THYROID ULTRASOUND IMAGE CLASSIFICATION USING DEEP LEARNING APPROACH

K. SAAI DEEPTHI REDDY

(19951A04B8)

THYROID ULTRASOUND IMAGE CLASSIFICATION USING DEEP LEARNING APPROACH

A project Report
Submitted in partial fulfillment of the
requirements for the award of the degree of

Bachelor of Technology
in

Electronics & Communication Engineering

by

K. SAAI DEEPTHI REDDY (19951A04B8)

Under the Guidance of

Mr. J. Siva Rama Krishna

Assistant Professor

Department of Electronics & Communication Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad – 500043, Telangana

January 2023

DECLARATION

I certify that

a. The work contained in this report is original and has been done by me under the guidance of my supervisor(s)
b. The work has not been submitted to any other Institute for any degree or diploma
c. We have followed the guidelines provided by the Institute in preparing the report
d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute
e. Whenever We have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Place: Hyderabad **Signature of the Student**

Date: **Roll No.:19951A04B8**

CERTIFICATE

This is to certify that the project report entitled **Thyroid Ultrasound Image Classification Using Deep Learning Approach** submitted by **Ms. K. Saai Deepthi Reddy** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology **Electronics & CommunicationEngineering** is bonafide in record of work carried out by him/her under my/our guidance and supervision. The content of this report, in full or parts, have not been submitted to any other Institutefor the award of any Degree.

Supervisor	Head of the Department
Mr. J. Siva Rama Krishna	**Dr. P Ashok Babu**
Assistant Professor, ECE	Professor, ECE

Date:

APPROVAL SHEET

This project report entitled **Thyroid Ultrasound Image Classification Using Deep Learning Approach** done by **K. Saai Deepthi Reddy(19951A04B8)** is approved for the award of the Degree Bachelor of Technology in Department of **Electronics & Communication Engineering.**

Examiners

Supervisor(s)
Mr. J. Siva Rama Krishna
Assistant Professor, ECE

Principal

Dr. L.V. Narasimha Prasad

Date:

Place:

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and engagement crown all the efforts with success. We thank our college management and respected Sri **M. Rajashekar Reddy**, Chairman, I.A.R.E, Dundigal for providing us the necessary infrastructure to carry out the project work. We express our sincere thanks to **Dr. L V NarasimhaPrasad**, who has been a great source of information for our work and to our beloved Principal, **Dr. P. Ashok Babu** Professor and Head, Department of ECE for extending his support to carry onthis project work **Dr. J. Siva Rama Krishna** , **Associate Professor**, Department of ECE and special thanks to our Project Coordinator **Ms. S. Sushma z,** Associate Professor, Department of ECE for their internal support and professionalism who helped us in shaping the project into successful one.

We take this opportunity to express our thanks to one and all who directly or indirectly helped usin bringing this effort to present form.

.

ABSTRACT

Nowadays, diagnosis of thyroid nodules is mainly based on clinical methods, which requires a lot of manpower and medical resources. Therefore, this work proposes an automated thyroid ultrasound nodule diagnosis method that combines convolutional neural networks and image texture features. The main steps include: Firstly, ultrasound thyroid nodule dataset is established by collecting positive and negative samples, standardizing of images and segmentation of nodule area. Secondly, through texture features extraction, feature selection and data dimensionality reduction, texture features model is obtained; Thirdly, by transfer learning, deep neural network is used to obtain feature model of the nodule in images; Then, texture features model and convolutional neural network feature model are combined to form a new nodule feature model called Feature Fusion Network; Finally, Feature Fusion Network is applied to train and improve performance than single network, and a deep neural network diagnosis model that can adapt to the characteristics of thyroid nodules is built. In order to test this method, 1874 groups of clinical ultrasound thyroid nodules are collected. Harmonic average F-score based on Precision and Recall is used as an evaluation indicator. Experimental results show that Feature Fusion Network can distinguish between benign and malignant thyroid nodules with an F-score of 92.52%. Compared with traditional machine learning methods and convolutional neural networks, performance of this work is better.

Keywords- Ultrasound image, diagnosis of thyroid nodules, texture features, convolutional neural network, feature fusion.

LIST OF CONTENTS

LIST OF FIGURES

Chapter 1
INTRODUCTION

With the increase of people's life pressure, prevalence of thyroid nodules has increased year by year in the world. It has become one of the most important diseases and is threatening human health. Therefore, early diagnosis of thyroid nodules is very important. The diagnostic methods of thyroid nodules mainly include ultrasound examination, CT examination, aspiration biopsy and pathological examination. CT examination requires nuclear scanning, which is harmful to patients and is expensive. Needle biopsy and pathological examination are more commonly used and reliable methods, but these two methods are very traumatic to thyroid tissue. Also, their diagnosis process is cumbersome, which will occupy more medical resources. Ultrasonography is currently the common imaging method for diagnosing thyroid diseases. It has the advantages of simplicity, good reproducibility, non-invasive, fast and low price. Usually, doctors can only judge benign and malignant based on clinical experience, which are highly subjective and easily affected. Therefore, the ability to accurately and quickly identify and diagnose the pathology of ultrasound thyroid nodules has become an increasingly urgent need. In recent years, application of artificial intelligence technology in the medicine has gradually increased, especially in imaging and signal.

How to use information of ultrasound images to establish a computer-assisted automated thyroid diagnosis system is an important direction of current research. The commonly applied method of assisting medical diagnosis is to use features extraction engineering and classifiers for classification. For example, Zheng et al. used LR (Logistic Regression) to screen out indicators that had a greater impact on judging benign and malignant thyroid.

This regression models can achieve pathological classification of images. Liu et al. extracted local texture features of thyroid nodules from region of interest, and applied KNN (K-NearestNeighbor) algorithm to obtain diagnosis results. Choi and Choi took thresholds and 3D connected region labeling methods to assist doctors in detection by classifiers based on genetic planning. These technologies are based on computer theoretical systems and establish accurate computer diagnosis methods. However, it depends on the completeness of feature textures information and selection of a suitable classifier.

1.1 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Platform – In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries.

Operating system is one of the first requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

APIs and drivers – Software making extensive use of special hardware devices, like high-end display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms.

Web browser – Most web applications and software depending heavily on Internet technologies make use of the default browser installed on system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, which makes use of ActiveX controls, despite their vulnerabilities.

1) Visual Studio Community Version

2)Nodejs (Version 12.3.1)

3)Python IDEL (Python 3.7)

1.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Architecture – All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.

Processing power – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity,and are often mentioned in this category.

Memory – All software, when run, resides in the random-access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement.

Secondary storage – Hard-disk requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

Display adapter – Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements.

Peripherals – Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc.

1)Operating System : Windows Only
2)Processor : i5 and above
3)Ram: 4gb and above
4)Hard Disk: 50 GB

1.3 Existing Method

With the development of deep learning, some researchers are studying convolutional neural networks to diagnose thyroid ultrasound nodules. For example, Moran et al. established S-Detect technology based on the Google Net. They cooperated with clinical sonographers for joint diagnosis to improve diagnostic performance. Xie et al. decomposed nodules into 9 views to learn

3D features. They built a multi-view knowledge-based collaborative model for each view and input three images into ResNet-50 network for training to represent appearance, voxel, and shape specificity. In summary, convolutional neural network usually does not require too much pre-processing operations, and has advantages of convenience and simplicity. However, it is very dependent on feature completeness of training data due to lack of sufficient prior theoretical support. In this case, direction and details of feature training are usually unknown. How to further improve diagnosis accuracy is still urgently needed.

1.4 Proposed Method

This research suggests an automated thyroid ultrasonography nodule diagnosis technique that incorporates picture texture data and convolutional neural networks. The key actions consist of: By gathering both positive and negative samples, normalizing the pictures, and segmenting the nodule area, the ultrasonography thyroid nodule dataset is first created. Second, a texture features model is created through the extraction of texture features, feature selection, and data dimensionality reduction; Thirdly, a deep neural network is employed to obtain a feature representation of the nodule in photos by transfer learning; Then, a new nodule feature model called Feature Fusion Network is created by fusing the texture and convolutional neural network feature models; A deep neural network diagnosis model that can adapt and feature fusion network are used to train and increase performance over single networks.

Chapter 2
LITERATURE SURVEY

Although early diagnosis of thyroid nodule type is very important, the diagnostic accuracy of standard tests is a challenging issue. We here aimed to find an optimal combination of factors to improve diagnostic accuracy for distinguishing malignant from benign thyroid nodules before surgery. In a prospective study from 2008 to 2012, 345 patients referred for thyroidectomy were enrolled. The sample size was split into a training set and testing set as a ratio of 7:3. The former was used for estimation and variable selection and obtaining a linear combination of factors. We utilized smoothly clipped absolute deviation (SCAD) logistic regression to achieve the sparse optimal combination of factors

Although early diagnosis of thyroid nodule type is very important, the diagnostic accuracy of standard tests is a challenging issue. We here aimed to find an optimal combination of factors to improve diagnostic accuracy for distinguishing malignant from benign thyroid nodules before surgery. In a prospective study from 2008 to 2012, 345 patients referred for thyroidectomy were enrolled. The sample size was split into a training set and testing set as a ratio of 7:3. The former was used for estimation and variable selection and obtaining a linear combination of factors. We utilized smoothly clipped absolute deviation (SCAD) logistic regression to achieve the sparse optimal combination of factors. To evaluate the performance of the estimated model in the testing set, a receiver operating characteristic (ROC) curve was utilized.

At present, cancer imaging examination relies mainly on manual reading of doctors, which requests a high standard of doctors' professional skills, clinical experience, and concentration. However, the increasing amount of medical imaging data has brought more and more challenges to radiologists. The detection of digestive system cancer (DSC) based on artificial intelligence (AI) can providea solution for automatic analysis of medical images and assist doctors to achieve high-precision intelligent diagnosis of cancers

This paper proposed a non-segmentation radiological method for classification of benign and malignant thyroid tumors using B mode ultrasound data. This method aimed to combine the advantages of morphological information provided by ultrasound and convolutional neural networks in automatic feature extraction and accurate classification. Compared with the traditional feature extraction method, this method directly extracted features from the data set without the need for segmentation and manual operations. 861 benign nodule images and 740 malignant nodule images were collected for training data.

Background In this study, images of 2450 benign thyroid nodules and 2557 malignant thyroid nodules were collected and labeled, and an automatic image recognition and diagnosis system was established by deep learning using the YOLOv2 neural network. The performance of the system in the diagnosis of thyroid nodules was evaluated, and the application value of artificial intelligence in clinical practice was investigated. Methods The ultrasound images of 276 patients were retrospectively selected. The diagnoses of the radiologists were determined according to the Thyroid Imaging Reporting and Data System; the images were automatically recognized and diagnosed by the established artificial intelligence system. Pathological diagnosis was the gold standard for the final diagnosis.

The sensitivity, positive predictive value, negative predictive value, and accuracy of the artificial intelligence diagnosis system for the diagnosis of malignant thyroid nodules were 90.5%, 95.22%, 80.99%, and 90.31%, respectively, and the performance did not significantly differ from that of the radiologists ($p > 0.05$). The artificial intelligence diagnosis system had a higher specificity (89.91% vs 77.98%, $p = 0.026$). Conclusions Compared with the performance of experienced radiologists, the artificial intelligence system has comparable sensitivity and accuracy for the diagnosis of malignant thyroid nodules and better diagnostic ability for benign thyroid nodules. As an auxiliary tool, this artificial intelligence diagnosis system can provide radiologists with sufficient assistance in the diagnosis of benign and malignant thyroid nodules.

Chapter-3

METHODOLOGY

3.1 Functional Requirements

1. Data Collection
2. Data Pre processing
3. Training and Testing
4. Modiling
5. Predicting

3.2 Non-Functional Requirements

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement

- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

3.3 System Architecture

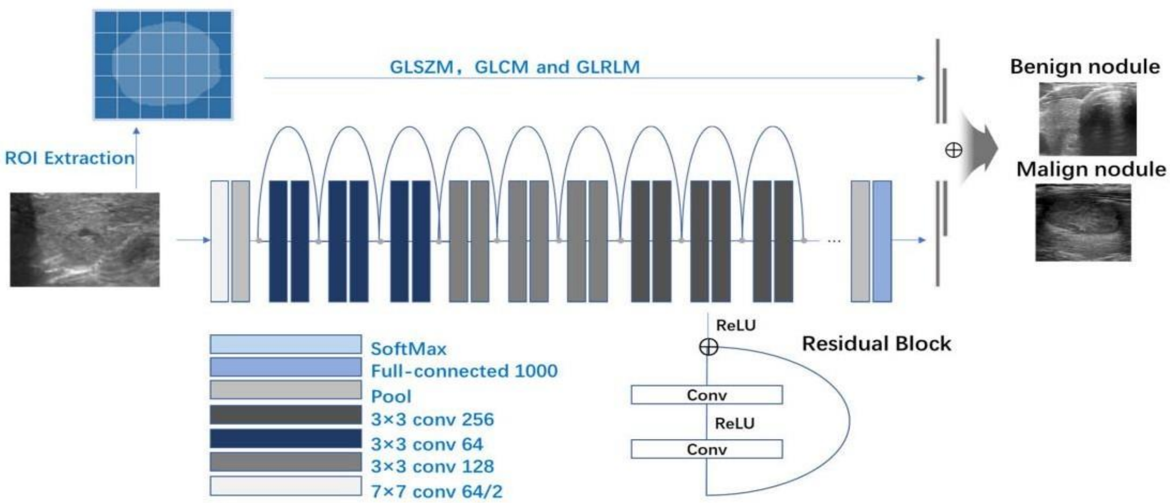

Fig 3.3.1 System architecture

3.4 Data Flow Diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. Itis used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

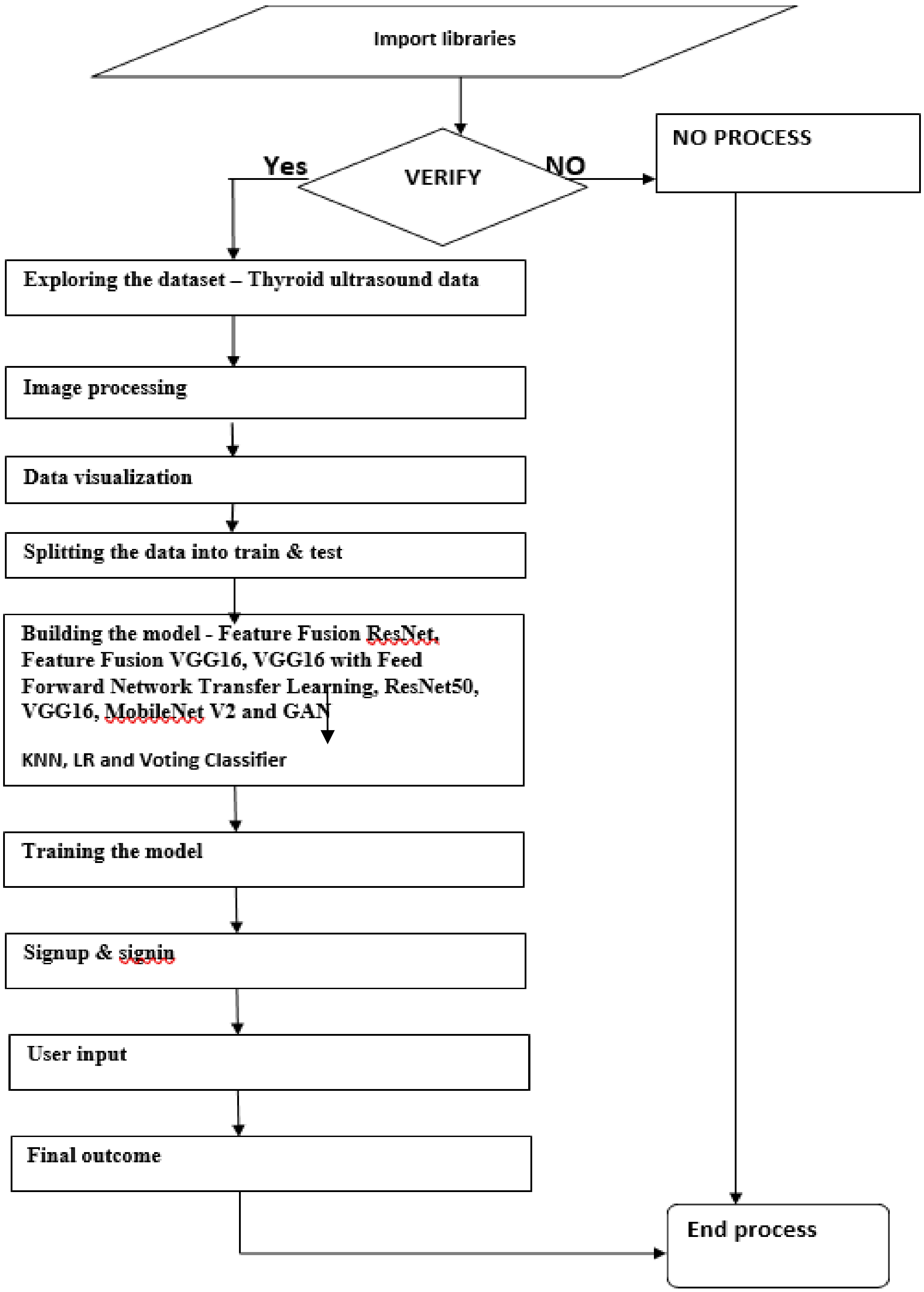

Fig 3.4.1 Data Flow Diagram

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

3.5 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by the Object Management Group.

The goal is for UML to become a common language for creating models of object- oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to or associated with UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects

3.6 Activity Diagram

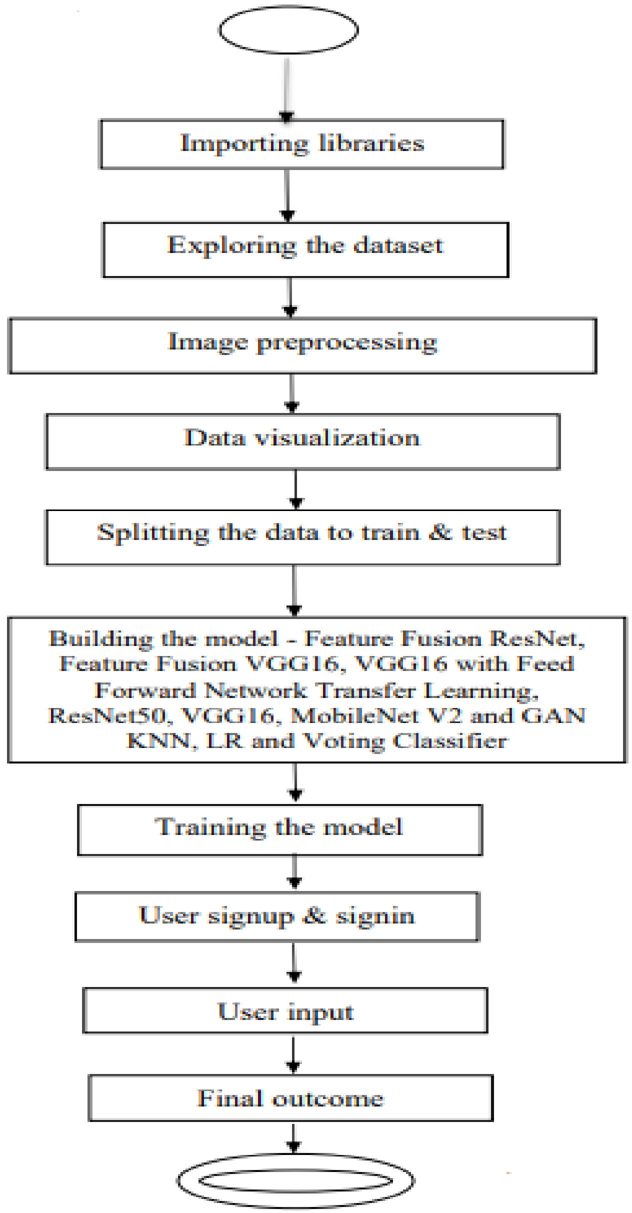

Fig 3.6.1 Activity Diagram

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

3.7 Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

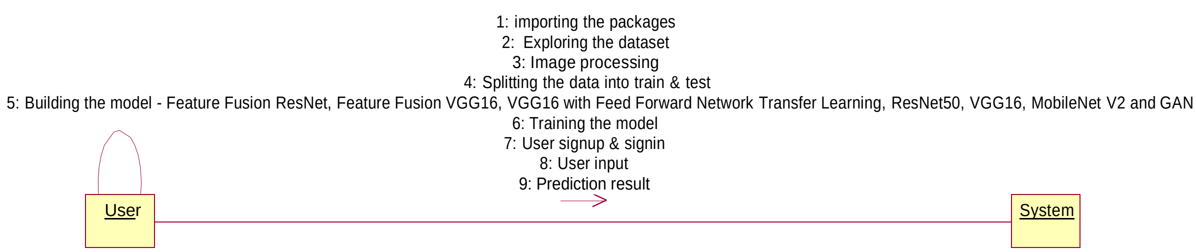

Fig.3.7.1 Collaboration diagram

3.8 Deployment diagram

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

Fig 3.8.1 Deployment diagram

3.9 Sequence diagram

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

=

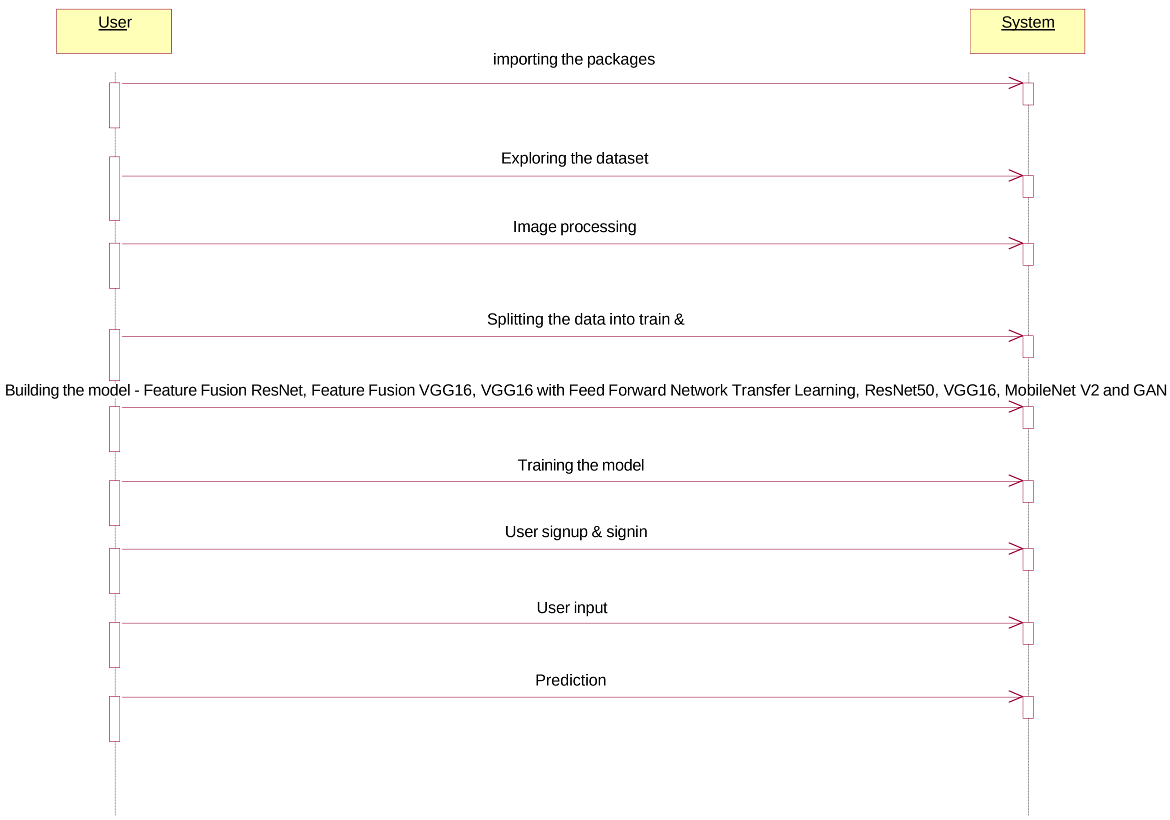

Fig 3.9.1 Sequence diagram

3.10 Component diagram

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.

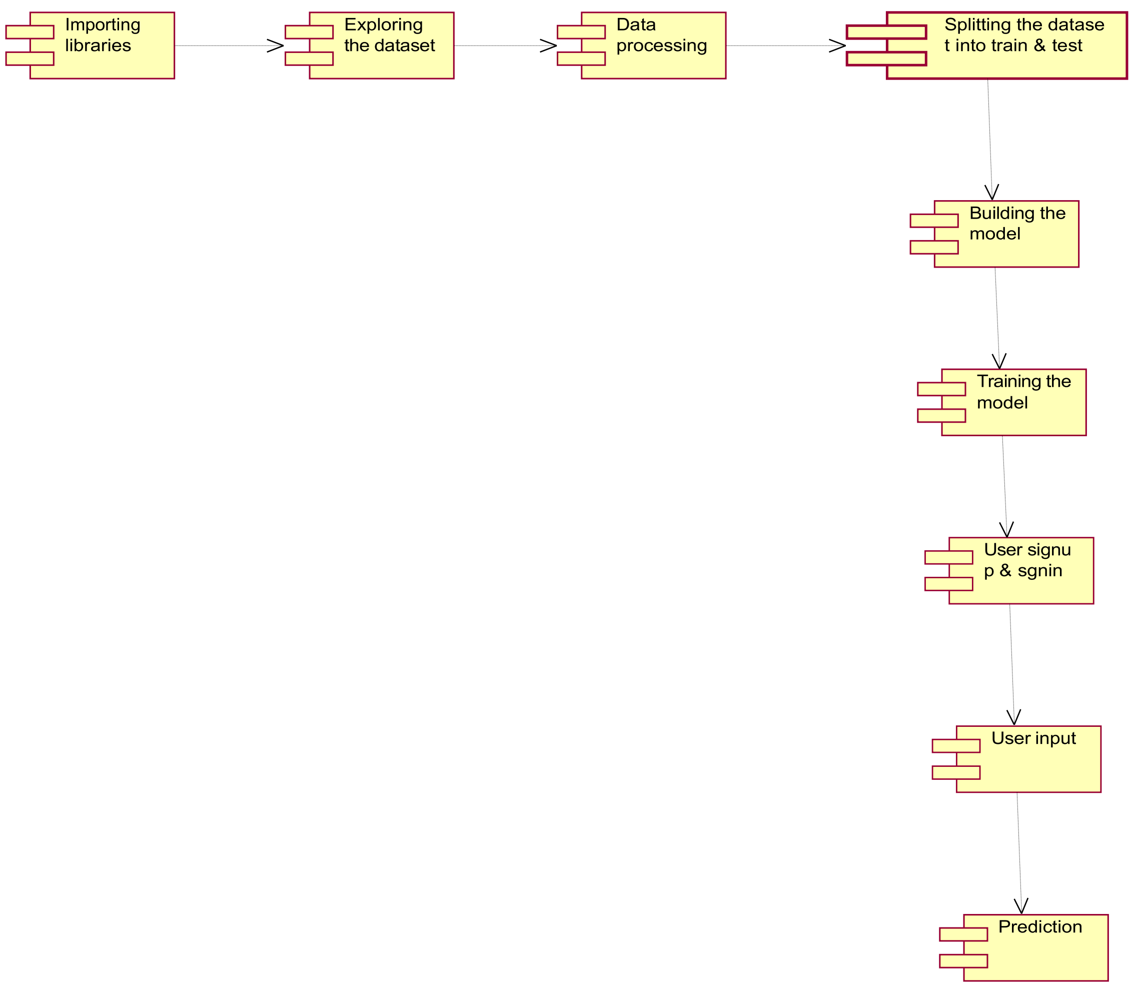

Fig 3.10.1 Component diagram

Chapter-4
IMPLEMENTATION & WORKING

4.1 Modules

- Data exploration: using this module we will load data into system
- Processing: Using the module we will read data for processing
- Splitting data into train & test: using this module data will be divided into train & test
- Model generation: Building the model - Feature Fusion Res Net, Feature Fusion VGG16, VGG16 with Feed Forward Network Transfer Learning, ResNet50, VGG16, Mobile Net V2 and GAN, KNN , LR and Voting Classifiers, Algorithms accuracy calculated
- User signup & login: Using this module will get registration and login
- User input: Using this module will give input for prediction
- Prediction: final predicted displayed

4.2 Algorithms

Feature Fusion ResNet: Feature fusion refers to the fusion of feature vectors of training images extracted from shared weight network layer and feature vectors composed of other numerical data, so that the proposed model can utilize features as many as possible for the further classification. ResNet is an artificial neural network that introduced a so-called “identity shortcut connection,” which allows the model to skip one or more layers. This approach makes it possible to train the network on thousands of layers without affecting performance.

Feature Fusion VGG16: VGG-16 is a convolutional neural network that is 16 layers deep.

You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

VGG16 with Feed Forward Network Transfer Learning: VGG16 is a 16-layer transfer learning architecture and is quite similar to earlier architectures as its foundation is based on CNN only but the arrangement is a bit different. The standard input image size which was taken by the researchers for this architecture was 224*224*3 where 3 represents the RGB channel.

ResNet50: ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

VGG16: VGG-16 is a convolutional neural network that is 16 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

MobileNet V2: MobileNet-v2 is a convolutional neural network that is 53 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

GAN: A generative adversarial network (GAN) is a machine learning (ML) model in which two neural networks compete with each other to become more accurate in their predictions. GANs typically run unsupervised and use a cooperative zero-sum game framework to learn.

KNN: The k-nearest neighbours algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

LR: Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

Voting Classifiers: A voting classifier is a machine learning estimator that trains various base models or estimators and predicts on the basis of aggregating the findings of each base estimator. The aggregating criteria can be combined decision of voting for each estimator output.

4.3 Libraries / Packages

4.3.1 Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and productionat Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

4.3.2 Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

4.3.3 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

4.3.4 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and I Python shells, the Jupiter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with I Python. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object-oriented interface or via a set of functions familiar to MATLAB users.

4.3.5 Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

4.3.6 System Testing

System testing, also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

4.4 Software Testing Strategies:

Optimization of the approach to testing in software engineering is the best way to make it effective. A software testing strategy defines what, when, and how to do whatever is necessary to make an end-product of high quality. Usually, the following software testing strategies and their combinations are used to achieve this major objective:

The early-stage testing strategy is static testing: it is performed without actually running the developing product. Basically, such desk-checking is required to detect bugs and issues that are present in the code itself. Such a check-up is important at the pre-deployment stage as it helps avoid problems caused by errors in the code and software structure deficits.

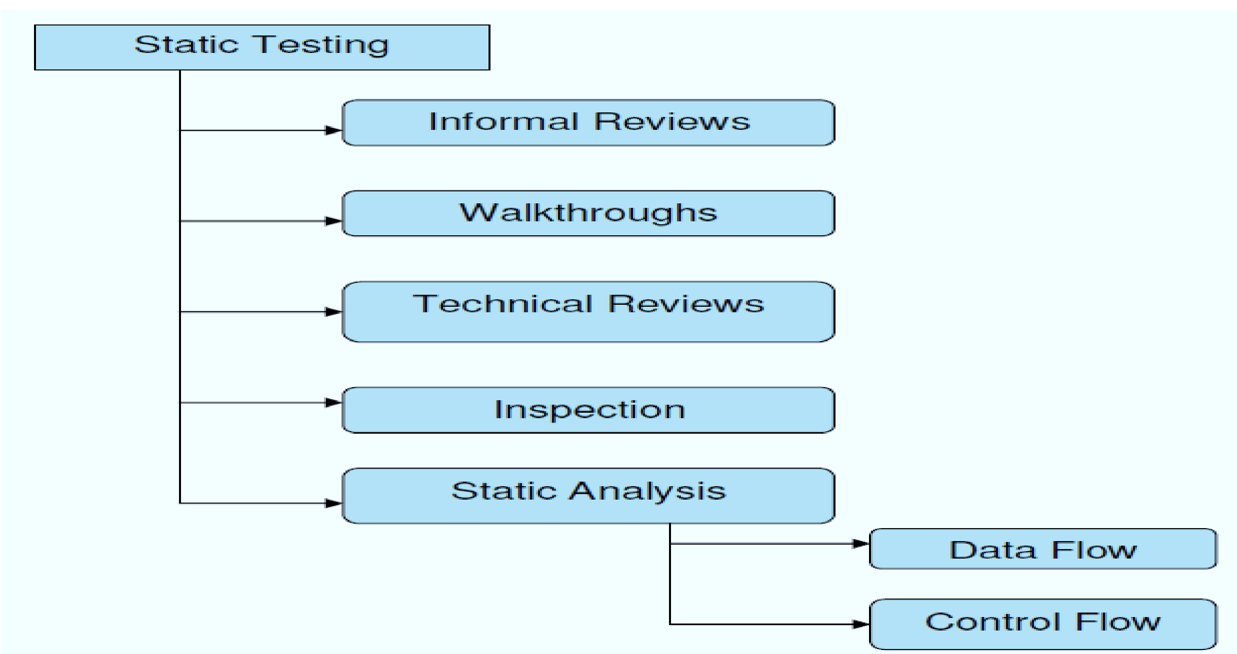

Fig 4.4.1 Software testing

4.5 Structural Testing:

It is not possible to effectively test software without running it. Structural testing, also known as white-box testing, is required to detect and fix bugs and errors emerging during the pre-production stage of the software development process. At this stage, unit testing based on the software structure is performed using regression testing. In most cases, it is an automated process working within the test automation framework to speed up the development process at this stage. Developers and QA engineers have full access to the software's structure and data flows (data flows testing), so they could track any changes (mutation testing) in the system's behavior by comparing the tests' outcomes with the results of previous iterations (control flow testing).

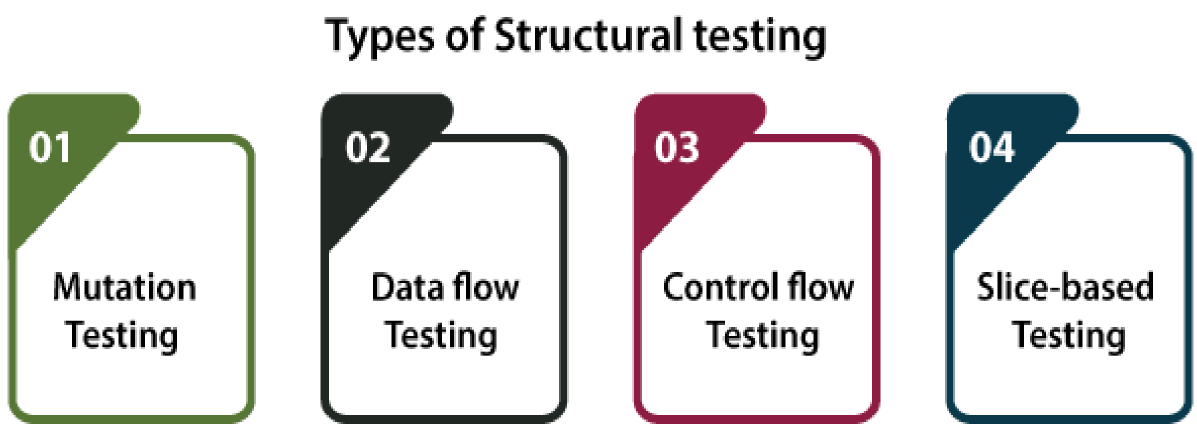

Fig 4.5.1 Structural testing

4.6 Behavioral Testing:

The final stage of testing focuses on the software's reactions to various activities rather than on the mechanisms behind these reactions. In other words, behavioral testing, also known as black-box testing, presupposes running numerous tests, mostly manual, to see the product from the user's point of view. QA engineers usually have some specific information about a business or other purposes of the software ('the black box') to run usability tests, for example, and react to bugs as regular users of the product will do. Behavioral testing also may include automation (regression tests) to eliminate human error if repetitive activities are required. For example, you may need to fill 100 registration forms on the website to see how the product copes with such an activity, so the automation of this test is preferable.

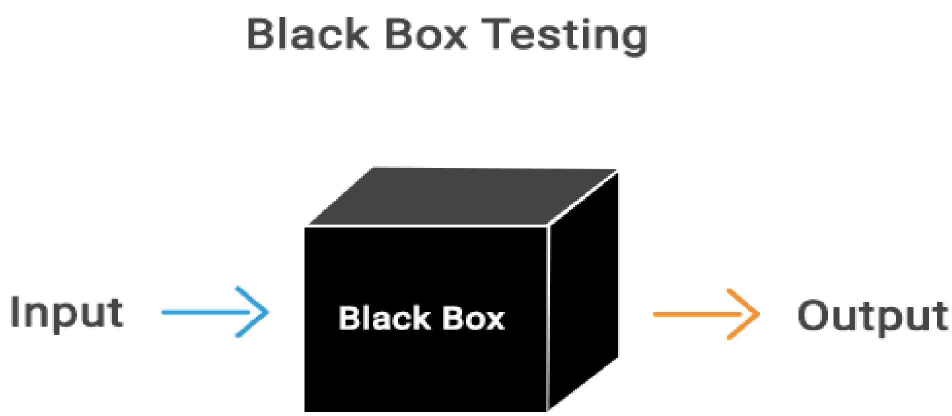

Fig 4.6.1 Behavioral testing

Chapter 5
RESULTS & DISCUSSION

5.1 TEST CASES:

S.NO	INPUT	If available	If not available
1	User signup	User get registered into the application	There is no process
2	User sign in	User get login into the application	There is no process
3	Enter input for prediction	Prediction result displayed	There is no process

The significant results show that there are significant differences between them, so the proposed method has obvious improvement. Comparing the five methods, Feature Fusion ResNet has reached the highest value in most indicators. In addition, some models have also achieved better results on some indicators. For example, VGG-16 model obtained 99.19% Recall and 87.54% F-score, but it did not perform well on Accuracy and Precision. ResNet model reached the lowest performance, and the four indicators were significantly lower than many methods. Performance of VGG-Net and ResNet are different from each other.

Chapter 6
CONCLUSION

Since clinical diagnosis of benign and malignant thyroid nodules by ultrasound is a subjective and tedious process, this work aims to assist doctors in making clinical diagnosis of thyroid nodules, thereby improving the accuracy and efficiency of diagnosis. Firstly, it is necessary to preprocess the clinically collected data, including cropping, enhancement, and extraction of regions of interest. Then, feature engineering is applied to obtain texture features of nodules based on the nodules area, and feature dimensionality reduction is realized through the correlation between features and nodules. Finally, a deep neural network model is established, and texture features from the previous step are merged to achieve the goal of further improving network performance. Under assessment of 1874 cases with thyroid nodules, this method obtained the best performance, which has clinical potential. This work combines the advantages of feature engineering and deep neural networks, and proposes a novel way of fusing features. Although this work mainly validates the diagnostic performance of ultrasound imaging of thyroid nodules, this work can also be applied to various domains under the transfer learning and fusion feature structure, such as breast nodules, lung nodules and other tumor diagnosis. It is worth mentioning that the method of fusing features is mainly to introduce more features and information for deep neural network, so that the network can converge more accurately and quickly. This is also a future direction for new fusion information. This work has certain inspirations for computer aided diagnosis, application of deep convolutional networks, and image analysis.

REFERENCES

[1] J. Kim, J. E. Gosnell, and S. A. Roman, ‘‘Geographic influences in the global rise of thyroid cancer,’’ Nature Rev. Endocrinol., vol. 16, no. 1, pp. 17–29, Jan. 2020.

[2] H. R. Shahraki, S. Pourahmad, S. Paydar, and M. Azad, ‘‘Improving the accuracy of early diagnosis of thyroid nodule type based on the SCAD method,’’ Asian Pacific J. Cancer Prevention, vol. 17, no. 4, pp. 1861–1864, Jun. 2016.

[3] J. Xu, M. Jing, S. Wang, C. Yang, and X. Chen, ‘‘A review of medical image detection for cancers in digestive system based on artificial intelligence,’’ Expert Rev. Med. Devices, vol. 16, no. 10, pp. 877–889, Oct. 2019.

[4] H. Ye, J. Hang, X. Chen, D. Xu, J. Chen, X. Ye, and D. Zhang, ‘‘An intelligent platform for ultrasound diagnosis of thyroid nodules,’’ Sci. Rep., vol. 10, no. 1, Aug. 2020, Art. no. 13223.

[5] L. Wang, S. Yang, S. Yang, C. Zhao, G. Tian, Y. Gao, Y. Chen, and Y. Lu, ‘‘Automatic thyroid nodule recognition and diagnosis in ultrasound imaging with the YOLOv2 neural network,’’ World J. Surg. Oncol., vol. 17, no. 1, p. 12, Jan. 2019.

[6] C. Chen, L. Zhan, X. Pan, Z. Wang, X. Guo, H. Qin, F. Xiong, W. Shi, M. Shi, F. Ji, Q. Wang, N. Yu, and R. Xiao, ‘‘Automatic recognition of auditory brainstem response characteristic waveform based on bidirectional long short-term memory,’’ Frontiers Med., vol. 7, Jan. 2021, Art. no. 613708.

[7] D. Koundal, S. Gupta, and S. Singh, ‘‘Computer aided thyroid nodule detection system using medical ultrasound images,’’ Biomed. Signal Process. Control, vol. 40, pp. 117–130, Feb. 2018.

[8] D. Koundal, S. Gupta, and S. Singh, ‘‘Automated delineation of thyroid nodules in ultrasound images using spatial neutrosophic clustering and level set,’’ Appl. Soft Comput., vol. 40, pp. 86–97, Mar. 2016.

[9] Y. Zheng, S. Xu, Z. Zheng, L. Wu, L. Chen, and W. Zhan, ‘‘Ultrasonic classification of multicategory thyroid nodules based on logistic regression,’’ Ultrasound Quart., vol. 36, no. 2, pp. 149–157, Jun. 2020.

[10] W. Sun, S. Xie, J. Yu, L. Niu, and W. Sun, ‘‘Classification of thyroid nodules in ultrasound images using deep model based transfer learning and hybrid features,’’ in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2017, pp. 919–923.