parts to later get contaminated. Most of the deaths are occurred around the world based on cancer. Breast and Cervical cancers are the most live killing diseases of women patients in and around the world. Breast cancer is occurred in women patient externally and it can be detected by scanning the breast region. The women patients who are affected by breast cancer, can check themselves, hence it can be detected in women patients at an earlier manner as depicted in many works. The cervical cancer is also occurred in the women patient internally and it can

be detected by scanning the internal region of the vagina. The women patients who are affected by cervical cancer, unable to check themselves, hence it cannot be detected at an earlier stage. Hence, death ratio of the women patients is high due to cervical cancer. The cells inside the cervix are categorized into squamous cells and glandular cells as stated in (Mouelhi et al 2013). Squamous cell carcinoma is the type of cervical. The cervical cancer in women can be screened by either Pap smear test or Cervicogram test. This is painful procedure where the doctor will use a plastic or metal instrument, called a speculum, to look inside your vagina. This helps the doctor examine the vagina and the cervix, and collect a few cells and mucus from the cervix and the area around it. The cells are sent to a laboratory. This procedure may also result in false detection also, the procedure is shown in Fig.1.1.

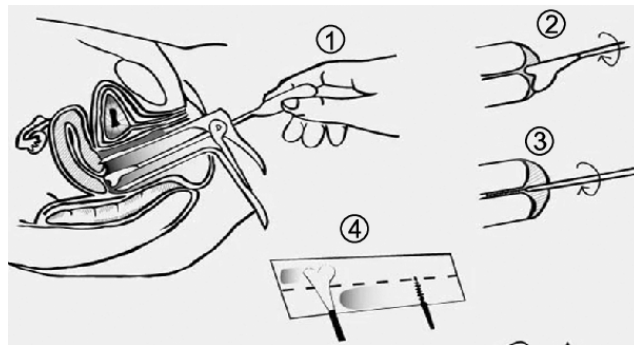


Fig1.1 Conventional pap smear technique

The main reason for the death due to cervical cancer in women patients is that it cannot be detected at an earlier stage and the patients can not receive any symptoms until they reach final stage of the cancer. The death ratio of the women patients can be reduced if it is detected at an earlier stage only. Hence, this paper proposes a methodology to detect the cervical cancer at an earlier stage to prevent death of women patients. Conventional methods are based on human experts and it is time consuming process. Further, it is not suitable for the developing countries having large population due to the lack of trained experts or radiologist in this field (Susan et al., 2018). Hence, there is a requirement of fully automated methodology for cervical cancer detection. The cervix region, which is the main region of interest within the cervigram, is located

in the central part of the image (Song et al. 2015, Sulaimana et al., 2011). Data mining is a process for locating relevant information from sources of information. Grimy information, like that is erroneous and incomplete, is part of real-world knowledge. In light of this, the outcome can be accurately reflected by purging and altering unreliable data to provide a reliable analytic delivery. It is fed into the dataset to discover the hidden information that would make a problem diagnosis and early identification simple. These methods are useful in the medical field for forecasting diseases as well as identifying similarities and connections between such symptoms. Numerous data mining tools can be used to rapidly recommend ongoing medical research and life-saving treatment, particularly for cervical cancer. Pre- processing 80% of the information is the initial phase, and it is crucial to all data mining processes. To mine, the data from the images collected from diagnostic machines like Magnetic Resonance Imaging(MRI) various machine learning and deep learning algorithms are to classify the images. Before that several features from the source, the image should be extracted and their texture should be analyzed to get a better classification. This paper suggest the early detection of cervical cancer using the MR-images collected from various sources for detection of presence of cancer. This also aims in overruling the conventional method of cancer detection

CHAPTER 2

LITERATURE SURVEY

NAZAH AH MUSTAFA et.al (2008) have proposed a technique where the characteristics of cervical cells are employed as neural network inputs and the classification of cervical cell types is used as the output target in order to use neural networks as a cervical cancer diagnosis method. In order to improve the cervical cell categorization system, this study suggests novel characteristics of cervical cells that can be employed as inputs for neural networks. The new cervical cell characteristics are taken from ThinPrep® pictures, and the hierarchical hybrid multilayered perceptron (H 2 MLP) is used to assess their appropriateness. The findings demonstrate that the suggested features—size, grey level, perimeter, red, green, blue, intensity1, intensity2, and saturation—are adequate for feeding as input to neural networks used in cervical cancer diagnosis systems to classify cervical cells.

SOUMYA. M. K, etal (2016) have proposed that Image processing techniques can be applied to some diagnostic equipment, such as X-ray, CT, MRI, etc., to determine the stage of a disease. For the output prediction, transform features with a strong energy component, like contourlet and gabor features, are used. Pre-treatment MR images of cervical cancer tumours are extensively used to predict outcome using second-order statistical characteristics based on contrast, correlation, energy, and homogeneity. In order to determine a patient's stage for cervical cancer, this research suggests a classification method using magnetic resonance images (MRI).

Alexandre G. de Brevern, et al (2015) have studied that how the NGS Big Data revolution has altered data management and analysis practices. They describe how there are numerous methodologies, tools, and data formats available to biologists. They focus on Big Data Information Technology advances from online and business intelligence to solve these issues. They stress how valuable NoSQL databases are

since they are so much more productive than relational databases. They describe business intelligence (BI) solutions that enable one to reclaim interactivity regardless of the volume of data since Big Data causes the loss of interactivity with data during analysis due to excessive processing time. They use the Amadea platform as their focal point to demonstrate this concept. Finally, they explore the difficulties that Big Data presents for visualization and outline the most recent advancements in JavaScript graphic libraries.

Noviyanti T M Sagala, et al (2017) have proposed several data mining techniques for identifying malignant cancer samples. Four separate medical tests—Biopsy, Cytology, Hinselmann, and Schiller—were used as four different target variables when SVM, Nave Bayes, and KNN data mining techniques were employed. Since the disease has no symptoms in the early stages, the characteristics that affect the disease the most are retrieved. Over 32 attributes and two separate algorithms, including Correlation-based Filter (CFS) and Random Forest, were used in the extraction process. The evaluation utilizing the 10-fold cross-validation approach in the R environment revealed that the Nave Bayes classifier surpasses other classifiers in terms of performance. Additionally, it has been demonstrated that the use of attribute selection can improve the performance of all classifiers on the dataset for cervical cancer in addition to allowing the selection of very important features. The results of this study show that classifiers can successfully obtain the best performance with the fewest possible highly significant attributes.

Alanna Vial, et al (2022) have made a review on radiomics—an objective tool for prognostic modelling of cancer tumors found within radiography images—is discussed. Radiomics is a cutting-edge feature transformation technique for identifying therapeutically important signals from radiological imaging data that are challenging to see with the naked eye. Machine learning and deep learning techniques are being studied more and more to aid in detection with the goal of improving patient diagnosis, treatment options, and results. A survey of the pertinent literature in the developing field of radiomics for cancer survival prediction is given. Research studies outside of

radiomics that outline methods that could be used in the future to enhance feature extraction and analysis are also examined. The subject of clinical image analysis known as radiomics is developing quickly and has a great deal of potential to help with cancer diagnosis and therapy decisions. In order for developments in computer vision and machine learning techniques to be utilized effectively, the realization of this aim of more effective decision making necessitates significant individual and integrated expertise from domain specialists in medical, biology, and computer science. If systems for data sharing or distributed learning are built to enhance the availability of data across all patient and tumor types, deep learning paired with machine learning has the potential to dramatically advance the science of radiomics in the years to come.

Mithlesh Arya, et al (2018) have proposed a technique in which the extraction of pertinent and instructive features from single-cell images is accomplished using a variety of texture-based features. GLCM, LBP, Laws, and DWT are employed to extract texture feature information. These techniques aid in identifying the cytoplasmic and nucleus contours. Based on the trained characteristics, ANN and SVM are utilized to categorize the single-cell pictures as normal or malignant. On each individual feature as well as the sum of all characteristics, ANN and SVM are employed. The best outcomes are attained when all qualities are combined. The created dataset MNITJ, which contains 330 single cervical cell pictures, as well as the publicly accessible benchmark Herlev data set are used to test the system. When compared to state-of-the-art shape-based features in terms of accuracy, experimental results demonstrate that the suggested texture-based features provide much better outcomes for cervical cancer detection.

Elayaraja P, et al (2019) have made a novel strategy for detecting cervical cancer using cervigram pictures is proposed. To improve the edges of the cervical picture, the Oriented Local Histogram Technique (OLHT) is used. Next, the Dual Tree Complex Wavelet Transform (DT-CWT) is used to produce a multi-resolution image. Then, from this altered multi-resolution cervical picture, features like wavelet, Grey Level Co-occurrence Matrix (GLCM), moment invariant, and Local Binary Pattern (LBP) features

are extracted. By using a feed forward back propagation neural network to train and evaluate these extracted properties, a cervical picture can be classified as normal or abnormal. To identify and segment the cancer location, morphological operations are applied to the aberrant cervical picture. Sensitivity, specificity, accuracy, positive predictive value, negative predictive value, likelihood ratio positive, likelihood ratio negative, precision, false positive rate, and false negative rate are all used to evaluate the performance of the suggested cervical cancer detection system. The cervical cancer detection system's performance metrics include 97.42% sensitivity, 99.36% specificity, 98.29% accuracy, 97.28% PPV, 92.17% NPV, 141.71 LRP, 0.0936 LRN, 97.38% precision, 96.72% FPR, and 91.36% NPR. According to the simulation results, the proposed approach performs better than the established methodologies for the segmentation and detection of cervical cancer.

François Lucia, et al (2018) have made a study to ascertain whether radiomics characteristics from 18fluorodeoxyglucose (FDG) positron emission tomography/computed tomography (PET/CT) and magnetic resonance imaging (MRI) pictures could affect cervical cancer prognosis. In their study, 102 patients with locally advanced cervical cancer (LACC) receiving chemoradiotherapy (CRT) from 2008 to 2016 were included (69 for training and 33 for testing). Each patient had an 18F-FDG PET/CT and MRI examination before to CRT, including T1, T2, and DWI (diffusion-weighted imaging). In the PET pictures and the MRI images, primary tumor volumes were identified using the fuzzy locally adaptive Bayesian algorithm and 3D SlicerTM, respectively. Radiomics characteristics (intensity, shape, and texture) were extracted, and their predictive usefulness for locoregional control and recurrence-free survival was compared with clinical parameters. They found that, in the training cohort, the median follow-up was 3.0 years (0.43-6.56 years); 36% of patients experienced relapse. Contrary to several radiomics parameters from both PET and MRI sequences, FIGO stage (I-II vs. III-IV) and metabolic response (complete vs. non-complete) were likely linked with outcome in univariate analysis but did not achieve statistical significance. Grey Level Non Uniformity GLRLM in PET and EntropyGLCM in ADC maps from DWI MRI were found to be independent predictive variables in training test multivariate

analysis. According to evaluations in the testing cohort, these showed much stronger predictive power than clinical parameters, with accuracy of 94% for predicting recurrence and 100% for indicating a lack of loco-regional control (compared to 50–60% for clinical markers). They concluded that, radiomics characteristics from functional imaging, such as Entropy GLCM and GLNUGLRLM from DWI-MRI and PET, respectively, are independent predictors of loco-regional control and recurrence in LACC treated with CRT, and they have a much higher predictive power than standard clinical indicators. It is need to conduct more study to confirm them, which could support more aggressive therapy for patients identified as having a high likelihood of recurrence.

Xin Zhao, et al (2022) have proposed a method in order to aid clinical diagnosis and treatment, their work intends to build an MRI-based classifier to discriminate early stages of cervical cancer with improved diagnostic performance. Before surgery, MRI exams were conducted. These included T1-weighted image (T1WI), T2-weighted image (T2W), diffusion weighted imaging (DWI), and dynamic contrast enhanced (DCE). As input, MR images from patients with segmented tumors and stage Ib or IIa cervical cancer were employed. First-order statistics, texturing, and filters were extracted as part of the feature extraction procedure. The least absolute shrinkage and selection operator was used to decrease the dimensionality of the radiomic characteristics (LASSO). Three machine learning techniques—k-nearest neighbor (KNN), support vector machine (SVM), and logistic regression (LR)—were used to train the models, and their diagnostic performance in identifying stage Ib and stage IIa cases was assessed. 27 features in all, comprising 2 features from T1WI, 5 features from T2WI, 5 features from DWI ($b = 50$), 4 features from DWI ($b = 800$), 5 features from DCE, and 6 features from ADC, were extracted in order to create models. Six training set and testing set sequences are modelled and assessed for each machine learning (ML) classifier. The training set and testing set of the T2WI model created by the SVM classifier were the best of all the models (Area under the curve (AUC) 0.915) / (AUC 0.907). To preoperatively stage early cervical cancer, radiomic analysis of ML-based texture characteristics and first-order statistics features can be applied.

XIN-XIANG LI, et al (2020) was to determine if whole-tumor dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI) and whole-lesion texture characteristics based on T2-weighted images. They concluded that a potential technique for detecting cervical cancer with parametrial invasion uses DCE-MRI together with texture analysis. Additionally, Ktrans, energy, and entropy work better together than they do separately, especially when it comes to increasing diagnostic sensitivity.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

Cervical cancer has become one of the major causes of cancer death among women worldwide. This can be cured in its earlier stage if found at early stages. The major technique used for the detection is pap smear technique which is found to cause greater pain in the patients and also the pathological tests are not always right and causes panic in various cases. Hence the pathological tests are correlated with the MRI and CT images. The main Aim of this work is to give a painless solution that can predict the cancer by processing the MRI images with various techniques and extract the features for classification of the Cervical cancer. The prediction of disease is done with the extracted feature parameters.

3.2 SCOPE

Magnetic Resonance Imaging (MRI) is a widely-used method of high quality medical imaging. The soft tissue contrast and non-invasiveness are the important advantages of MRIs. The radiologist examined the MRI to identify the presence of tumor abnormal tissues. The shortage of radiologist, the time conception and the misinterpretations caused by viewing the MRI through the naked eye called out for an automated system to analyze and classify medical images. This project proposes a classification model with MR Images, with the help of texture analysis techniques and support vector machine (SVM) classification. The model can predict the presence and absence of the by processing the MR-images.

CHAPTER 4

MATERIALS AND METHOD USED

4.1 BLOCK DIAGRAM

The input images are collected from various internet sources and it is enhanced with the help of various techniques such as adaptive gamma filtering and segmentation techniques such as Otsu segmentation for extracting certain features for the classification of cervical cancer. Then the extracted features are given to the SVM classifier for prediction of cervical cancer Fig.4.1 Shows the block diagram for the proposed method.

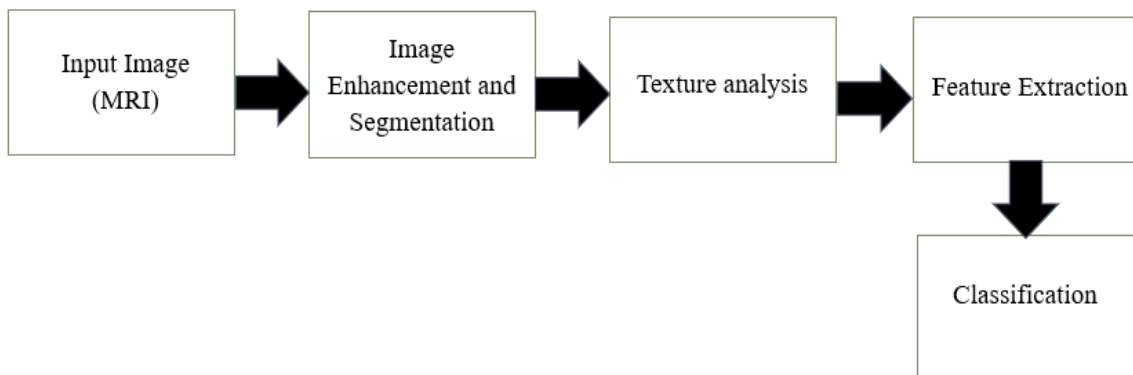


Fig. 4.1 Block diagram

4.2 PYTHON PACKAGES

4.2.1 Keras

Keras is an open-source Python library designed for building deep learning models. It provides a high-level interface for building neural networks that can run on top of different backends. Keras also supports a wide range of neural network architectures, including convolutional networks, recurrent networks, and combinations of both. Keras includes a set of pre-built layers, such as dense layers, convolutional layers, and

recurrent layers, that can be combined to create complex neural network architectures. It also includes a range of activation functions, loss functions, and optimization algorithms, making it easy to configure and train neural networks. One of the key features of Keras is its ability to support both CPU and GPU acceleration, which makes it well-suited for large-scale deep learning projects. Keras also supports distributed training across multiple GPUs and multiple machines, making it possible to scale up to even larger models and datasets. One of the benefits of using Keras for image processing is its support for data augmentation. Data augmentation is a technique used to artificially increase the size of a dataset by generating new samples through transformations of existing samples. Keras includes a range of pre-built image augmentation techniques, such as rotation, zooming, and flipping, which can be used to improve the robustness of a model to different types of input data. Keras also supports pre-trained models, such as VGG16, ResNet50, and InceptionV3, which have been trained on large datasets and can be used as a starting point for training new image classification models. These pre-trained models can be fine-tuned to specific image classification tasks by retraining the last few layers on a smaller dataset.

4.2.2 NumPy

NumPy is a popular Python package for scientific computing, particularly in the areas of numerical analysis and data science. It provides a powerful set of tools for working with arrays, matrices, and other multi-dimensional data structures. Contourlet transform is a multiscale and multidirectional image decomposition technique that is widely used in image processing and analysis. It is based on the wavelet transform, but unlike the wavelet transform, which decomposes images into a set of low-pass and high-pass subbands, the contourlet transform decomposes images into a set of subbands with both directional and frequency selectivity. NumPy is a useful Python package for implementing the contourlet transform because it provides efficient functions for working with multi-dimensional arrays, which are used to represent images in the transform. Here are some of the ways in which NumPy can be used in the contourlet transform:

Image representation: NumPy provides a simple and efficient way to represent images as multi-dimensional arrays. The contourlet transform operates on these arrays by decomposing them into subbands at different scales and orientations.

Wavelet transform: The contourlet transform is based on the wavelet transform, and NumPy provides efficient functions for computing the wavelet transform of an image. The wavelet transform is used to decompose the image into subbands at different scales and orientations, which are then further decomposed using the contourlet transform.

Subband filtering: The contourlet transform involves filtering the subbands of the image to extract features at different scales and orientations. NumPy provides functions for implementing various types of filters, including low-pass and high-pass filters.

Subband merging: After filtering, the subbands are merged to create a new image representation. NumPy provides functions for merging subbands, which can be used to reconstruct the image from the transformed representation.

Feature extraction: The contourlet transform is often used for feature extraction in image analysis. NumPy provides functions for computing various types of image features, such as texture, edge, and shape features, which can be extracted from the subbands of the contourlet transform.

4.2.3 Pywt

PyWavelets (pywt) is a Python package that provides tools for performing wavelet transforms and analyzing signals and images in both 1D and 2D domains. It is a popular package for wavelet-based image processing because it provides a variety of efficient algorithms and functions for performing wavelet transforms and other related tasks. Image compression: Wavelet transforms are often used for image compression because they can represent an image using fewer coefficients than the original pixel values. PyWavelets provides functions for performing discrete wavelet transforms (DWT) and inverse discrete wavelet transforms (IDWT), which can be used to compress and decompress images.

1. Image denoising: Wavelet transforms are also useful for removing noise from images. PyWavelets provides functions for performing wavelet denoising, which involves applying a wavelet transform to an image, thresholding the coefficients, and then applying an inverse wavelet transform to obtain a denoised image.
2. Image segmentation: Wavelet-based image segmentation involves dividing an image into regions based on differences in texture or intensity. PyWavelets provides functions for computing image gradients, which can be used to identify edges in an image, and for computing the multiscale local extrema, which can be used to segment an image.
3. Feature extraction: PyWavelets provides functions for computing various types of image features, such as texture, edge, and shape features. These features can be extracted from the wavelet coefficients obtained from a wavelet transform.
4. Image registration: Wavelet-based image registration involves aligning two or more images by minimizing the differences between their wavelet coefficients. PyWavelets provides functions for computing cross-correlation between wavelet coefficients, which can be used for image registration.

Overall, PyWavelets is a powerful tool for wavelet-based image processing and analysis. Its efficient algorithms and functions for performing wavelet transforms, denoising, segmentation, feature extraction, and registration make it a popular choice for researchers and practitioners in image processing and computer vision.

4.2.4 Cv2

cv2 is a Python package that is widely used for image and video processing. It is based on the OpenCV (Open Source Computer Vision) library and provides a set of efficient tools and functions for various tasks in image processing and computer vision. Here are some of the ways in which cv2 can be used in image processing:

1. Image reading and writing: cv2 provides functions for reading and writing images in various formats, such as JPEG, PNG, and BMP. These functions can be used to load images into Python programs for further processing or to save the processed images to disk.
2. Image filtering: cv2 provides a variety of functions for filtering images, such as blurring, sharpening, and edge detection. These functions can be used to preprocess images before further analysis or to enhance specific features of an image.
3. Image transformation: cv2 provides functions for transforming images, such as resizing, rotating, and cropping. These functions can be used to adjust the size and orientation of images to fit specific requirements.
4. Image segmentation: cv2 provides functions for segmenting images into regions based on color, texture, or other properties. These functions can be used for various applications, such as object detection and recognition.
5. Image feature extraction: cv2 provides functions for extracting various features from images, such as corners, edges, and histograms. These features can be used for various applications, such as image matching and object tracking.
6. Image analysis: cv2 provides functions for analyzing images, such as counting objects, measuring distances, and detecting patterns. These functions can be used for various applications, such as quality control and medical imaging.

4.2.5 Pickle

Pickle is a Python library that allows you to serialize and deserialize Python objects. It can be used to save the state of an object to a file, and then later load the object back into memory. While it is not directly related to image processing, it can be useful for saving and loading images, especially when used in combination with other Python libraries.

Here are some ways in which pickle can be used in image processing:

1. Saving and loading image data: You can use pickle to save and load the data associated with an image, such as the pixel values and metadata. This can be useful for storing large image datasets or for transferring image data between different programs.
2. Saving and loading trained models: You can use pickle to save and load trained models that have been used for image processing tasks, such as image classification or object detection. This can be useful for reusing models in future projects or for sharing them with other researchers.
3. Saving and loading intermediate results: You can use pickle to save and load intermediate results of image processing algorithms, such as feature vectors or image representations. This can be useful for speeding up the processing of large datasets or for reducing the memory requirements of your programs.

4.2.6 Sklearn

scikit-learn (sklearn) is a Python package that provides tools for data analysis, machine learning, and statistical modeling. While it is not specifically designed for image processing, it can be used for various tasks in this field. Here are some ways in which sklearn can be used in image processing:

1. Image classification: Sklearn provides a variety of classifiers that can be used for image classification tasks. These include logistic regression, support vector machines, decision trees, and random forests. These classifiers can be trained on labeled image datasets to predict the class of new images.
2. Dimensionality reduction: Sklearn provides tools for reducing the dimensionality of image data. These tools can be used to reduce the number of features in an image dataset while retaining the most important information. This can be useful for reducing the computational complexity of image processing algorithms.
3. Clustering: Sklearn provides tools for clustering image data based on similarities in their features. This can be useful for segmenting images into regions based on their properties.

4. Feature extraction: Sklearn provides tools for extracting various features from image data, such as color histograms, texture features, and edge features. These features can be used for various image processing tasks, such as object recognition and image retrieval.
5. Cross-validation: Sklearn provides tools for evaluating the performance of image processing algorithms using cross-validation techniques. This involves splitting the image dataset into training and testing sets and evaluating the performance of the algorithm on the testing set.

4.2.7 SVC

SVC (Support Vector Classification) is a classification algorithm implemented in scikit-learn (sklearn), a popular Python package for machine learning. It can be used in image processing for various tasks, such as image classification and object detection.

1. Image classification: SVC can be used to classify images into different categories based on their features. These features can be extracted from the images using various techniques, such as histogram of gradients (HOG) or deep learning-based feature extraction.
2. Object detection: SVC can be used in object detection tasks to classify regions of an image as containing a particular object or not. This can be useful for applications such as identifying specific objects in a complex scene or tracking objects over time.
3. Feature selection: SVC can be used in feature selection tasks to identify the most relevant features for a particular image processing task. This can be useful for reducing the dimensionality of the feature space and improving the efficiency of the classification algorithm.
4. Performance evaluation: SVC can be used to evaluate the performance of other image processing algorithms. For example, it can be used to evaluate the performance of different feature extraction methods or to compare the performance of different classifiers.

4.3 METHODS

4.3.1 Data Collection

This work uses Magnetic Resonance Imaging (MRI) as the diagnosing tool. The cervical cancer MRI in dicom (Digital Imaging and Communications in Medicine) format is taken as the input to this work. The image is converted into gray scale and the noise is removed to improve the image quality to get more surety and ease in detecting the tumor. The dataset consist of magnetic resonance images with locally advanced cervical cancer. The data set has been collected from a cancer archive website <https://www.cancerimagingarchive.net/nbia-search/?CollectionCriteria=TCGA-CESC>

4.3.2 Data Augmentation

Image is augmented for adding artificial changes to existing photos in order to increase the size of an image data set. By doing this, the existing image data set which represents a vast array of potential images is transformed into new and unique images. Here the images are augmented using reshaping and resizing methods.

4.3.3 Image Enhancement

The augmented images undergo various preprocessing techniques like enhancement and adaptive gamma correction.

Enhancement improves the brightness of dimmed images via the gamma correction and probability distribution of luminance pixels. The adaptive gamma correction (AGC) is formulated as per the following Equation(1),

$$T(l) = l_{max} \left(\frac{l}{l_{max}} \right)^{\gamma} = l_{max} \left(\frac{l}{l_{max}} \right)^{1-cdf(l)} \quad (1)$$

Gamma parameters are calculated as per Equation(2),

$$\gamma = 1 - cdf_{\omega}(l) \quad (2)$$

Gamma correction, a non-linear process, can be used to automatically increase the image contrast. Gamma correction is used to adjust an image's brightness or blackness. Gamma values range from 0.0 to 10.0. Gamma is only used to show images, not to alter the image data.

An adaptive gamma correction method is proposed to improve the contrast of the image where the appropriate gamma value is set automatically based on the statistics extracted from images.

4.3.4 Image Segmentation

Image segmentation is a crucial component of image processing. Image segmentation is the process of dividing a visual representation into useful structures. The goal of segmentation is to transform an image's representation into something more useful and understandable. Otsu is one the best image segmentation techniques for cancer images. It is a global thresholding method, it is simple and effective to use.

The method processes image histogram, segmenting the objects by minimization of the variance on each of the classes. Usually, this technique produces the appropriate results for bimodal images. The histogram of such an image contains two clearly expressed peaks, which represent different ranges of intensity values.

The core idea is separating the image histogram into two clusters with a threshold defined as a result of minimization the weighted variance of these classes

Otsu's equation is $\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$ (3)

The Equation(3) states the Otsu equation this algorithm can better cope with grayscale images contaminated by noise, and the segmented grayscale images are not only clear but also can better retain the detailed features of grayscale images.

4.3.5 Feature Extraction

Features represent the properties of a pixel pattern in a picture. To identify recurring patterns in the images, the feature extraction process makes use of a number of picture processing methods and mathematical analysis. It has been discovered that one of the most crucial imaging aspects of the field of the quantitative approach to medical imaging is texture features. In this project the features were extracted using Gabor feature, Contourlet feature and GLCM(Gray Level Co-occurrence Matrix).

Gabor Feature

Features from Gabor filters are frequently used to extract features from images. The Gabor filter wavelets take the shape of a sine wave that has had its Gaussian coefficient modified. The Gabor filters are employed in edge detection, image coding, texture segmentation, image representation, and document analysis, among other applications. In both the space and time domains, it offers the best resolution. Additionally, it gives texture images that make it a better visual representation. The final gabor 1-D filter function can be represented as the following Equation(4),

$$\psi(t) = \frac{|f|}{\gamma\sqrt{\pi}} e^{-\left(\frac{t}{\gamma}\right)^2} e^{j2\pi ft} \quad (4)$$

Features were extracted directly from gray-scale character images by Gabor filters which are designed from statistical information of character structures that are obtained from the Otsu segmented image.

Contourlet Feature

The Contourlet transform contains qualities like localization, multi-resolution, specimen, significant radial asymmetry, and property, which suggests that it might be a strategy for improving the picture nature and textures on distant surfaces, and sharply angular reading. Multi-resolutions and multi-dimensional operations are their basic capabilities. When compared to wavelets, contourlets will express a smooth contour with fewer coefficients. Using Contourlet to Detect Edges This method will more precisely find edges by using the contourlet transform of test gray scale images.

The Contourlet transform is done on the Gabor filtered images, it gives the properties of multi-resolution, localization, radial asymmetry important, specimen and property which could be a methodology of enhancing the image quality of textures.

GLCM - Gray Level Co-Occurrence Matrix

Using the feature extraction method known as GLCM, energy features of the cervical images are obtained. For any single channel image, GLCM can be constructed. The number of grey levels in the image which is taken as the source accords with the number of rows and columns in the GLCM square matrix.

GLCM Second-order statistical texture features characterize the texture of an image by calculating how often pairs of pixels with specific values and in a specified spatial relationship occur in an image.

The GLCM characteristics are used to distinguish between the cervical picture of cancer and the normal image of the cervix.

The characteristics of the picture of the cervix are extracted using the four features contrast, correlation, energy, and homogeneity.

- Contrast – The contrast measures the spatial frequency of an image and is a different moment of GLCM. It is the difference between the highest and the

lowest values of the adjacent set of pixels. The contrast texture measures the local variations present in the image.

- Correlation – Gives joint probability occurrence of specified pixel pairs.
- Energy – Provides the sum of squared elements in the GLCM. Also known as uniformity or the angular second moment.
- Homogeneity - Returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Range = [0 1] Homogeneity is 1 for a diagonal GLCM.

4.3.6 Classification

Classification is a task that involves assigning a class label to a given input based on a set of training data. The goal of classification is to build a model that can accurately predict the class label for new, unseen data. The classification of input is done by SVM (Support Vector Machine).

Support Vector Machine (SVM) is a regression prediction and classification tool that automatically prevents overfitting to the data while increasing predictive accuracy by using machine learning theory. Support vector machines (SVM) select the hyperplane with the largest margin in order to determine the best separating hyperplane between classes.

Supervised learning in SVM is demonstrated through classification. Using recognised labels, one can determine whether the system is operating correctly or not. This information can be utilized to either validate the system's accuracy or train it to respond in the appropriate manner.

Using SVM, we found 87% accuracy.

4.4 WORKING

Dataset from various Internet sources such as <https://www.cancerimagingarchive.net/nbia-search/?CollectionCriteria=TCGA-CESC> has been collected. The data is collected separately



Fig.4.3 Adaptive gamma corrected image for Absence data set

Then the OTSU segmentation has been done in both the data files for all the images in it. An important part in image processing is image segmentation. The divisions of an image into meaningful structures are called Image segmentation. The purpose of segmentation is to change the representation of an image into something that is more meaningful and easier to analyze. The methods are categorized on the basis of two properties discontinuity and similarity. Based on this property image segmentation is categorized as Edged based segmentation and region-based segmentation. The segmentation methods that are based on the discontinuity property of pixels are considered as boundary or edges based techniques. region-based segmentation is partitioning of an image into similar areas of connected pixels. There are different type of the Region based method like thresholding, region growing and region splitting and merging. Thresholding is a main technique in image segmentation applications. The basic idea is to select an optimal gray-level threshold value for dividing objects of interest in an image from the background based on their gray-level distribution. The minimum and maximum threshold for segmentation is kept at 120 and 255 respectively. The segmented images are saved as presence_otsu and absence_otsu for presence and absence of the cancer data set separately.



Fig.4.4 Otsu segmented image from Presence data set

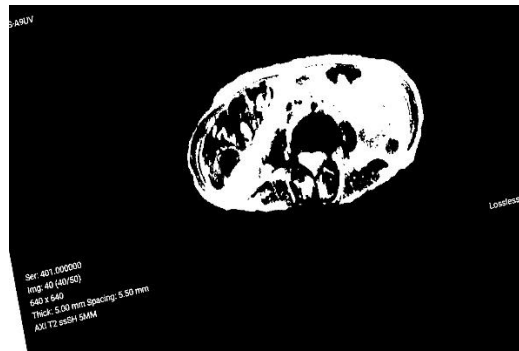


Fig.4.5 Otsu segmented image from Absence data set

Reducing the amount of resources needed to describe a huge set of data is feature extraction. The features used for the proposed system are statistical GLCM features and transform features such as Contourlet and Gabor. The obtained features are then taken as parameters for classification using SVM, and a set of test images has been taken for the prediction of cancer. Firstly the Gabor filtering is done on the Otsu segmented images. A function is designed to produce a set of Gabor Filters for which an even distribution of theta values equally distributed amongst π rad /180 degree. The local area to evaluate is set to 35 and larger values produce more edges. The difference between Offset value and lower generates gives cleaner results. Theta takes the orientation for edge detection. Brightness is also normalized using the function. This general function is designed to apply filters to our image. It first creates a numpy array for the saame size as our input image. Starting with a blank image, we loop through the images and apply our Gabor Filter. On each iteration, we take the highest value(super

impose), until we have the max value across all filters. It also loops through the kernels in our Gabor Filter and apply filter to image. The final image is returned. We use Numpy.maximum to compare our filter and cumulative image is obtained, taking the higher value(max). The presence of Gabor wavelets on the edges shows that the image is Gabor Filtered.



Fig.4.6 Gabor Filtered image from Presence data set



Fig.4.7 Gabor Filtered image from Absence data set

Next the Contourlet transform is done on the Gabor filtered images which is shown in Fig.4.8 and Fig.4.9, it gives the properties of multi-resolution, localization, radial asymmetry important, specimen and property which could be a methodology of enhancing the image quality of textures. The function is defined in such a way that it

performs Contourlet transform and then scale selection is performed, then invert transform, with the obtained mean pixels are then set to a threshold value of 0 and 255 minimum and maximum respectively. The python packages pywt,Cv2 is used in the above transforms.



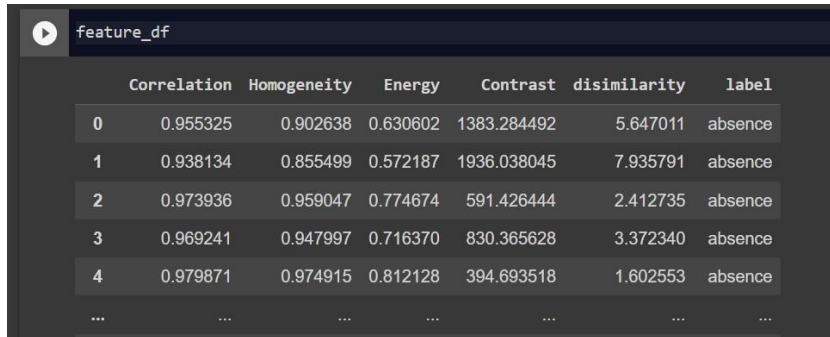
Fig.4.8 Contourlet Filtered image from Presence data set



Fig.4.9 Contourlet Filtered image from Absence data set

To extract Second order statistical features GLCM is is extraced from the filtered images the parameters such as Energy, Homogentiy, Correlation, Contrast, Dissimilarity has been obtained and fes into a Data frame, the datas are stored in a

.csv file, which is then trained for the classification of the cancer. The obtained data is also split into two for training and testing the model which is shown in Fig.4.10.



	Correlation	Homogeneity	Energy	Contrast	dissimilarity	label
0	0.955325	0.902638	0.630602	1383.284492	5.647011	absence
1	0.938134	0.855499	0.572187	1936.038045	7.935791	absence
2	0.973936	0.959047	0.774674	591.426444	2.412735	absence
3	0.969241	0.947997	0.716370	830.365628	3.372340	absence
4	0.979871	0.974915	0.812128	394.693518	1.602553	absence
...	...	...	...	...	...	...

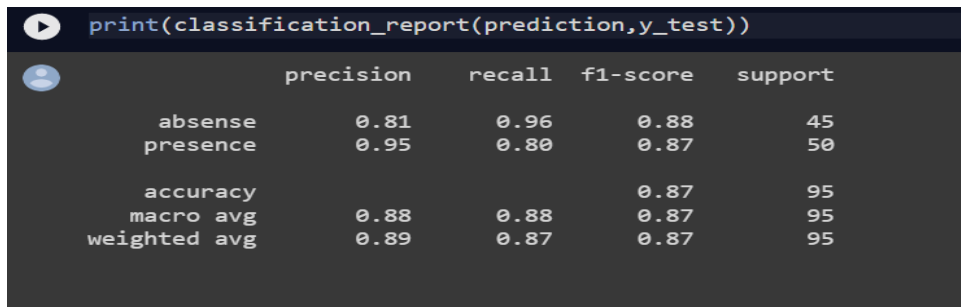
Fig.4.10 Obtained Parameters

Then the model is trained with the obtained numerical parameters to train and test the Data set for detection of presence or absence of the Cancer.

CHAPTER 5

RESULTS AND DISCUSSION

The performance analysis can be done with clinical measures. Analysis of the proposed system includes a comparison with the clinical results. The proposed system accuracy is compared with the old method accuracy value. The proposed system accuracy is calculated as 87% for the dataset1 that is T1-weighted axial images and T2-weighted axial images, accuracy is 82%. The performance analysis can be done with accuracy, sensitivity and specificity of the datasets. From the Confusion matrix, it is clear that the proposed method is having a better accuracy value. Calculating the TP, TN, FP, and FN values with the result of classification will give the accuracy, sensitivity and, specificity.



	precision	recall	f1-score	support
absense	0.81	0.96	0.88	45
presence	0.95	0.80	0.87	50
accuracy			0.87	95
macro avg	0.88	0.88	0.87	95
weighted avg	0.89	0.87	0.87	95

Fig.5.1 Classification Report

The Accuracy of SVM Classifier shown in Fig.5.1 is 87%. Precision - measures the accuracy of positive predictions. Recall – Measures the completeness of Positive prediction.

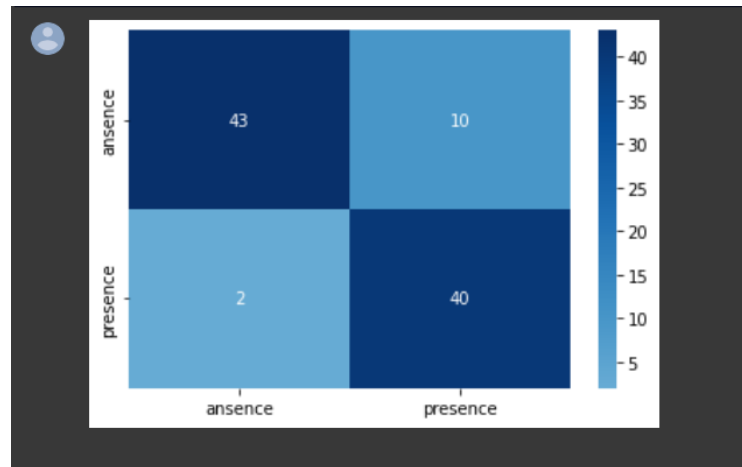


Fig.5.2 The Confusion Matrix of the Prediction model

The confusion matrix Fig.5.2 shows that out of 45 images taken from cervical cancer absent 43 images have been classified correctly by the prediction model. Similarly out of 50 images taken from cervical cancer present 40 images have been classified correctly by the prediction model. A set of 10 test images for both presence and absence of cancer has been taken and given to the model. It has successfully predicted the presence and absence of cancer in those images.

CHAPTER 6

CONCLUSION

The purpose of this study was to investigate whether features derived from MR images with locally advanced cervical cancer could be used to predict the presence of disease. Nonlinear SVM classification models were constructed based on both second-order texture features and transform features of the tumors. Transform features such as contourlet and Gabor features mainly based on energy are used for the prediction of output. However, second-order statistical features based on contrast, correlation, energy and homogeneity are significantly used to predict the outcome from MR images of cervical cancer. The best performing transform-based models had accuracies of around 87% for the axial T1-weighted and T2-weighted images. Thus, texture features outperformed transform features as well as statistical features for prediction.

REFERENCE

- [1] Arya, M., Mittal, N. and Singh, G. (2018), Texture-based feature extraction of smear images for the detection of cervical cancer. IET Comput. Vis. Vol. 12, No. 8. (2018) Pages 1049-1059.
- [2] de Brevern AG, Meyniel JP, Fairhead C, Neuvéglise C, Malpertuy A. Trends in IT innovation to build a next generation bioinformatics solution to manage and analyse biological big data produced by NGS technologies. BioMed Res Int. (2015)
- [3] Karl Ulrich Petry HPV and cervical cancer, Scandinavian Journal of Clinical and Laboratory Investigation, 74:sup244,(2014) 59-62,
- [4] Lucia, F., Visvikis, D., Desseroit, MC. et al. Prediction of outcome using pretreatment 18F-FDG PET/CT and MRI radiomics in locally advanced cervical cancer treated with chemoradiotherapy. Eur J Nucl Med Mol Imaging 45, (2018) Pages 768–786 .
- [5] Mouelhi A, Sayadi M, Fnaiech F, Mrad K, Romdhane KB. Automatic image segmentation of nuclear stained breast tissue sections using color active contour model and an improved watershed method. *Biomed Signal Process Control*. 2013;8:421–36.
- [6] Mustafa, N., N. A. Mat Isa, M. Y. Mashor, and N. H. Othman. "New Features of Cervical Cells for Cervical Cancer Diagnostic System Using Neural Network." IJSSST, Vol. 9, No. 2, | May 2008.
- [7] Punjani DN, Atkotiya KH. Cervical cancer prediction using data mining. Int J for Res Appl Sci Eng Tech. (2017) 5:1856.
- [8] P E, M S. Automatic Approach for Cervical Cancer Detection and Segmentation Using Neural Network Classifier. Asian Pac J Cancer Prev. 2018 Dec 25;19(12): Pages 3571-3580.
- [9] Song D, Edward K, Xiaolei H, et al. Multimodal entity coreference for cervical

dysplasia diagnosis. *IEEE Trans Med Imaging*. 2015;34:229–35. [[PubMed](#)] [[Google Scholar](#)]

[10] Sulaimana SN, Isab NAM, Othmanc NH. Semi-automated pseudo color features extraction technique for cervical cancers Pap smear images. *Int J Knowledge-based Intell Eng Syst*. 2011;15:131–43. [[Google Scholar](#)]

[11] Susan JC, Alex HK. Screening for cervical cancer US preventive services task force recommendation statement. *JAMA*. 2018;320: Pages 674–686.

[12] SoumyaM.K,SnehaK,ArunvinodhC.CervicalCancerDetectionand Classification Using Texture Analysis. *Biomed Pharmacol J* 2016;9(2).

[13] Vial, Alanna, David Stirling, Matthew Field, Montserrat Ros, Christian Ritz, Martin Carolan, Lois Holloway, & Alexis A. Miller. "The role of deep learning and radiomic feature extraction in cancer-specific predictive modelling: a review." *Translational Cancer Research* [Online], 7.3 (2018):Pages 803-816.

[14] Xin-xiang Li, Ting-ting Lin, Bin Liu, Wei. Diagnosis of Cervical Cancer With Parametrial Invasion on Whole-Tumor Dynamic Contrast-Enhanced Magnetic Resonance Imaging Combined With Whole-Lesion Texture Analysis Based on T2-Weighted Images. *Frontiers in Bioengineering and Biotechnology*, Vol. 8 , June 2020

[15] Xin Zhao, Xueyuan Wang, Bohao Zhang, Xianglong Liu, Desheng Xuan, Yuwei Xia, Xiaoan Zhang, Classifying early stages of cervicalcancer with MRI-based radiomics, *Magnetic Resonance Imaging*,Volume 89, 2022, Pages 70-76,

PROOF OF ACCEPTANCE



ACCEPTANCE LETTER

Manuscript ID : ICSSIT – 908

Manuscript Title: VARIOUS METHODS IN TEXTURE ANALYSIS AND CLASSIFICATION TECHNIQUES USED IN CERVICAL CANCER DETECTION

Author's: Poojahsri.S, Prerana.S, Grace kanmanni.P, Krishnakumar.S,

Sathyabama Institute of Science and Technology.

Dear Author's,

Greetings from Francis Xavier Engineering College!

5th International Conference on Smart Systems and Inventive Technology ICSSIT 2023 would like to congratulate you on the acceptance of your research manuscript to the International Conference ICSSIT 2023 which will be held on **23-25, January 2023** at Francis Xavier Engineering College, Tirunelveli, India. You have selected to deliver an oral presentation on your research work at ICSSIT 2023 conference.



APPENDIX

```
from google.colab import drive
drive.mount("/content/gdrive")
```

Importing necessary library

```
from keras.preprocessing.image import ImageDataGenerator
from imutils import paths
import keras
import numpy as np
from google.colab.patches import cv2_imshow
import cv2
from matplotlib import pyplot as plt
from scipy.linalg import fractional_matrix_power
import io
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from skimage import io, exposure, morphology, filters, color, \
    segmentation, feature, measure, img_as_float, img_as_ubyte
import pickle
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.preprocessing import StandardScaler
```

Data Augmentation

```

aug = ImageDataGenerator(rotation_range=30, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
    horizontal_flip=True, fill_mode="nearest")

# path of image folder
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/
dataset/presence"))

for img in imagePaths:
    img = cv2.imread(img)
    x = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    x = x.reshape((1,) + x.shape)

    i = 1
    for batch in aug.flow(x, batch_size=1, save_to_dir ='/content/gdrive/MyDrive/cervic
al_cancer_project/dataset/presence', save_prefix = '1', save_format='jpg'):
        i+=1
        if i==20:
            break

aug = ImageDataGenerator(rotation_range=30, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
    horizontal_flip=True, fill_mode="nearest")

# path of image folder
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/
dataset/absense"))

for img in imagePaths:
    img = cv2.imread(img)
    x = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    x = x.reshape((1,) + x.shape)

```

```

i = 1
for batch in aug.flow(x, batch_size=1, save_to_dir ='/content/gdrive/MyDrive/cervic
al_cancer_project/dataset/absense', save_prefix ='1', save_format='jpg'):
    i+=1
    if i==20:
        break

```

Adaptive Gamma Correction

```

def image_agcwd(img, a=0.25, truncated_cdf=False):
    h,w = img.shape[:2]
    hist,bins = np.histogram(img.flatten(),256,[0,256])
    cdf = hist.cumsum()
    cdf_normalized = cdf / cdf.max()
    prob_normalized = hist / hist.sum()

    unique_intensity = np.unique(img)
    intensity_max = unique_intensity.max()
    intensity_min = unique_intensity.min()
    prob_min = prob_normalized.min()
    prob_max = prob_normalized.max()

    pn_temp = (prob_normalized - prob_min) / (prob_max - prob_min)
    pn_temp[pn_temp>0] = prob_max * (pn_temp[pn_temp>0]**a)
    pn_temp[pn_temp<0] = prob_max * (-((-pn_temp[pn_temp<0])**a))
    prob_normalized_wd = pn_temp / pn_temp.sum() # normalize to [0,1]
    cdf_prob_normalized_wd = prob_normalized_wd.cumsum()

    if truncated_cdf:
        inverse_cdf = np.maximum(0.5,1 - cdf_prob_normalized_wd)
    else:

```

```

inverse_cdf = 1 - cdf_prob_normalized_wd

img_new = img.copy()
for i in unique_intensity:
    img_new[img==i] = np.round(255 * (i / 255)**inverse_cdf[i])

return img_new

def process_bright(img):
    img_negative = 255 - img
    agcwd = image_agcwd(img_negative, a=0.25, truncated_cdf=False)
    reversed = 255 - agcwd
    return reversed

imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/
dataset/presence"))
count = 1
for path in imagePaths:
    img = cv2.imread(path)
    # name = path.split('\')[ -1].split('.')[0]

    # Extract intensity component of the image
    YCrCb = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
    Y = YCrCb[:, :, 0]
    # Determine whether image is bright or dimmed
    threshold = 0.3
    exp_in = 112 # Expected global average intensity
    M,N = img.shape[:2]
    mean_in = np.sum(Y/(M*N))
    t = (mean_in - exp_in)/ exp_in

    # Process image for gamma correction

```

```

img_output = None
if t > threshold:
    result = process_bright(Y)
    YCrCb[:, :, 0] = result
    img_output = cv2.cvtColor(YCrCb, cv2.COLOR_YCrCb2BGR)
else:
    img_output = img
cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/presence_ga
mma/img_{}.jpg".format(count), img_output)
count = count + 1
def image_agc_wd(img, a=0.25, truncated_cdf=False):
    h, w = img.shape[:2]
    hist, bins = np.histogram(img.flatten(), 256, [0, 256])
    cdf = hist.cumsum()
    cdf_normalized = cdf / cdf.max()
    prob_normalized = hist / hist.sum()

    unique_intensity = np.unique(img)
    intensity_max = unique_intensity.max()
    intensity_min = unique_intensity.min()
    prob_min = prob_normalized.min()
    prob_max = prob_normalized.max()

    pn_temp = (prob_normalized - prob_min) / (prob_max - prob_min)
    pn_temp[pn_temp > 0] = prob_max * (pn_temp[pn_temp > 0]**a)
    pn_temp[pn_temp < 0] = prob_max * (-((-pn_temp[pn_temp < 0])**a))
    prob_normalized_wd = pn_temp / pn_temp.sum() # normalize to [0,1]
    cdf_prob_normalized_wd = prob_normalized_wd.cumsum()

    if truncated_cdf:
        inverse_cdf = np.maximum(0.5, 1 - cdf_prob_normalized_wd)

```

```

else:
    inverse_cdf = 1 - cdf_prob_normalized_wd

img_new = img.copy()
for i in unique_intensity:
    img_new[img==i] = np.round(255 * (i / 255)**inverse_cdf[i])

return img_new

def process_bright(img):
    img_negative = 255 - img
    agcwd = image_agcwd(img_negative, a=0.25, truncated_cdf=False)
    reversed = 255 - agcwd
    return reversed

imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/
dataset/absense"))
count = 1
for path in imagePaths:
    img = cv2.imread(path)
    # name = path.split('\\')[-1].split('.')[0]

    # Extract intensity component of the image
    YCrCb = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
    Y = YCrCb[:, :, 0]
    # Determine whether image is bright or dimmed
    threshold = 0.3
    exp_in = 112 # Expected global average intensity
    M,N = img.shape[:2]
    mean_in = np.sum(Y/(M*N))
    t = (mean_in - exp_in)/ exp_in

```

```

# Process image for gamma correction
img_output = None
if t > threshold:
    result = process_bright(Y)
    YCrCb[:, :, 0] = result
    img_output = cv2.cvtColor(YCrCb, cv2.COLOR_YCrCb2BGR)
else:
    img_output = img
cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/absense_gamma/img_{}.jpg".format(count), img_output)
count = count + 1

```

Segmentation - Otsu's technique

```

imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/dataset/absense_gamma"))
count = 1
for img_path in imagePaths:
    img_read = cv2.imread(img_path)
    img = cv2.cvtColor(img_read, cv2.COLOR_BGR2GRAY)
    ret, thresh1 = cv2.threshold(img, 120, 255, cv2.THRESH_BINARY +
                                cv2.THRESH_OTSU)
    cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/absense_ostu/img_{}.jpg".format(count), thresh1)
    count = count + 1
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/dataset/presence_gamma"))
count = 1
for img_path in imagePaths:
    img_read = cv2.imread(img_path)
    img = cv2.cvtColor(img_read, cv2.COLOR_BGR2GRAY)
    ret, thresh1 = cv2.threshold(img, 120, 255, cv2.THRESH_BINARY +

```

```

cv2.THRESH_OTSU)
cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/presence_ost
u/img_{}.jpg".format(count),thresh1)
count = count+1
Gabor Filter
def create_gaborfilter():
    # This function is designed to produce a set of GaborFilters
    # an even distribution of theta values equally distributed amongst pi rad / 180 degree
    e

    filters = []
    num_filters = 16
    ksize = 35 # The local area to evaluate
    sigma = 3.0 # Larger Values produce more edges
    lambd = 10.0
    gamma = 0.5
    psi = 0 # Offset value - lower generates cleaner results
    for theta in np.arange(0, np.pi, np.pi / num_filters): # Theta is the orientation for ed
ge detection
        kern = cv2.getGaborKernel((ksize, ksize), sigma, theta, lambd, gamma, psi, ktyp
e=cv2.CV_64F)
        kern /= 1.0 * kern.sum() # Brightness normalization
        filters.append(kern)
    return filters
def apply_filter(img, filters):
    # This general function is designed to apply filters to our image

    # First create a numpy array the same size as our input image
    newimage = np.zeros_like(img)

```

```

    # Starting with a blank image, we loop through the images and apply our Gabor Filter
    # On each iteration, we take the highest value (super impose), until we have the max value across all filters
    # The final image is returned
    depth = -1 # remain depth same as original image

    for kern in filters: # Loop through the kernels in our GaborFilter
        image_filter = cv2.filter2D(img, depth, kern) #Apply filter to image

        # Using Numpy.maximum to compare our filter and cumulative image, taking the higher value (max)
        np.maximum(newimage, image_filter, newimage)
    return newimage

# We create our gabor filters, and then apply them to our image
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/dataset/absense_ostu"))
count = 1
for img_path in imagePaths:
    gfilters = create_gaborfilter()
    img_read = cv2.imread(img_path)
    image_g = apply_filter(img_read, gfilters)
    cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/gabor_absense/img_{}.jpg".format(count), image_g)
    count = count+1

# We create our gabor filters, and then apply them to our image
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/dataset/presence_ostu"))
count = 1
for img_path in imagePaths:
    gfilters = create_gaborfilter()

```

```

img_read = cv2.imread(img_path)
image_g = apply_filter(img_path, gfilters)
cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/gabor_prese
nce/img_{}.jpg".format(count),image_g)
count = count+1

```

Contoulet Transform Feature

```

import numpy as np
import pywt
import cv2

def contourlet_edge_detection(image, scale_factor):
    # 1. Contourlet transform
    coeffs = pywt.dwt2(image, 'db1', mode='symmetric')
    LL, (LH, HL, HH) = coeffs

    # 2. Scale selection
    LH[abs(LH) < scale_factor*np.mean(abs(LH))] = 0
    HL[abs(HL) < scale_factor*np.mean(abs(HL))] = 0
    HH[abs(HH) < scale_factor*np.mean(abs(HH))] = 0

    # 3. Invert the transform
    coeffs = (LL, (LH, HL, HH))
    output = pywt.idwt2(coeffs, 'db1', mode='symmetric')

    # 4. Threshold using mean of pixel values
    threshold = np.mean(output)
    output[output < threshold] = 0
    output[output >= threshold] = 255

    return output.astype(np.uint8)

```

```

# contourlet transform for absense
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/
dataset/gabor_absense"))
count = 1
for image in imagePaths:
    img_read = cv2.imread(image)
    gray_img = cv2.cvtColor(img_read,cv2.COLOR_BGR2GRAY)
    contour_img = contourlet_edge_detection(gray_img,5)
    cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/contour_abse
nse/ct_{}.jpg".format(count),contour_img)
    count = count+1
# contourlet transform for presence
imagePaths = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/
dataset/gabor_presence"))
count = 1
for image in imagePaths:
    img_read = cv2.imread(image)
    gray_img = cv2.cvtColor(img_read,cv2.COLOR_BGR2GRAY)
    contour_img = contourlet_edge_detection(gray_img,5)
    cv2.imwrite("/content/gdrive/MyDrive/cervical_cancer_project/dataset/contour_pres
ence/ct_{}.jpg".format(count),contour_img)
    count = count+1
def glcmFeature(gray_img):
    glcm = feature.greycomatrix(image=img_as_ubyte(gray_img), distances=[1],
                                angles=[0, np.pi/4, np.pi/2, np.pi * 3/2],
                                symmetric=True)

    Correlation=(np.mean(feature.greycoprops(glcm, prop='correlation'))))
    Homogeneity=(np.mean(feature.greycoprops(glcm, prop='homogeneity'))))
    Energy=(np.mean(feature.greycoprops(glcm, prop='energy'))))

```

```

Contrast=(np.mean(feature.greycoprops(glcmm, prop='contrast'))))
disimilarity=(np.mean(feature.greycoprops(glcmm, prop ="disimilarity"))))
asm=(np.mean(feature.greycoprops(glcmm, prop='ASM'))))
return Correlation,Homogeneity,Energy,Contrast,disimilarity,asm
glcmm_feature=[]
image_absense = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/dataset/contour_absense"))
image_presence = list(paths.list_images("/content/gdrive/MyDrive/cervical_cancer_project/dataset/contour_presence"))
image = image_absense+image_presence
for img_path in image:
    label = img_path.split("/")[-2].split("_")[1]
    image = cv2.imread(img_path)
    gray_img = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    Correlation,Homogeneity,Energy,Contrast,disimilarity,asm = glcmmFeature(gray_img
)
    glcmm_feature.append([Correlation,Homogeneity,Energy,Contrast,disimilarity,asm,label])
# extracted feature convert into dataframe
feature_df = pd.DataFrame(glcmm_feature)
feature_df.columns = ["Correlation","Homogeneity","Energy","Contrast","disimilarity","asm","label"]
# feature_df.to_csv("/content/gdrive/MyDrive/cervical_cancer_project/features.csv",index = None)s
# Read extracted feature using pd library
features_df = pd.read_csv("/content/gdrive/MyDrive/cervical_cancer_project/features.csv")
X = features_df.iloc[:, :-1]
y = features_df.iloc[:, -1]
Scaler = StandardScaler()
Scaler.fit(X)

```

```
# pickle.dump(Scaler,open("/content/gdrive/MyDrive/cervical_cancer_project/scaled_
weight.p","wb"))
X = Scaler.transform(X)
# Dataset split into training and testing part
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.25, random_state =
42)
```

Training using support vector machine

```
svm_model = SVC(C = 1, kernel = "rbf")
svm_model.fit(X_train, y_train)
prediction = svm_model.predict(X_test)
# pickle.dump(svm_model,open("/content/gdrive/MyDrive/cervical_cancer_project/sv
m_weight.p","wb"))
print('Accuracy of Svm = {} %'.format(round(accuracy_score(y_test,prediction)*100,2))
)
```

classification report

```
print(classification_report(prediction,y_test))
```

confusion matrix

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
labels = ['ansence','presence']
CM = confusion_matrix(y_test,prediction)
# drawing confusion matrix
sns.heatmap(CM, center = True , annot=True, fmt="d" ,cmap="Blues", xticklabels=lab
els, yticklabels=labels)
plt.show()
```

Prediction

```
!unzip /content/test.zip -d .
```

```
def image_agcwd(img, a=0.25, truncated_cdf=False):
    h,w = img.shape[:2]
    hist,bins = np.histogram(img.flatten(),256,[0,256])
    cdf = hist.cumsum()
    cdf_normalized = cdf / cdf.max()
    probb_normalized = hist / hist.sum()

    unique_intensity = np.unique(img)
    intensity_max = unique_intensity.max()
    intensity_min = unique_intensity.min()
    probb_min = probb_normalized.min()
    probb_max = probb_normalized.max()

    pn_temp = (probb_normalized - probb_min) / (probb_max - probb_min)
    pn_temp[pn_temp>0] = probb_max * (pn_temp[pn_temp>0]**a)
    pn_temp[pn_temp<0] = probb_max * (-((-pn_temp[pn_temp<0])**a))
    probb_normalized_wd = pn_temp / pn_temp.sum() # normalize to [0,1]
    cdf_probb_normalized_wd = probb_normalized_wd.cumsum()

    if truncated_cdf:
        inverse_cdf = np.maximum(0.5,1 - cdf_probb_normalized_wd)
    else:
        inverse_cdf = 1 - cdf_probb_normalized_wd

    img_new = img.copy()
    for i in unique_intensity:
        img_new[img==i] = np.round(255 * (i / 255)**inverse_cdf[i])
```

```

    return img_new

def process_bright(img):
    img_negative = 255 - img
    agcwd = image_agcwd(img_negative, a=0.25, truncated_cdf=False)
    reversed = 255 - agcwd
    return reversed

import warnings
warnings.filterwarnings("ignore")

input_image = "/content/test/presence/1.jpg"
img = cv2.imread(input_image)
# name = path.split('\\')[-1].split('.')[0]
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Extract intensity component of the image
YCrCb = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
Y = YCrCb[:, :, 0]
# Determine whether image is bright or dimmed
threshold = 0.3
exp_in = 112 # Expected global average intensity
M, N = img.shape[:2]
mean_in = np.sum(Y / (M * N))
t = (mean_in - exp_in) / exp_in

# Process image for gamma correction
img_output = None
if t > threshold:
    result = process_bright(Y)
    YCrCb[:, :, 0] = result

```

```

img_output = cv2.cvtColor(YCrCb,cv2.COLOR_YCrCb2BGR)
else:
    img_output = img
gamma_img = cv2.cvtColor(img_output, cv2.COLOR_BGR2GRAY)
ret, thresh1 = cv2.threshold(gamma_img, 120, 255, cv2.THRESH_BINARY +
                             cv2.THRESH_OTSU)
gfilters = create_gaborfilter()
image_g = apply_filter(thresh1, gfilters)
contourlet_img = contourlet_edge_detection(image_g,5)
Correlation,Homogeneity,Energy,Contrast,disimilarity,asm = glcmFeature(gray_img)
feature_data = [Correlation,Homogeneity,Energy,Contrast,disimilarity,asm]

feature_data = np.array(feature_data).reshape(1, -1)
scaled_weight = pickle.load(open("/content/gdrive/MyDrive/cervical_cancer_project/s
caled_weight.p","rb"))
Scaled_data = scaled_weight.transform(feature_data)
svm_model = pickle.load(open("/content/gdrive/MyDrive/cervical_cancer_project/svm
_weight.p","rb"))
prediction = svm_model.predict(Scaled_data)
print("predicted class is:",prediction[0])

```