A Project Report on

**CANCER DETECTION USING CONVOLUTIONAL NEURAL NETWORK**

*is submitted in partial fulfillment of the requirement for the award of**the Degree of*

***Bachelor of Technology***

*to*

****

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU**

**by**

**K.RADHAKRISHNA REDDY**

**(17711A0567)**

**H.MOHAMMED NAUMAN P.VENKATA NITHISH REDDY**

**(17711A0565) (17711A0589)**

**K.VISHNU VARDHAN SK.SUHAIL BASHA**

**(17711A0569) (17711A05A1)**

**Under the Guidance of**

**Dr.K.M.Vara Prasad,** M.Tech, PhD

**ASSOCIATE PROFESSOR**

****

Department of Computer Science and Engineering

****

**JULY 2021**

****

**Department of Computer Science and Engineering**

**CERTIFICATE**

This is to certify that the project report entitled **“CANCER DETECTION USING CONVOLUTIONAL NEURAL NETWORK”** being submitted by **K.RADHAKRISHNA REDDY (17711A0567), H.MOHAMMED NAUMAN (17711A0565), P.VENKATA NITHISH REDDY (17711A0589), K.VISHNU VARDHAN (17711A0569), SK.SUHAIL BASHA (17711A05A1)** in partial fulfillment for the award of the Degree ofBachelor of Technology in Computer science and Engineering Department to the Jawaharlal Nehru Technological University Anantapur, Ananthapuramu is a record of bonafied work carried out by them under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr.K.M.VARA PRASAD, M.Tech, PhD Dr.C.RAJENDRA, M.Tech, Ph.D.

Associate Professor HOD & Professor

Project Supervisor Department of CSE

Date of Viva-Voce\_\_\_\_\_\_\_\_\_\_\_

INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We are extremely grateful to **Dr. P. NARAYANA, Ph.D. Founder**, Narayana Educational Institutions, Andhra Pradesh for the kind blessings. We are extremely thankful to **Mr.R.Sambasiva Rao**, **Registrar,** Narayana Educational Group, Andhra Pradesh.

We are much obliged to **Dr. A.V.S Prasad, Ph.D. Director,** Narayana Engineering &Pharmacy Colleges, for the continuous encouragement and support. We owe indebtedness to our **Principal Dr.G.Srinivasulu Reddy, M.Tech., Ph.D.,** Narayana Engineering College, Nellore for providing us the required facilities.

We express our deep sense of gratitude and sincere thanks to **Dr.C.Rajendra, M.Tech, PhD, HOD,** Department of Computer Science and Engineering, Narayana Engineering College, Nellore for providing the necessary facilities and encouragement towards the project work.

We thank our project guide, **Dr.K.M.Varaprasad, Associate Professor, Dept of CSE,** for his guidance, valuable suggestions and support in the completion of the project.

We gratefully acknowledge and express our thanks to teaching and non-teaching staff of C.S.E Department. We would like to express our love and affection to our parents for their encouragement throughout this project.

### Project Associates

K.Radha Krishna (17711A0567)

H.Md.Nauman (17711A0565)

P.Nithish (17711A0589)

K.Vishnu Vardhan (17711A0569)

Sk.Suhail Basha (17711A05A1)

**CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER NAMES** | **PAGE.NO** |
| **ABSTRACT** | i |
| **LIST OF FIGURES** | ii |
| **1.INTRODUCTION** |  |
| 1.1 Motivation | 3 |
| 1.2 Aim | 3 |
| 1.3 Objectives | 4 |
| **2. LITERATURE SURVEY** |  |
| 2.1 Machine Learning | 5 |
| 2.2 Research Papers Referred | 6 |
| 2.2.1 Breast Cancer detection Using Convolutional Neural  Networks for Mammogram Imaging System | 6 |
| 2.2.2 Cancer type prediction based on copy number aberration | 7 |
| 2.2.3 Deep Learning based Identification of Cancer | 8 |
| 2.2.4 Identification of Cancer Subtypes by Integrating Multiple  Types of Transcriptomic Data | 9 |
| 2.2.5 Early Detection of Cancer: Past, Future and Present | 11 |
| 2.2.6 Image denoising via combination Anisotropic diffusion  And Bilateral filtering | 12 |
| 2.2.7 Molecular characterization of soft tissue tumours: a gene  expression study | 13 |
| **3.SYSTEM ANALYSIS** |  |
| 3.1 Existing system | 15 |
| 3.2 Proposed system | 16 |
| 3.3 Problem statement | 16 |
| 3.3.1 Solution | 17 |
| 3.3.2 Algorithm used | 17 |
| **4. SYSTEM DESIGN** |  |
| 4.1 Data Flow Diagram | 28 |
| 4.2 UML Diagrams | 30 |
| 4.2.1 Class Diagram | 30 |
| 4.2.2 Sequence Diagram | 31 |
| 4.3 Control Flow Diagram | 32 |
| 4.4 System Architecture | 34 |
| **5. SYSTEM IMPLEMENTATION** |  |
| 5.1 Module Description | 37 |
| 5.2 Algorithm Analysis | 38 |
| 5.3 Technologies And Tools Used | 40 |
| 5.3.1 Python | 40 |
| 5.3.2 Package Introduction | 40 |
| 5.3.3 Jupyter Notebook | 41 |
| **6. CODING** | 42 |
| **7. TESTING** |  |
| 7.1 System Testing | 56 |
| 7.1.1 Test Objectives | 56 |
| 7.1.2 Features to be tested | 57 |
| 7.2 Types Of Tests | 58 |
| 7.2.1 Unit testing | 58 |
| 7.2.2 System Testing | 58 |
| 7.3 Testing Random Images within the image data sets | 60 |
| **8.EXPERIMENTAL RESULTS** | 64 |
| **9.CONCLUSION AND FUTURESCOPE** |  |
| 9.1 Conclusion | 67 |
| 9.2 Future scope | 67 |
| **REFERENCES** | 69 |

**CANCER DETECTION USING**

**CONVOLUTIONAL NEURAL NETWORKS**

**ABSTRACT**

Cancer has been distinguish as a dangerous disease that has many different sub categories. Machine learning is used widely in detection and diagnosis of cancer among patients. It is important to classify cancer patients into high or low risk groups, So many research teams from biomedical field have studied the application of machine learning methods. These techniques have been utilized as an aim to model the progression and treatment of cancerous conditions. In addition, the ability of Machine learning tools to trace key features from complex datasets reveals their importance. Here we are using convolutional neural networks to predict the cancer qualities. Convolutional neural network is useful in analyzing visual imagery. In this project, we use Convolutional Neural Network models that take unstructured gene expression inputs such as images to classify tumors and non-tumor samples into their cog nominate cancer types or as normal.

**Keywords:** Cancer, Machine learning, CNN

**LIST OF FIGURES**

|  |  |
| --- | --- |
| FIGURE NAME | PAGE.NO |
| Fig 2.1 Machine learning flow | 5 |
| Fig 3.3.2(a) CNN Algorithm | 18 |
| Fig 3.3.2(b) This figure illustrates the Convolution | 18 |
| Fig 3.3.2(c) Filter/Kernel | 20 |
| Fig 3.3.2(d) Filter (green) slides over the input image | 20 |
| Fig 3.3.2(e) Convolution Steps | 21 |
| Fig 3.3.2(f) Squashed one-depth channel convolution feature | 23 |
| Fig 3.3.2(g) Convolution Process | 23 |
| Fig 3.3.2(h) Max Pool | 24 |
| Fig 3.3.2(I) Average pool | 24 |
| Fig 3.3.2(j) Fully connected model | 25 |
| Fig.4.1 Data flow diagram | 29 |
| Fig 4.2.1 Class Diagram | 31 |
| Fig 4.2.2 Sequence Diagram | 32 |
| Fig 4.3 Control flow diagram of cancer prediction | 34 |
| Fig 4.4 System architecture | 35 |
| Fig 7.1 Random Image 1 | 60 |
| Fig 7.2 Random Image 2 | 60 |
| Fig 7.3 Random Image 3 | 61 |
| Fig 7.4 Random Image 4 | 61 |
| Fig 7.5 Random Image 5 | 62 |
| Fig 7.6 Random Image prediction 1 & 2 | 62 |
| Fig 7.7 Random Image prediction 3 & 4 | 63 |
| Fig 7.8 Random Image prediction 5 | 63 |
| Fig 8.1 Input Page of The Model | 64 |
| Fig 8.2 Selecting the image in the model | 64 |
| Fig 8.3 Image Selected | 65 |
| Fig 8.4 Output of Benign Type Cancer | 65 |
| Fig 8.5 Output of Invasive type cancer | 66 |
| Fig 8.6 Output of Insitu type cancer | 66 |

**CHAPTER 1**

**INTRODUCTION**

Cancer is a group of diseases involving abnormal [cell growth](https://en.wikipedia.org/wiki/Cell_growth) with the potential to invade or spread to other parts of the body. Cancer is a dangerous disease that characterized by the nature of the cell inside the body which has no control. Cancer is not a single disease but rather more than 100 types of disease. Cancer damages the human body gradually when cells starts growing uncontrollably to form many lumps of tissue inside the human body called tumours. Tumours may grow and interact with the other parts of the body. That parts may be nervous system, digestive system or circulatory system. The effect of infected parts of the body releases the hormones that causes change in the body. Mainly There are two types of cancer: Malignant and Benign

Malignant is spread into the surrounding tissue. Cell can grow to other cell and destroy the surrounding tissue that causes other tumor to develop. So malignant tumor can be a life-threatening and more dangerous in nature. Benign tumor usually do not cause much damage but can become more dangerous if they grow a lot or they might become malignant after certain amount of time.

There are four stages of cancer. They are as follows:

* Normal
* Benign
* Insitu
* Invasive

The cancer cells grow as a minute community in the normal stage. In benign stage, the cancer cells grow abnormally in only a particular tissue/organ. It is also called as a tumor. Insitu stage can either be converted into benign stage or invasive stage.

Cancer is the second leading cause of death worldwide, an average of one in six deaths is due to cancer. Considerable research efforts have been devoted to cancer diagnosis and treatment techniques to lessen its impact on human health. Cancer prediction’s major focus is on cancer susceptibility, recurrence, and prognosis, while the aim of cancer detection is the classification of tumor types and identification of markers for each cancer such that we can build a learning machine to identify specific metastatic tumor type or detect cancer at their earlier stage. With the increased awareness of precision medicine and early detection techniques matured over years of technology development, including particularly many detection screens achieving a sensitivity around 70-80%, the demand for applying novel machine learning methods to discover new biomarkers has become one of the key driving factors in many clinical and translational applications. Among all the types of cancer, the very popular and influenced is Breast cancer. The main risk factors of breast cancer include sex, obesity, less physical exercise, intakes of alcohol, hormonal misbalance during menopause, ionizing radiation, pre- menstruation, children at later age or not at all, and older age. The above factors are not the common factors. There may be some another reason to cause breast cancer with different stages or spread, aggressiveness and genetic makeup.

It would be nice to have a system that would allow to detect and prevent the cancer at an early stage. This can increase the survival rates for those who are going to effect of breast cancer. To prognosis and diagnoses cancer by a physician may become difficult by seeing the human body until their cells are treated. Further a research is required to diagnose and prognosis the cancer in a human body by biomedical and bioinformatics field. Hence Machine Learning is a field of AI in computer science which can implement many computational intelligent techniques for the prediction of cancer at early stage if data of patient are collected. In the view of providing better treatment to the patient, it is important to precisely predict different type of tumours.

Cnn’s enables computer system to analyze and classify the data. When applied to images cnn can recognize that an image shows as an animal or plant…etc. CNN system has also been developed to help medical department do their work including selecting cellular elements on pathological sides, correctly identifying the spatial orientation of some radiographs.

There have been many machine learning applications regarding medical field.

Firstly CNN is not one process. It’s actually a complex network of interconnected processes organized in layers. With each layer, the CNN can detect higher-level, more abstract features. When the CNN is identifying these features, it uses something called a filter.

A medical image in radiology or ophthalmology or dermatology is characterized by local structure like curves, corners, textures etc. These CNN filters does constituting little miniature versions of each little building blocks. And the way that the CNN looks for these building blocks is the C in CNN which stands for convolution. It’s a mathematical operation that looks pretty complex. It maintains the hierarchy like hierarchy from letters to documents. Here documents are at the complex stage when compared to the letters. So CNN is meant to reduce the complexity.

* 1. **MOTIVATION**

With the advancement in today’s technology, we can easily predict the presence of cancer in any organ just by providing data to the system, rather than predicting it manually. False cancer detection is also present in modern diagnosis. If we diagnose the presence of cancer manually that is with the help of an expert sometimes it may goes wrong in predicting the tumor.

Early detection of cancer is very important for successful treatment. There are few methods available to detect cancerous cells. Automatic method for cancer prediction Convolutional Neural Networks is implemented. Convolutional Neural Network models that take unstructured gene expression inputs to classify tumor and non-tumor samples into their designated cancer types or as normal. It takes high dimension cancer scanned image inputs and perform cancer type prediction while considering their tissue of origin. It helps in providing accurate results, automatic detection and also helps in reducing the false results.

**1.2 AIM**

The main aim of this project is to provide an automatic method for cancer prediction with accurate results. Here we will predict the cancer by providing the images to the system and then dividing these images into pixel by pixel. In order to capture the low level features of an image we will also use some filters like edges, colour, gradients etc. We will be using the Machine Learning algorithms to train the data. The more we train the data, the accuracy of predicting the cancer increases.

* 1. **OBJECTIVES**
* **To Reduce the false results:** Here we can get the results of the diagnosis without any false predictions. It predict the presence of the tumor in the body correctly.
* **To Increase Accuracy:** The level of accuracy in the proposed system will be higher. All operation would be done correctly and it provide us the accurate results in the detection of the tumor.
* **To Implement Automatic Detection Model:** We can detect the presence the tumor in the body automatically just by providing and training the data to the system. It does not require any experts help in predicting the cancer.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 MACHINE LEARNING:**

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model. The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. Algorithm is trained again and again with an augmented

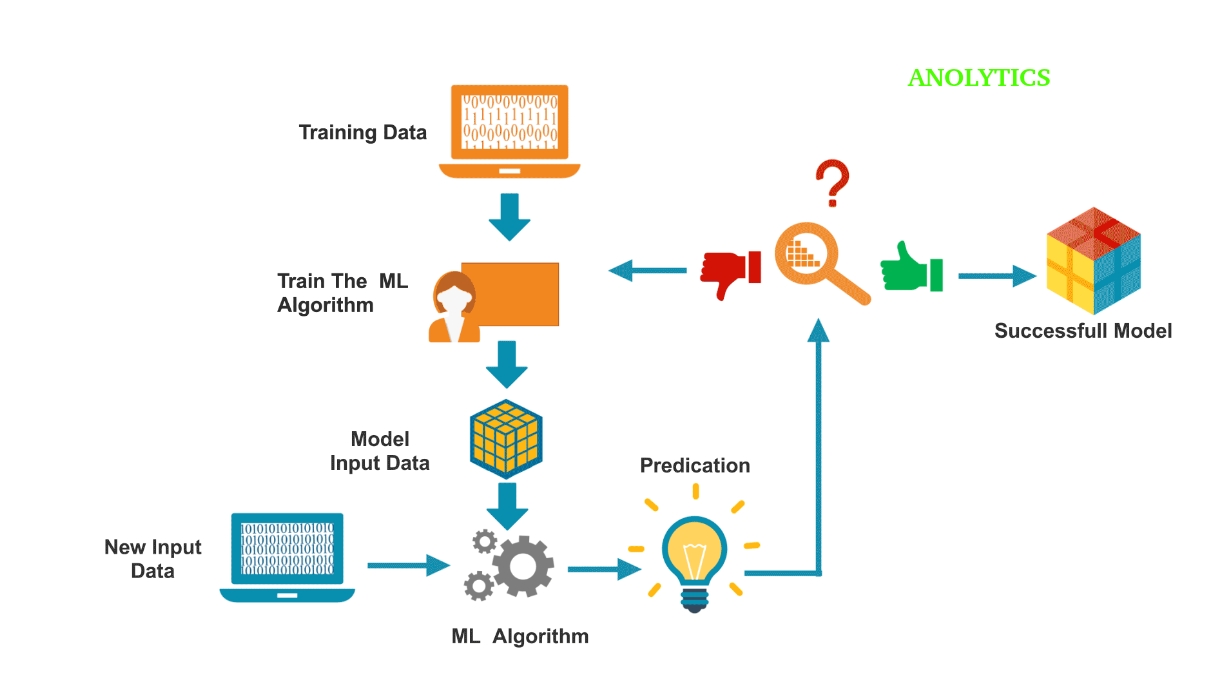


Fig 2.1 Machine learning flow

**2.2 RESEARCH PAPERS REFERRED**

**2.2.1 Breast Cancer detection Using Convolutional Neural Networks for Mammogram Imaging System**

**Theme:**

Breast cancer detection using convolutional neural network for mammogram imaging system is proposed to classify mammogram image into normal, benign (non-cancerous abnormality) and malignant (cancerous abnormality). Breast Cancer detection Using Convolutional Neural Networks (BCDCNN) is aimed to speed up the diagnosis process by assisting specialist to diagnosis and classification the breast cancer. A series of mammogram images are used to carry out preprocessing to convert a human visual image into a computer visual image and adjust suitable parameter for the CNN classifier.

**Advantages:**

* This system classifies mammogram images into 3categories: normal, benign and malignant with high rate than80%.
* The system which can assist and help the doctor or specialist nurse to speed diagnosed the mammograms, to cover shortage of specialist or time handling diagnosed.
* It has high accuracy results.

**Limitations:**

* The raw data pixel size is too large. Huge segment of times spent to adjust the input data for the classifier.

**Objectives:**

A series of mammogram images are used to carry out preprocessing to convert a human visual image into a computer visual image and adjust suitable parameter for the CNN classifier. After that, all changed images are assigned into CNN classifier as training source. The CNN classifier will then produce a model to recognize the mammogram image. BCDCNN has improved the accuracy toward classification on the mammogram images.

**2.2.2 Cancer type prediction based on copy number aberration**

**Theme:**

It used a DeepCNA, an advanced convolutional neural network (CNN) based classifier, which utilizes copy number aberrations (CNAs) and HiC data, to address this issue. DeepCNA first pre-process the CNA data by clipping, zero padding and reshaping. Then, the processed data is fed into a CNN classifier, which extracts high-level features for accurate classification. Experimental results on the COSMIC CNA dataset indicate that 2D CNN with both cell lines of HiC data lead to the best performance. We further compare DeepCNA with three widely adopted classifiers, and demonstrate that DeepCNA has at least 78% improvement of performance. Conclusions: This paper demonstrates the advantages and potential of the proposed DeepCNA.

**Advantages:**

* Facilitates high-level feature extraction.
* CNN is able to drastically lower the degree of freedom of the network, and thus reduce its overall size, making deeper networks practical.

**Limitations:**

* In existing SMCT methods however, the absence of high-level feature extraction is a major obstacle in improving the classification performance.
* Due to the intrinsic limitations, the classic machine learning classifiers do not always offer higher performances as the feature number of the input increases.

**Objectives:**

It proposes the DeepCNA method for SMCT. DeepCNA consists of two major steps. The pre-processing step regulates the CNA data with clipping, zero padding, and reshaping; while the CNN step takes the pre-processed data and generates the classification result with high-level data feature learning. Controlled variable experiments indicate that the 2D CNN with both cell lines of HiC data contributes to the optimal performance. DeepCNA with three widely adopted data classifiers, the results of which exhibit the remarkable advantages of DeepCNA, which has achieved significant performance improvements in terms of testing accuracy against the comparison methods.

**2.2.3 Deep Learning based Identification of Cancer**

**Theme:**

Deep learning has proven to show outstanding performance in resolving recognition and classification problems. It’ aim is to address the extent to which the machine can learn to recognize cancer. We integrated cancer and normal tissue data from the Gene Expression Omnibus (GEO), The Cancer Gene Atlas (TCGA), and Therapeutically Applicable Research to Generate Effective Treatments (TARGET), and Genotype-Tissue Expression (GTEx) databases, including 13,406 cancer and 12,842 normal gene expression data from 24 different tissues. s. It first trained the deep neural network (DNN) to discriminate between cancer and normal samples using various gene selection strategies and therapeutic target genes from commercial cancer panels and genes in NCI-curated cancer pathways. Interpretation method is not only a useful tool to identify cancer from gene expression data but can also contribute toward understanding the complex nature of cancer based on large public data.

**Advantages:**

* DNN has emerged as a useful gene expression-based classifier tool, as it performed generally higher than other methods, including logistic regression, ridge, lasso, elastic net, and SVM.
* Deep learning is not only a useful tool to accurately predict cancer but also to learn important characteristics of cancer.

**Limitations:**

* One of the main drawbacks of a DNN is the difficulty in interpreting the specific contribution of an individual feature to the outcome.
* This approach is not applicable for determining which genes contribute to the DNN outcome in an individual sample.

**Objectives:**

The main aim of this project is to develop an algorithm that quantifies an individual gene's contribution to the DNN output in the sample. We will first train the deep neural network (DNN) to discriminate between cancer and normal samples using various gene selection strategies and therapeutic target genes from commercial cancer panels and genes in NCI-curated cancer pathways. We also suggest systemic analyzation method to interpret trained deep neural network. We applied the method to find genes mostly contribute to classify cancer in an individual sample. The best DNN model classified cancer and normal samples with an accuracy of 0.997 in the training set and with an accuracy of 0.956 in the test set. Accumulating data and data analytic technologies can play an important role toward understanding and unraveling this complexity. The algorithm will figure out a subgroup in which the DNN considers individual genes more importantly. Based on this pattern, we believe that these samples represent an oncogene-addicted subgroup, as its cancer-like nature depends on a few genes from the DNN’s point of view.

**2.2.4 Identification of Cancer Subtypes by Integrating Multiple Types of Transcriptomic Data with Deep Learning in Breast Cancer**

**Theme:**

It is important to identify the accurate cancer subtypes to facilitate precision cancer diagnosis and personalized cancer therapy. A hierarchical deep learning framework to integrate multiple types of transcriptomic data to identify cancer subtypes. We only integrated the gene expression and transcriptome alternative splicing profiles data to ease the platform bias effect in data integration, since both of them can be obtained from RNA-Seq data. The integration of gene expression and transcriptome alternative splicing profiles not only naturally weakens the effect of systematic background bias of data sequencing platforms, but also provides more comprehensive transcriptome details of cancer subtypes. Although two types of transcriptomic data are considered at present, the intention of this work is to provide a new data type integration model, and we hope to provide an alternative to the data integration in the identification of cancer subtypes.

**Advantages:**

* A hierarchical deep learning framework to integrate multiple types of transcriptomic data to identify cancer subtypes.
* It k provides a new option of integrating multiple types of data to investigate cancer subtypes.
* The deep learning model provides a powerful tool to handle large and high dimensional genomics data without any prior knowledge

**Limitations:**

* Alternative splicing (AS) is not an effective method to identify the cancer subtypes.
* The integration of gene expression and transcriptome alternative splicing profiles not only naturally weakens the effect of systematic background bias of data sequencing platforms.

**Objectives:**

By integrating the gene expression and transcriptome alternative splicing profiles data, we can identify five cancer subtypes in breast cancer based on TCGA BRCA dataset. We only integrated the gene expression and transcriptome alternative splicing profiles data to ease the platform bias effect in data integration, since both of them can be obtained from RNA-Seq data. The integration of gene expression and transcriptome alternative splicing profiles not only naturally weakens the effect of systematic background bias of data sequencing platforms, but also provides more comprehensive transcriptome details of cancer subtypes.

**2.2.5 Early Detection of Cancer: Past, Future and Present**

**Theme:**

Screening in both healthy and high-risk populations offers the opportunity to detect cancer early and with an increased opportunity for treatment and curative intent. Unfortunately, many cancers still lack effective screening recommendations, or in some cases, the benefits from screening are

Marginal when weighed against the potential for harm, we examine the role of traditional tumor biomarkers, describe recommended imaging for early tumor surveillance, and explore the potential of promising novel cancer markers such as circulating tumor cells (CTC) and circulating tumor DNA. Consistent challenges for all of these screening tests include limited sensitivity and specificity. The risk for over diagnosis remains a particular concern in screening, whereby lesions of no clinical consequence may be detected and thus create difficult management decisions for the clinician and patient. If treatment is pursued following over diagnosis, patients may be exposed to morbidity from a treatment that may not provide any true benefit. The cost-effectiveness of screening tests also needs to be an ongoing focus. The improvement of genomic and surveillance technologies, which leads to more precise imaging and the ability to characterize blood-based tumor markers of greater specificity, offers opportunities for major progress in cancer screening.

**Advantages:**

* The early diagnosis of a particular cancer type does not necessarily lead to higher rates of cure, and potential risks.
* It is presumed that for each primary cancer there will be a typical window from the point at which ctDNA is initially detectable to when the lesion is incurable, a window that may be only a few months or may be several years, and potentially may vary widely within a particular cancer type.

**Limitations:**

* A multitude of blood-based biomarkers have been proposed as cancer screening tests, but none have yet proven to be clinically.
* ctDNA provides a more sensitive marker since it is present in over 80% of advanced cancers, including in many patients in whom CTCs are not detectable

**Objectives:**

Tumor markers have been used for decades in oncology. Tumor markers are biomarkers found in blood, urine, cerebrospinal fluid, or other body tissues that are elevated in association with cancer. However, to date, many tumor markers have demonstrated poor accuracy and effıcacy, particularly among the most prevalent cancers. The Positive Predictive Value (PPV) of a test is directly tied to cancer prevalence in the screened population. The lower the prevalence, the lower the PPV. PPV has been very high among traditional tumor markers and has led to their failure as mass cancer screening tests.

**2.2.6 IMAGE DENOISING VIA COMBINATION ANISOTROPIC DIFFUSION AND BILATERAL FILTERING**

**Theme:**

It proposed a new image demising technique based on combination of nonlinear diffusion and bilateral filtering. The proposed algorithm uses Weighted Local Variance (WLV) of the residual image to determine the texture and fine details. These regions are denoised by bilateral filtering and then, are returned to the image denoised by P-M method. In fact, the WLV have controlling role in amount of this combination. DE noising the residual image and adding it to the primary denoised image is the main novel idea of this paper. Experimental results confirm this image denoising approach is more efficient than nonlinear diffusion and the bilateral filtering while each algorithm accomplished separately.

**Advantages:**

* Anisotropic Diffusion (AD) has good performance in homogenous region and at the edges or boundaries of image.
* The AD is capable to noise removal and well preserves the edges and it can also enhance the edges in some cases.
* It has proper PSNR and well visual quality.

**Limitations:**

* The excessive smoothing affects the isotropic diffusion procedures.

**Objectives:**

In this paper we propose a new approach for image smoothing based on combination of non-linear diffusion and bilateral filtering. Weighted Local Variance of the residual image to determine the texture and fine details. These regions are denoised by bilateral filtering then, returned to the image denoised by P-M method.

**2.2.7 Molecular characterization of soft tissue tumors: a gene expression study**

**Theme:**

Analyzing gene-expression patterns of 41 soft tissue tumors with spotted cDNA microarrays. After removal of errors introduced by use of different microarray batches, the expression patterns of 5520 genes that were well defined were used to separate tumors into discrete groups by hierarchical clustering and singular value decomposition. We aimed to start molecular characterization of these rare neoplasms and to do a genome-wide search for new diagnostic markers.

**Advantages:**

* A new method for classification of soft-tissue tumors, which could improve on the method based on histological findings.
* Singular value decomposition analysis can be used to overcome bias introduced by use of different batches of arrays.
* The increase in reliability of tumor classification by genome-wide analysis compared with immunohistochemistry with limited numbers of markers.

**Limitations:**

* The classification is complicated by the fact that there are few reliable immune histochemical markers to aid in tumor sub classification or to help predict a patient’s outcome.

**Objectives:**

A new method for classification of soft-tissue tumors, which could improve on the method based on histological findings. Large numbers of uncharacterized genes contributed to distinctions between the tumors, and some of these could be useful markers for diagnosis, have prognostic significance, or prove possible targets for treatment. We have reported gene-expression profiles of 41 soft-tissue tumors with cDNA microarrays; the complete dataset is available in a searchable format on the website accompanying this report.9 we have shown that singular value decomposition analysis can be used to overcome bias introduced by use of different batches of arrays. The two methods used for removal of array bias showed strikingly similar results.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 Existing system:**

To detect the cancer the following tests, methods and techniques are required Oncological imaging is continually becoming more varied and accurate. Different imaging techniques aim to find the most suitable treatment option for each patient. Imaging techniques are often used in combination to obtain sufficient information.

The most common imaging method used to detect cancer and monitor its spread is **Computed Tomography (CT)**, which provides cross-sectional imaging by computer. CT scans are made using X-rays.

**Magnetic Resonance Imaging (MRI)** is a procedure that uses powerful magnetic fields. This does not generate ionizing radiation. Situations where MRI is used include examining cancer or sarcoma in the head and neck region.

**Positron Emission Tomography (PET)** is based on the faster metabolic rate of cancer cells compared to normal cells. With PET imaging the patient is given a radioactive tracer that is detected by scintigraphy. PET images can also be combined with CT.

**Ultra sound examination** is useful for examining the cervix, pancreas, liver and kidneys. Needle biopsies can also be taken in ultra sound examinations.

**Endoscopic examinations** are usually for inspecting the gastrointestinal tract, bronchial tubes, cervix, prostrate, bladder or head and neck region.

**In mammography,** an X-ray image is used to examine breast tumors. Mammography is also used in breast cancer screening.

**In isotopic diagnostics** a radioactive tracer is introduced into the patient’s body. The marker goes to the organ to be examined and various imaging methods can be used to determine whether the cancer has spread. Isotopic diagnostics can be used to identify the prevalence cancers such as breast, prostate and colorectal cancer. In the laboratory, doctors look at cell samples under the microscope. Normal cells look uniform, with similar sizes and orderly organization. Cancer cells look less orderly, with varying sizes and without apparent organization.

There is a machine learning algorithm called SVM (support vector machine) which is used to predict the cancer. But the main disadvantage of SVM algorithm is not suitable for large data sets.

**Disadvantages:**

False cancer detection is also present in modern diagnosis

Linked to a natural anxiety of specialists to avoid overlooking cancer at earlier stages.

**3.2 Proposed system:**

Early detection of cancer is very important for successful treatment. There are few methods available to detect cancerous cells.

Automatic method for cancer prediction based on convolutional neural networks is implemented. Convolutional Neural Network (CNN) models that take unstructured gene expression inputs to classify tumor and non-tumor samples into their designated cancer types or as normal.

**Advantages:**

* Reducing the False results
* Automatic Detection
* High Accuracy

**3.3 Problem statement**

Prediction of cancer is necessary at early stages. In existing system we are detecting cancer but not accurately because in existing system specialists are needed to predict the cancer. Sometimes they may go wrong and leads to high damage of patients.so, false detection takes place

**3.3.1 Solution**

We are implementing automatic detection technique to predict cancer at early stages. Convolutional neural network is used to predict the cancer. It is an automatic detection and gives high accuracy.

**3.3.2: Algorithm used**

A Convolutional Neural Network (CNN) is the foundation of most computer vision technologies. Unlike traditional [multilayer perceptron](https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/) architectures, it uses two operations called ‘convolution’ and pooling’ to reduce an image into its essential features, and uses those features to understand and classify the image

Convolutional Neural Networks are a bit different. First of all, the layers are organized in 3 dimensions: width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.

CNN is composed of two major parts:

Feature Extraction:

In this part, the network will perform a series of convolutions and pooling operations during which the features are detected. If you had a picture of a zebra, this is the part where the network would recognize its stripes, two ears, and four legs.

Classification:

Here, the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

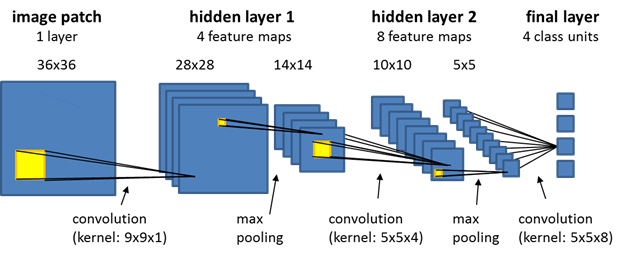


Fig 3.3.2(a) CNN Algorithm

Feature Extraction: Convolution

Convolution in CNN is performed on an input image using a filter or a kernel. To understand filtering and convolution you will have to scan the screen starting from top left to right and moving down a bit after covering the width of the screen and repeating the same process until you are done scanning the whole screen.

For instance if the input image and the filter look like following:

The filter (green) slides over the input image (blue) one pixel at a time starting from the top left. The filter multiplies its own values with the overlapping values of the image while sliding over it and adds all of them up to output a single value for each overlap until the entire image is traversed: Filter (green) slides over the input image

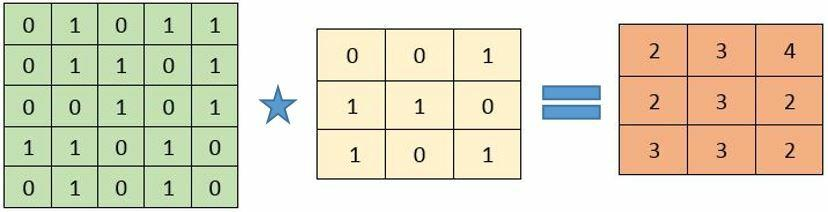


Fig 3.3.2 (b). This figure illustrates the Convolution

The convolutional layer is the first layer of the CNN which can extract features from the images by using kernel or filter. Because pixels are only related to the adjacent and close pixels, convolution operation allows us to preserve the relationship between different parts of an image. Convolution Operation is filtering the image with a smaller pixel filter or kernel to decrease the size of the image without losing the relationship between pixels. When we apply convolution operation to the 5 x 5 image by using a 3x3 filter with 1x1 stride (1-pixel shift at each step), we will end up having a 3x3 output image.

1st step of convolution

(1×1+0×1+1×1) + (0×0+1×1+1×0) + (1×0+0×0+1×1) = 4

Similarly we compute the other values of the output matrix. Note that the top left value, which is 4, in the output matrix depends only on the 9 values (3x3) on the top left of the original image matrix. It does not change even if the rest of the values in the image change. This is the receptive field of this output value or neuron in our CNN. Each value in our output matrix is sensitive to only a particular region in our original image.

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between 𝐾𝑛 and 𝐼𝑛 stack ([𝐾1, 1], [𝐾2,2],[𝐾3,𝐼3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output:

Squashed one-depth channel convolution feature

Each neuron in the output matrix has overlapping receptive fields. The animation below will give you a better sense of what happens in convolution.

Convolution example

Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc.

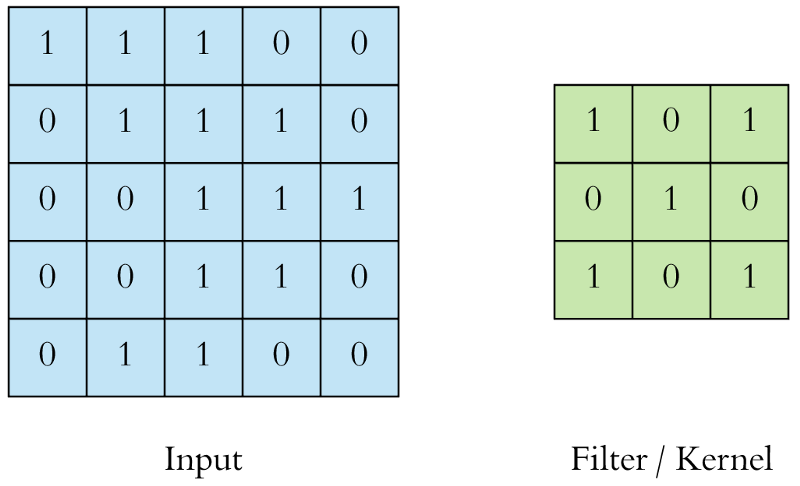


Fig 3.3.2(c) Filter/Kernel

The filter (green) slides over the input image (blue) one pixel at a time starting from the top left. The filter multiplies its own values with the overlapping values of the image while sliding over it and adds all of them up to output a single value for each overlap until the entire image is traversed:

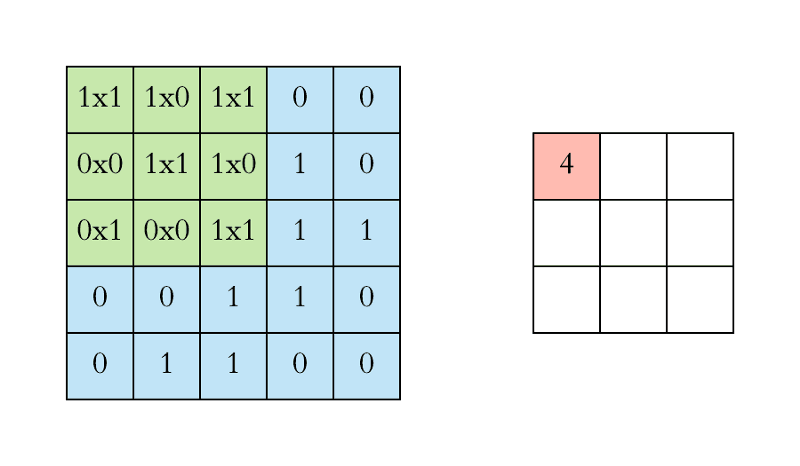


Fig 3.3.2(d) Filter (green) slides over the input image

The figure consists the value 4 (top left) in the output matrix (red) corresponds to the filter overlap on the top left of the image which is computed as:

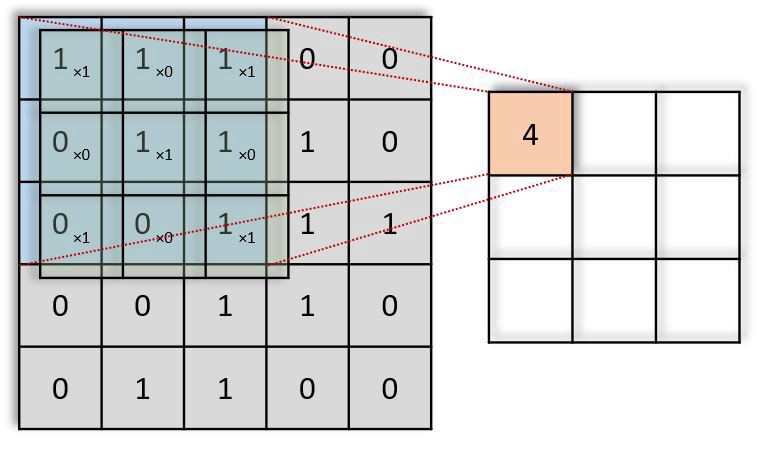


Fig 3.3.2(e) Convolution Steps

1st step of convolution

(1×1+0×1+1×1)+(0×0+1×1+1×0)+(1×0+0×0+1×1)=4

Similarly we compute the other values of the output matrix. Note that the top left value, which is 4, in the output matrix depends only on the 9 values (3x3) on the top left of the original image matrix. It does not change even if the rest of the values in the image change. This is the receptive field of this output value or neuron in our CNN. Each value in our output matrix is sensitive to only a particular region in our original image.

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between 𝐾𝑛 and 𝐼𝑛 stack ([𝐾1,1],[𝐾2,𝐼2],[𝐾3,𝐼3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output:

Each neuron in the output matrix has overlapping receptive fields. The animation below will give you a better sense of what happens in convolution.

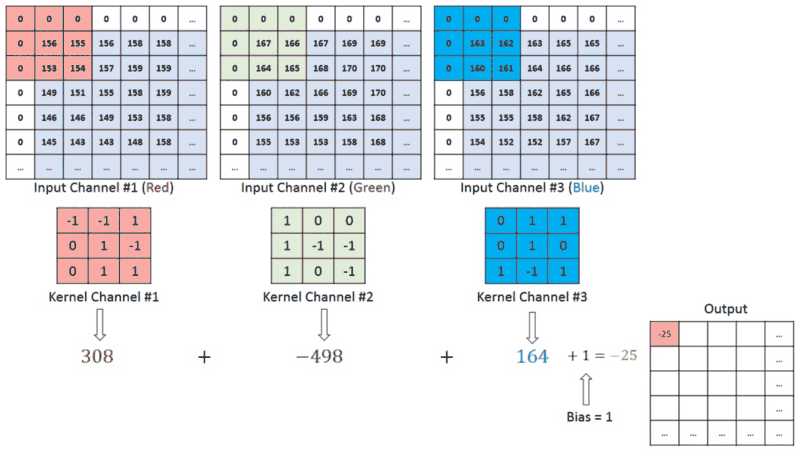


Fig 3.3.2(f) Squashed one-depth channel convolution feature

Convolution example

Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc.

Feature Extraction: padding

There are two types of results to the operation — one in which the convoluted feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding or Same Padding in the case of the latter.

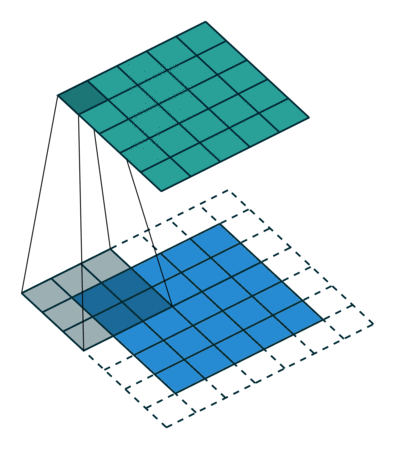


Fig 3.3.2(g) Convolution Process

Feature Extraction: Pooling

After a convolution layer once you get the feature maps, it is common to add a pooling or a sub-sampling layer in CNN layers. Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. Pooling shortens the training time and controls over-fitting.

There are two types of Pooling:

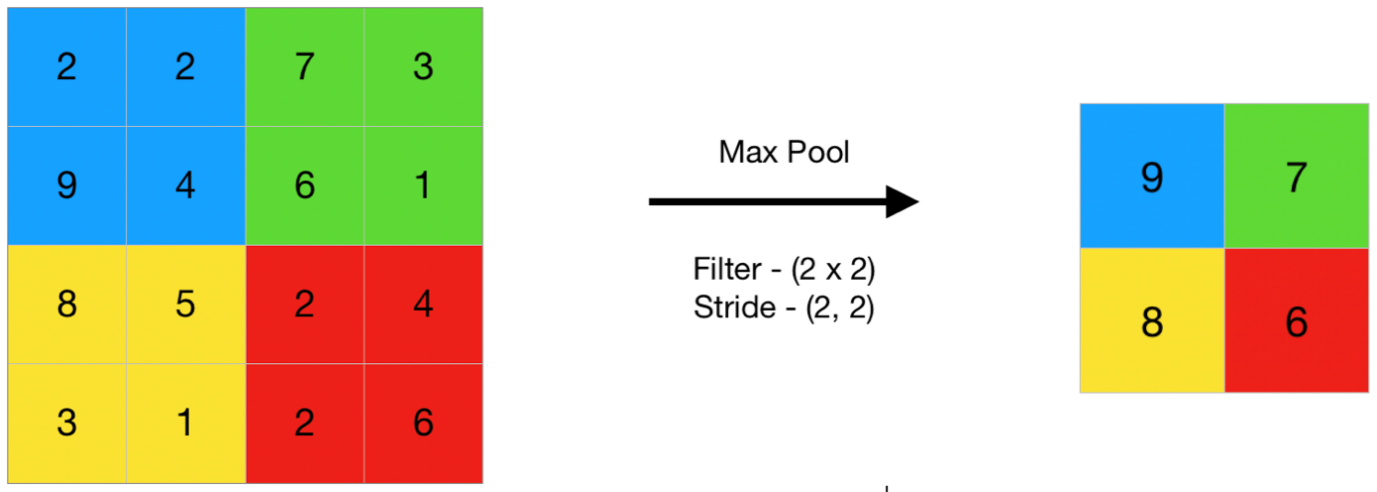


Fig 3.3.2(h) Max Pool

Max Pooling

Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel.  
Max Pooling also performs as a Noise Suppressant. It discards the noisy activation altogether and also performs de-noising along with dimensionality reduction.

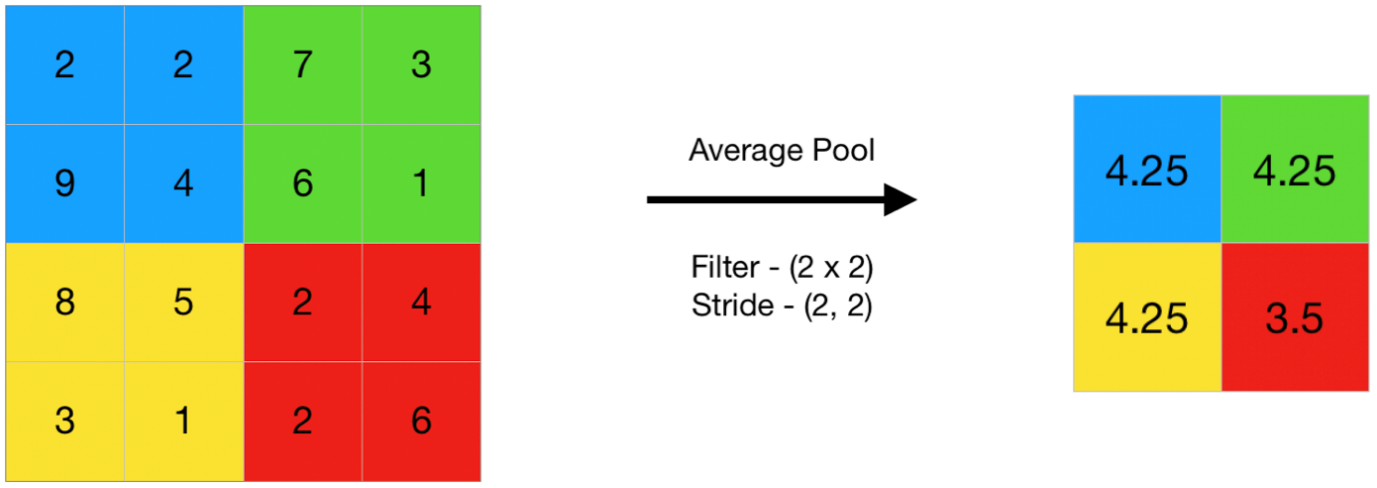


Fig 3.3.2(I) Average pool

Average Pooling

Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

Classification — Fully Connected Layer (FC Layer):

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. Example of CNN network:

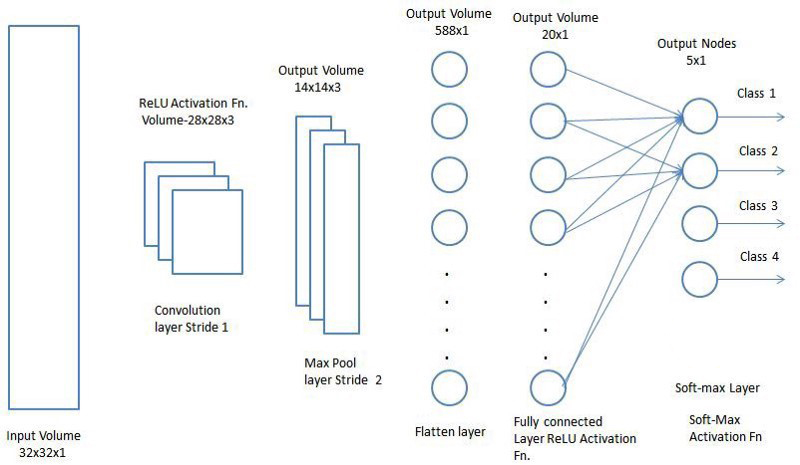


Fig 3.3.2(j) fully connected model

Now that we have converted our input image into a suitable form, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Soft ax Classification techniques

**Packages used:**

You need to have the following python packages installed

* Keras
* Tensorflow (Both CPU or GPU version should do)

**Keras:**

**KERAS** is an Open Source Neural Network library written in Python that runs on top of Thea no or Tensorflow. It is designed to be modular, fast and easy to use. It was developed by François Cholet, a Google engineer.

Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend. So Keras is high-level API wrapper for the low-level API, capable of running on top of Tensorflow, CNTK, or Thea no.

Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

**Tensor flow:**

Tensorflow is a Google-developed open source software library for high performance numerical computation. Basically, it’s a framework with a wide range of possibilities to work with Machine Learning.

Tensorflow reached high popularity because of the ease with which developers can build and deploy applications. And it has been developed in a way where you can abstract yourself sufficiently to a point that you don’t have to deal with tensors, graphs and all the math involved in creating complex models. Nevertheless, it also gives you interfaces to interact directly with its core and be able to manipulate it as well, constructing more specific models.

**Tensorflow Architecture:**

Tensorflow architecture works in three parts:

* Preprocessing the data
* Build the model
* Train and Test the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as tensors. You can construct a sort of flowchart of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. This is why it is called Tensorflow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 Data Flow Diagram:**

A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. A  DFD is a way of representing a flow of a data of a [process](https://en.wikipedia.org/wiki/Process) or a system .The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a [flowchart](https://en.wikipedia.org/wiki/Flowchart). Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical.

The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. Logical DFD depicts how the business operates. The processes represent the business activities. The data stores represent the collection of data regardless of how the data are stored. It’s how business controls. Clarifying which processes are manual and which are automate. Manual processes require detailed documentation and automated process require computer programs to be developed. Describing processes in more detail than do logical DFDs. Describes all steps for processing of data. Sequencing processes that have to be done in a particular order. Sequence of activities that lead to a meaningful result are described. Identifying temporary data storage. Temporary storage such as a sales transaction file for a customer receipt (report) in a grocery store, are described. The physical data flow diagram describes the implementation of the logical data flow. Physical DFD depicts how the system will be implemented

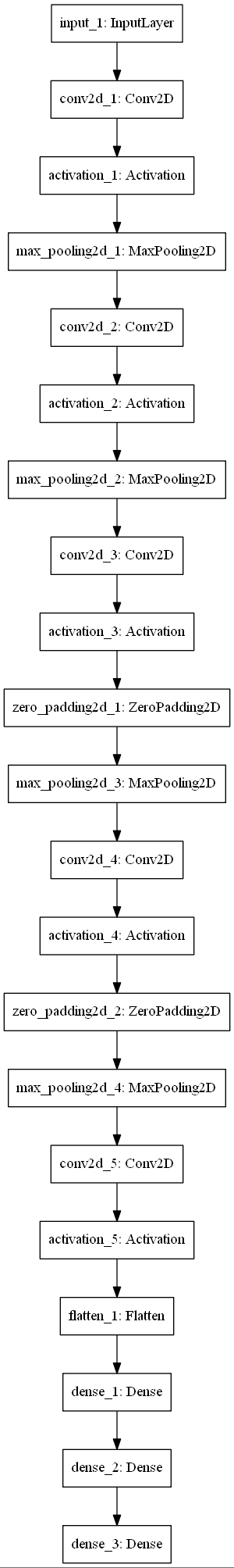
The processes represent the programs, program modules, and manual procedures. The data stores represent the physical files and databases, manual files.

Fig.4.1 Data flow diagram

It show controls for validating input data, for obtaining a record, for ensuring successful completion of a process, and for system security.

The above fig 4.1 Data flow diagram depicts the clear representation of each and every action that takes place in the system.

**4.2 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 objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

**4.2.1 Class Diagram**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

System



Name: String



Take Dataset ()



Preprocessing ()



Training ()

User



Upload testing image ()



Cancer\_detection ()

Fig 4.2.1 Class Diagram

The class diagram we have used two classes System and User. The System class has operations such as load\_dataset() for importing mammogram image dataset and assigning them to train and test variables, preprocessing() for converting the image from RGB to Gray image and normalizing it, training() for building a model with suitable convolution, pooling layers, accuracy() for evaluating the model , finding the training and testing accuracy and plotting the graph for accuracy.

**4.2.2 Sequence Diagram**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram is an interaction diagram that details how operations are carried out what messages are sent and when. Sequence diagrams are organized according to time. The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.

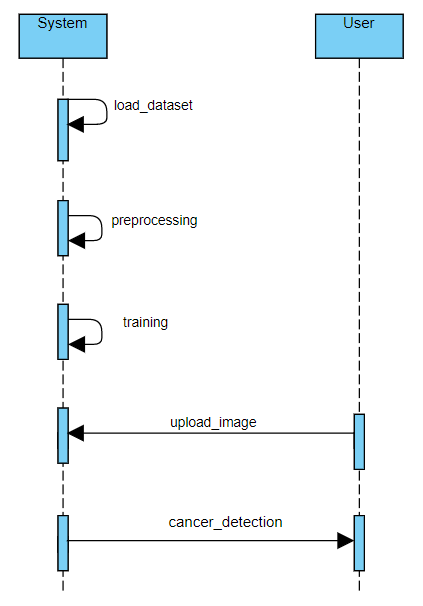


Fig 4.2.2 Sequence Diagram

**4.3 CONTROL FLOW DIAGRAM**

A control-flow diagram can consist of a subdivision to show sequential steps, with if-then-else conditions, repetition, and/or case conditions. Suitably annotated geometrical figures are used to represent operations, data, or equipment, and arrows are used to indicate the sequential flow from one to another. Control-flow diagrams can be used in [control-flow analysis](https://en.wikipedia.org/wiki/Control-flow_analysis), [data-flow analysis](https://en.wikipedia.org/wiki/Data-flow_analysis), [algorithm analysis](https://en.wikipedia.org/wiki/Algorithm_analysis), and [simulation](https://en.wikipedia.org/wiki/Simulation). Control and data are most applicable for real time and data-driven systems. These flow analyses transform logic and data requirements text into graphic flows which are easier to analyze than the text. Pert, state transition, and transaction diagrams are examples of control-flow diagrams.

The basic process of a control flow diagram is a pictorial representation of the cause and effect relationship between the input and output of a physical system. A block diagram provides a means to easily identify the functional relationships among the various components of a control system. A Control Flow Graph is the graphical representation of control flow or [computation during the execution of programs](https://www.geeksforgeeks.org/cyclomatic-complexity/) or applications. Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside of a program unit. A control flow diagram helps us understand the detail of a process. It shows us where control starts and ends and where it may branch off in another direction, given. The control flow diagram is mainly classified into two types. They are as follows:

**Process-Control-Flow Diagram:**

A flow diagram can be developed for the process [control system](https://en.wikipedia.org/wiki/Control_system) for each critical activity. Process control is normally a closed cycle in which a sensor. The application determines if the sensor information is within the predetermined data parameters and constraints. The results of this comparison, which controls the critical component. This is about the process control flow diagram which is one of the type of control flow diagram.

### Performance seeking control-flow diagram: A performance-seeking control-[flow diagram](https://en.wikipedia.org/wiki/Flow_diagram) of the algorithm. The control law consists of estimation, modeling, and optimization processes. In the inputs, outputs, and residuals were recorded. At the compact propulsion-system-modeling stage, all the estimated inlet and engine parameters were recorded. This is about the performance seeking control flow charts which is one type of control flow charts.

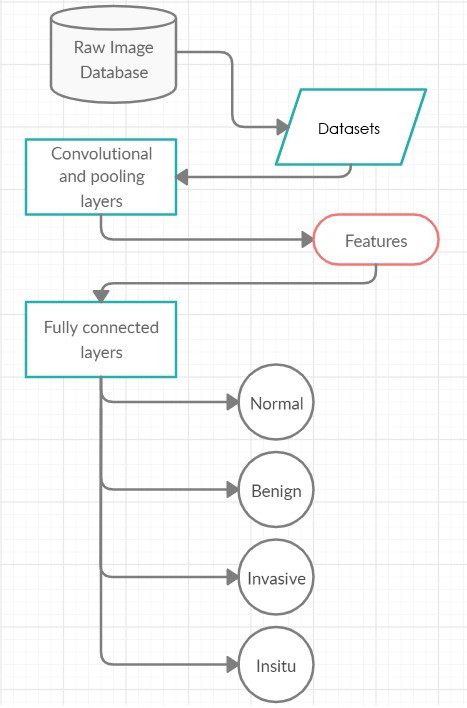
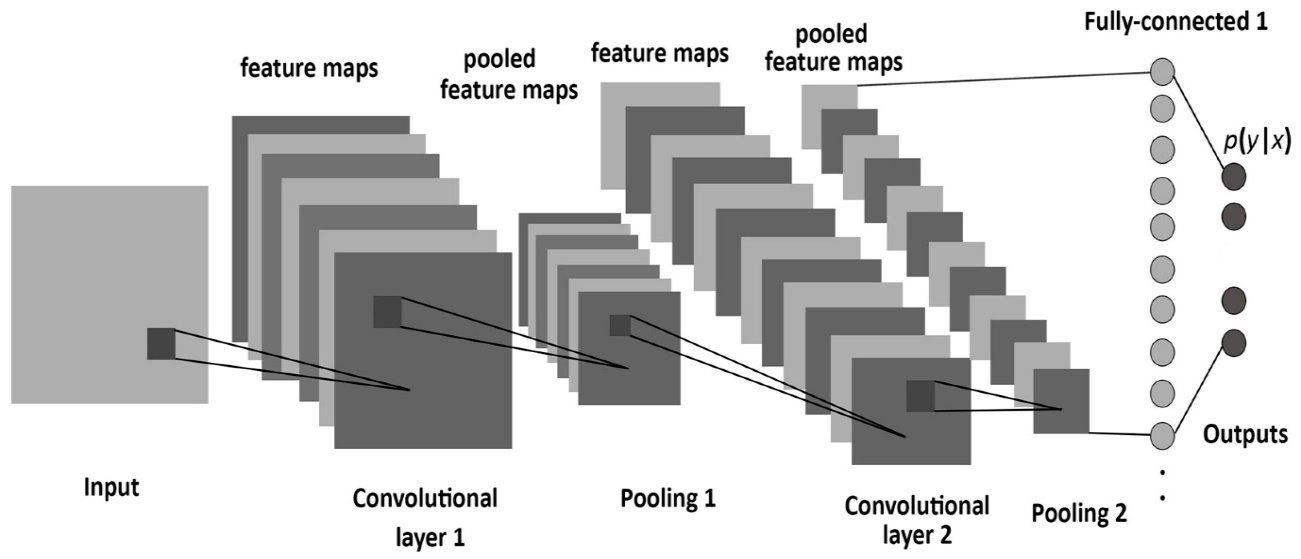


Fig 4.3 Control flow diagram of cancer prediction

The Figure represents that the CNN was trained using 4776 samples belonging to the RGB color. Therefore, the proposed system depicted in Fig. 4.3provides ineffective classification model for classifying breast tissue as being either b or malignant

**4.4 System Architecture:**

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



**Benign**

**Insitu**

**Invasive**

**normal**

Fig 4.4 System architecture

As shown in fig 4.4CNN architecture in fig 4.4 is inspired by the organization and functionality of the visual cortex and designed to mimic the connectivity pattern of neurons within the human brain.

The neurons within a CNN are split into a three-dimensional structure, with each set of neurons analyzing a small region or feature of the image. In other words, each group of neurons specializes in identifying one part of the image. CNNs use the predictions from the layers to produce a final output that presents a vector of probability scores to represent the likelihood that a specific feature belongs to a certain class.

### How a Convolutional Neural Network Works the CNN layers

A CNN is composed of several kinds of layers:

* **Convolutional layer** creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.
* **Pooling layer (down sampling)**━scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).
* **Fully connected input layer**—“flattens” the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.
* **Fully connected layer**—applies weights over the input generated by the feature analysis to predict an accurate label.
* **Fully connected output layer** generates the final probabilities to determine a class for the image.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.1 MODULE DESCRIPTION**

**Training Data:**

Training Data is labeled data used to train your machine learning algorithms and increase accuracy. To change a normal machine to a trained machine or intelligent machine, make them feed on relevant data. This is also referred to as Training data. Machine learning models are not too different from a human child. When a child observes any new object, for example a dog and receives constant feedback from its environment, the child is able to learn this new piece of knowledge.

Machines too can learn when they see enough relevant data. Using this you can model algorithms to find relationships, detect patterns, understand complex problems and make decisions. Eventually, the quality, variety, and quantity of your training data determine the success of your machine learning models.

**Test Data:**

Every machine learning model needs to be tested in the real world to measure how robust its predictions are. This is data that it has never seen before. Just as a student comes across fresh problems while in an exam, models too, need to be similarly challenged so as to evaluate their performance. The test data set contains data you are going to apply your model to. In contrast, this data doesn’t have any "expected" output. During the test phase of machine learning, this data is used to estimate how well your model has been trained and to estimate model properties. In the testing module, the system processes the input data to predict the cancer.

**5.2 ALGORITHM ANALYSIS**

**Convolutional Neural Networks:**

The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm i.e. a **Convolutional Neural Network.**

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNet have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

**Algorithm Implemented:**

Following are the steps involved in convolutional neural networks.

[**Step 1**: Convolution Operation](http://www.superdatascience.com/blogs/deep-learning-a-z-convolutional-neural-networks-cnn-step-1-convolution-operation/)

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.

[**Step 1(b):** ReLU Layer](http://www.superdatascience.com/blogs/deep-learning-a-z-convolutional-neural-networks-cnn-step-1b-relu-layer/)

The second part of this step will involve the Rectified Linear Unit or ReLU. We will cover ReLU layers and explore how linearity functions in the context of Convolutional Neural Networks. Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.

[**Step 2:** Pooling](http://www.superdatascience.com/blogs/deep-learning-a-z-convolutional-neural-networks-cnn-step-2-max-pooling/)

In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

[**Step 3:** Flattening](http://www.superdatascience.com/blogs/deep-learning-a-z-convolutional-neural-networks-cnn-step-3-flattening/)

This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

[**Step 4:** Full Connection](http://www.superdatascience.com/blogs/deep-learning-a-z-convolutional-neural-networks-cnn-step-4-full-connection/)

In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

**Step 5:** Detection

This part plays a major role in convolutional neural network. The classified image is compared with the images in the data set. After the comparison, the system will detect the type of cancer and produces the results.

**5.3 TECHNOLOGIES AND TOOLS USED**

**5.3.1 Python**

**A. Introduction**

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. sources newsgroup in 1991, and version 1.0 was released in 1994.

Here in our project, we use python 3.6 version which is efficient and effective when compared with all other versions existed in python programming language. Because it provides high security and the installation of this version is complex as it require quires and commands.

**5.3.2 Package Introduction**

Keras is a minimalist Python library for deep learning that can run on top of Thea no or Tensorflow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license. **KERAS** is an Open Source Neural Network library written in Python that runs on top of Thea no or Tensorflow. It is designed to be modular, fast and easy to use. It was developed by François Cholet, a Google engineer.

Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function. Keras doesn't handle Low-Level API such as making the computational graph, making tensors or other variables because it has been handled by the "backend" engine.

* files with custom file formats. Everything is native Python.

We can summarize the construction of deep learning models in Keras as follows:

* **Define your model**. Create a sequence and add layers.
* **Compile your model**. Specify loss functions and optimizers.
* **Fit your model**. Execute the model using data.
* **Make predictions**. Use the model to generate predictions on new data.

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.

**5.3.3 Jupyter Notebook**

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

Jupyter Notebooks are a spin-off project from the I Python project, which used to have an I Python Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the I Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also be used.

**CHAPTER 6**

**CODING**

from IPython.display import SVG

from keras.utils.vis\_utils import modfel\_to\_dot

from keras.utils import plot\_model

import tensorflow as tf

import keras

from PIL import Image,ImageTk

import keras.backend as K

import numpy as np

from keras.layers import Input, Dense, Activation, ZeroPadding2D, Flatten,

Conv2D

from keras.layers import MaxPooling2D

from keras import metrics

from keras.wrappers.scikit\_learn import KerasClassifier

from keras.applications.imagenet\_utils import preprocess\_input

from keras.models import Model

from keras.preprocessing import image as image1

from keras.models import load\_model

K.set\_image\_data\_format('channels\_last')

from matplotlib.pyplot import imshow

import os

class system:

classes=[]

@staticmethod

def pre\_process(fname):

if (fname == 'b'):

return [1, 0, 0, 0]

elif (fname == 'is'):

return [0, 1, 0, 0]

elif (fname == 'iv'):

return [0, 0, 1, 0]

else:

return [0, 0, 0, 1]

@staticmethod

def cropimages(img):

z = np.asarray(img, dtype=np.int8)

c = []

for i in range(3):

for j in range(4):

crop = z[512 \* i:512 \* (i + 1), 512 \* j:512 \* (j + 1), :]

c.append(crop)

return c

@staticmethod

def uploadimages(path):

x = []

y = []

cnt = 0

for foldname in os.listdir(path):

for filename in os.listdir(os.path.join(path, foldname)):

img = Image.open(os.path.join(os.path.join(path, foldname), filename))

crpImgs = cropimages(img)

cnt += 1

if cnt % 10 == 0:

print(str(cnt) + " Images are loading")

for im in crpImgs:

x.append(np.divide(np.asarray(im, np.float16), 255.))

y.append(pre\_process(foldname))

print("Images are cropped")

print("converting images to array")

return x, y, cnt

@staticmethod

def loadingdatsets(testNum=2):

print("Loading data set images..")

train\_set\_x\_orig, train\_set\_y\_orig, cnt = uploadimages(dataTrainPath)

testNum = 24

trainNum = (cnt \* 12) - testNum

print(testNum, trainNum)

train\_set\_x\_orig = np.array(train\_set\_x\_orig, np.float16)

train\_set\_y\_orig = np.array(train\_set\_y\_orig, np.int8)

nshapeX = train\_set\_x\_orig.shape

nshapeY = train\_set\_y\_orig.shape

print("folder trainX" + str(nshapeX))

print("folder trainY" + str(nshapeY))

print("Images loaded")

print("Loading all data")

test\_set\_x\_orig = train\_set\_x\_orig[trainNum:, :, :, :]

train\_set\_x\_orig = train\_set\_x\_orig[0:trainNum, :, :, :]

test\_set\_y\_orig = train\_set\_y\_orig[trainNum:]

train\_set\_y\_orig = train\_set\_y\_orig[0:trainNum]

classes = np.array(os.listdir(dataTrainPath))

print(train\_set\_y\_orig[0:50, :])

print(train\_set\_x\_orig[1])

print("Data load complete")

return train\_set\_x\_orig, train\_set\_y\_orig, test\_set\_x\_orig, test\_set\_y\_orig,

classes

@staticmethod

def definingmodel(input\_shape):

X\_input = Input(input\_shape)

X = Conv2D(16, (3, 3), strides=(1, 1))(X\_input)

X = Activation('relu')(X)

X = MaxPooling2D((3, 3), strides=3)(X)

X = Conv2D(32, (3, 3), strides=(1, 1))(X)

X = Activation('relu')(X)

X = MaxPooling2D((2, 2), strides=2)(X)

X = Conv2D(64, (2, 2), strides=(1, 1))(X)

X = Activation('relu')(X)

X = ZeroPadding2D(padding=(2, 2))(X)

X = MaxPooling2D((2, 2), strides=2)(X)

X = Conv2D(64, (2, 2), strides=(1, 1))(X)

X = Activation('relu')(X)

X = ZeroPadding2D(padding=(2, 2))(X)

X = MaxPooling2D((3, 3), strides=3)(X)

X = Conv2D(32, (3, 3), strides=(1, 1))(X)

X = Activation('relu')(X)

X = Flatten()(X) # Convert it to FC

X = Dense(256, activation='relu')(X)

X = Dense(128, activation='relu')(X)

X = Dense(4, activation='softmax')(X)

model = Model(inputs=X\_input, outputs=X, name='Model')

return model

@staticmethod

def train(batch\_size, epochs):

config = tf.compat.v1.ConfigProto()

sess = tf.compat.v1.Session(config=config)

tf.compat.v1.keras.backend.set\_session(sess)

model = definingmodel(X\_train.shape[1:])

model.compile('adam', 'categorical\_crossentropy', metrics=['accuracy'])

while True:

try:

model = load\_model('my\_model3.h5')

except:

print("Training a new model")

model.fit(X\_train, Y\_train, epochs=epochs, batch\_size=batch\_size)

model.save('my\_model3.h5')

preds = model.evaluate(X\_test, Y\_test\_orig, batch\_size=1, verbose=1,

sample\_weight=None)

print(preds)

print()

print("Loss = " + str(preds[0]))

print("Test Accuracy = " + str(preds[1]) + "\n\n\n\n\n")

ch = input("Do you wish to continue training? (y/n) ")

if ch == 'y':

epochs = int(input("How many epochs this time? : "))

continue

else:

break

return model

@staticmethod

def predict(img, savedModelPath, showImg=False):

model = load\_model(savedModelPath)

x = img

Image.fromarray(np.array(img, np.float16), 'RGB')

x = np.expand\_dims(x, axis=0)

softMaxPred = model.predict(x)

probs = softmaxToProbs(softMaxPred)

maxprob = 0

maxI = 0

for j in range(len(probs)):

if probs[j] > maxprob:

maxprob = probs[j]

maxI = j

return maxI, probs

@staticmethod

def softmaxToProbs(soft):

z\_exp = [np.math.exp(i) for i in soft[0]]

sum\_z\_exp = sum(z\_exp)

return [(i / sum\_z\_exp) \* 100 for i in z\_exp]

@staticmethod

def predictImage(img\_path='my\_image.jpg', arrayImg=None,

printData=False):

crops = []

if arrayImg == None:

img = image1.load\_img(img\_path)

crops = np.array(cropimages(img), np.float16)

crops = np.divide(crops, 255.)

Image.fromarray(np.array(crops[0]), "RGB")

classes = []

classes.append("Benign")

classes.append("InSitu")

classes.append("Invasive")

classes.append("Normal")

compProbs = [0,0,0,0]

for i in range(len(crops)):

if printData:

print("\n\nCrop " + str(i + 1) + " prediction:\n")

\_, probs = predict(crops[i], 'my\_model3.h5', showImg=False)

for j in range(len(classes)):

if printData:

print(str(classes[j]) + " : " + str(round(probs[j], 4)) + "%")

compProbs[j] += probs[j]

root.destroy()

root1=Tk()

root1.title("cancer results")

width, height = root1.winfo\_screenwidth(), root1.winfo\_screenheight()

root1.geometry('%dx%d+0+0' % (width,height))

image = Image.open('Insitu1.jpg')

image = image.resize((500,400), Image.ANTIALIAS)

my\_img = ImageTk.PhotoImage(image)

my\_lbl = Label(root1,image = my\_img)

my\_lbl.place(relx=0.2,rely=0.28)

Label(root1,text="REPORT",font=("Arial",30)).place(relx=0.5,rely=0.1)

Label(root1,text="Cancer Test Results\n" ,font=("Arial",20)).place(relx=0.6,rely=0.3)

Label(root1,text=str(classes[0]) + " : " + str(round(compProbs[0] / 12, 4)) + "%" ,font=("Arial",20)).place(relx=0.6,rely=0.4)

Label(root1,text=str(classes[1]) + " : " + str(round(compProbs[1] / 12, 4)) + "%" ,font=("Arial",20)).place(relx=0.6,rely=0.5)

Label(root1,text=str(classes[2]) + " : " + str(round(compProbs[2] / 12, 4)) + "%" ,font=("Arial",20)).place(relx=0.6,rely=0.6)

Label(root1,text=str(classes[3]) + " : " + str(round(compProbs[3] / 12, 4)) + "%",font=("Arial",20)).place(relx=0.6,rely=0.7)

Label(root1,text="The type of cancer detected is"+" : "+classes[\_],font=("Arial",25)).place(relx=0.3,rely=0.82)

root1.mainloop()

class user:

@staticmethod

def cancer\_detection():

ch =2

if ch == 1:

try:

classes = np.load('classes.npy')

print("Loading")

X\_train = np.load('X\_train.npy')

Y\_train = np.load('Y\_train.npy')

X\_test = np.load('X\_test.npy')

Y\_test\_orig = np.load('Y\_test\_orig.npy')

except:

X\_train, Y\_train, X\_test, Y\_test\_orig, classes =

system.loadingdatsets()

print("Saving...")

np.save('X\_train', X\_train)

np.save('Y\_train', Y\_train)

np.save('X\_test', X\_test)

np.save('Y\_test\_orig', Y\_test\_orig)

np.save('classes', classes)

print("number of training examples = " + str(X\_train.shape[0]))

print("number of test examples = " + str(X\_test.shape[0]))

print("X\_train shape: " + str(X\_train.shape))

print("Y\_train shape: " + str(Y\_train.shape))

print("X\_test shape: " + str(X\_test.shape))

print("Y\_test shape: " + str(Y\_test\_orig.shape))

model = system.train(batch\_size=100, epochs=1)

elif ch == 2:

bot=tempdir.name

system.predictImage(img\_path=bot)

else:

print("Please enter only 1 or 2")

from tkinter import

Label,Entry,Button,Tk,filedialog,PhotoImage,Canvas,Toplevel

root=Tk()

root.title("cancer")

width, height = root.winfo\_screenwidth(), root.winfo\_screenheight()

root.geometry('%dx%d+0+0' % (width,height))

image = Image.open('nadhi.jpg')

image = image.resize((width, height), Image.ANTIALIAS)

my\_img = ImageTk.PhotoImage(image)

my\_lbl = Label(image = my\_img)

my\_lbl.place(relx=0,rely=0)

Label(root,text="Cancer Detection Using Convoluton Neural

Networks",bg='salmon2',font=("Arial",30)).place(relx=0.2,rely=0.3)

Label(root,text="Click on Browse To Choose

Filename",bg='salmon2',font=("Arial",25)).place(relx=0.2,rely=0.5)

Button(root, text =

'browse',bg='ivory2',fg='black',font=("Arial",25),command=lambda:[upload\_im

age(),user.cancer\_detection()]).place(relx=0.6,rely=0.489)

tempdir=''

def upload\_image ():

global tempdir

currdir = os.getcwd()

tempdir = filedialog.askopenfile(parent=root,initialdir="/")

root.mainloop()

**CHAPTER 7**

**TESTING**

**7.1 SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product

It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

**7.1.1 Test objectives**

* **All Function Inputs Must Access Properly:**

The modules in the application should access all the inputs properly. The methods and functions which are used in the project should process the user inputs and produce the appropriate outputs effectively and efficiently.

* **Output of one function must be redirected to the identical functions:**

The output generated from one function should be the input for the function. The interlinks between the modules should be provided correctly so that the output generated from one method is directly send to another method that accepts the output as input and generates a new output.

* **The entry inputs, messages and responses must not be delayed:**

The time taken to access the inputs from the user and to generate the output should be with high speed. The processing speed of the application is mainly depends on the time complexity and space complexity of the program. The methods and functions

used in the application and the lines of code is also plays a major role in deciding the processing speed of an application.

**7.1.2 Features to be tested**

* **Verify that the correct packages are imported:**

The [import](https://docs.python.org/3/reference/simple_stmts.html#import) statement combines two operations; it searches for the named module,

then it binds the results of that search to a name in the local scope. The search operation of the import statement is defined as a call to the [\_\_import\_\_ ()](https://docs.python.org/3/library/functions.html#__import__) function, with the appropriate arguments. The return value of [\_\_import\_\_ ()](https://docs.python.org/3/library/functions.html#__import__) is used to perform the name binding operation of the import statement. See the import statement for the exact details of that name binding operation.

* **The functions should follow the syntax:**

Before you run a program in any language, you should check it for correct syntax. Since it does not force you to compile a program before running it, the syntax check is optional, but if the program has errors, it will run in correctly. It only takes a few seconds to syntax check a program.

When the syntax check is completed, you will be presented with a list of errors if there are any. Clicking on an error will cause you to jump to that error so you can fix it. You can also syntax check a list of programs all at once. This is a good thing to do before putting a new application into production, to ensure no errors have crept in.

* **Exceptions should be handled correctly:**

The core advantage of exception handling is to maintain the normal flow of the application, An exception normally disrupts the normal flow of the application that is why we use exception handling.

We can handle exceptions still without having catch blocks also, only thing you need to do is declare the throws clause in your method signature, so that the calling function would handle the exception. Before throwing exception, it executes the finally block.

**7.2 TYPES OF TESTS**

**7.2.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

**Test Results**: In System class the load\_dataset() successful loaded the Mammogram Image dataset and split the dataset into training and testing. The preprocessing()unit successfully preprocessed the image by converting into Gray image and normalizing them. The training() unit has built the model successfully without missing any layers .The accuracy() unit has successfully evaluated the model and accuracy of the model is given as output along with graph between accuracy and epochs.

In the User class the upload\_image() successfully taken the x-ray images uploaded by the user. The Cancer\_detection() has successfully get the required result about the type of cancer percentage present in the organ.

**7.2.2 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing.

System testing ensures that the tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**Test Results**: When we try to access the system class system from the user class function they are successfully invoked and hence it is supported by the system.

**7.3 Testing Random Images within the image data sets**

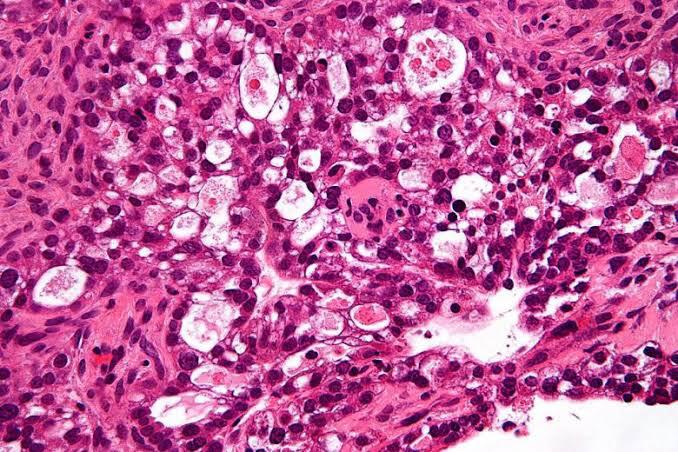


Fig 7.1 Random Image 1

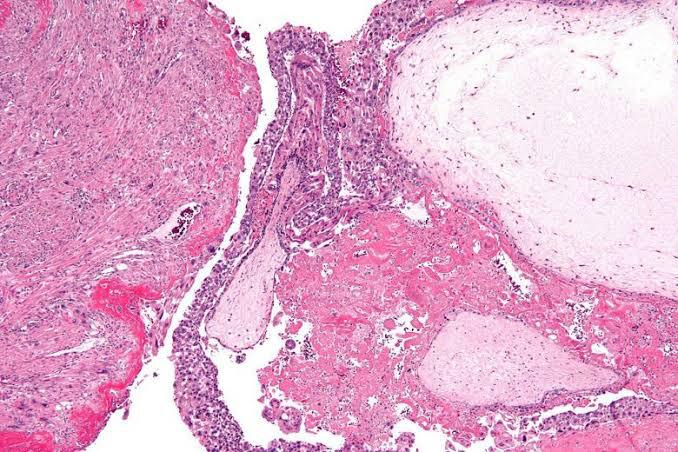


Fig 7.2 Random Image 2

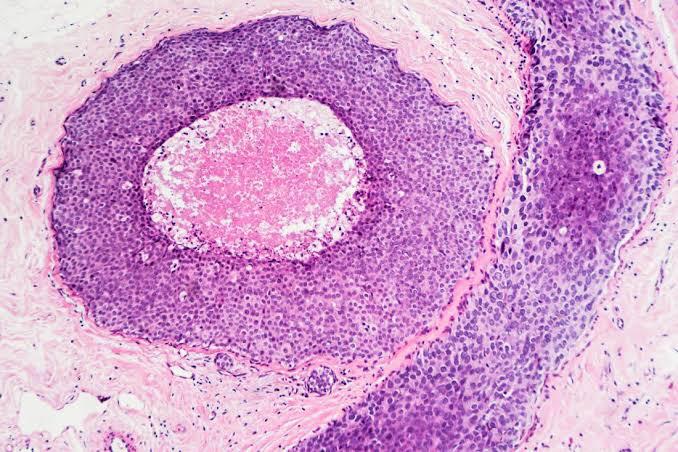


Fig 7.3 Random Image 3

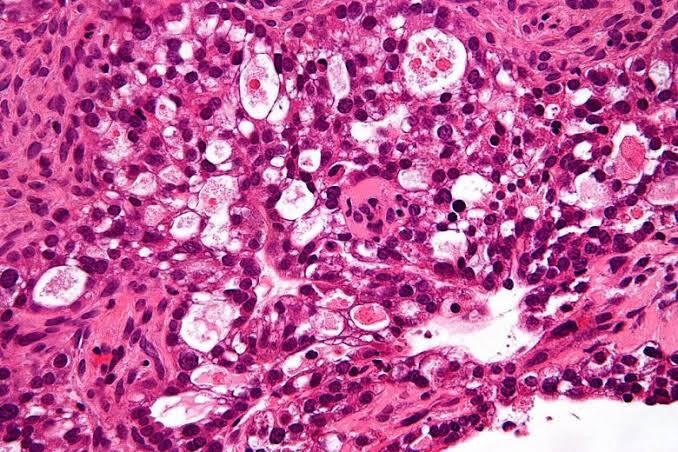


Fig 7.4 Random Image 4



Fig 7.5 Random Image 5

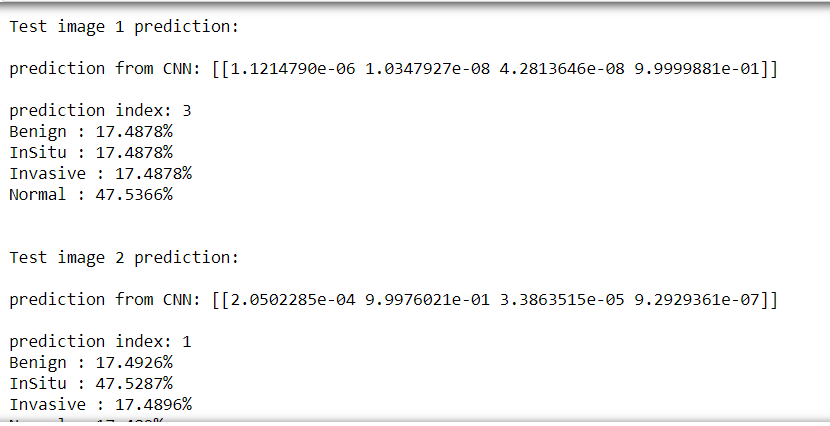


Fig 7.6 Random Image prediction 1 and 2

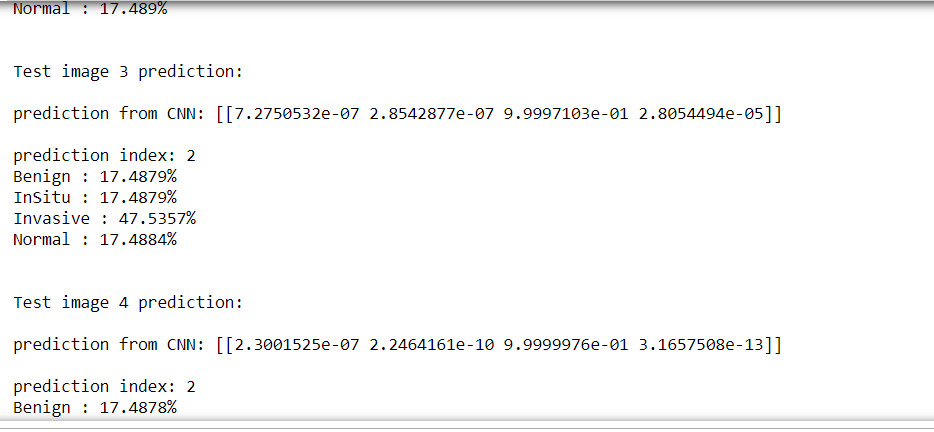


Fig 7.7 Random Image prediction 3 and 4

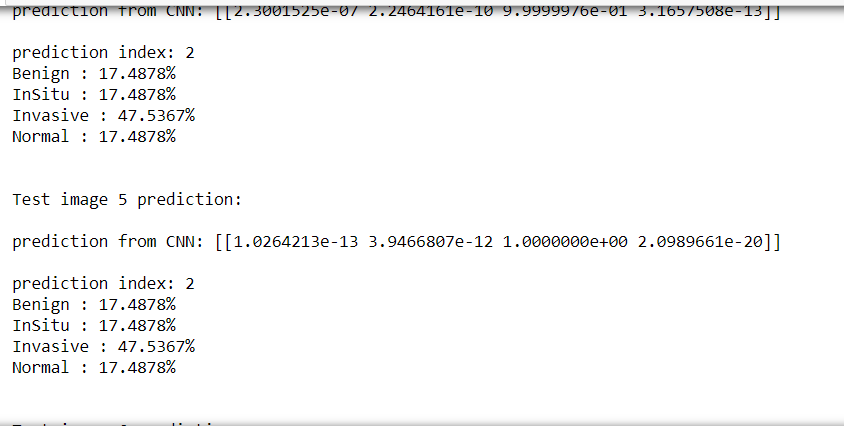


Fig 7.8 Random Image prediction 5

Here we have taken sample images randomly and have tested using our model. Based on the resultant output we can declare that the efficiency of this project is 95.6%.

**CHAPTER 8**

**EXPERIMENTAL RESULTS**

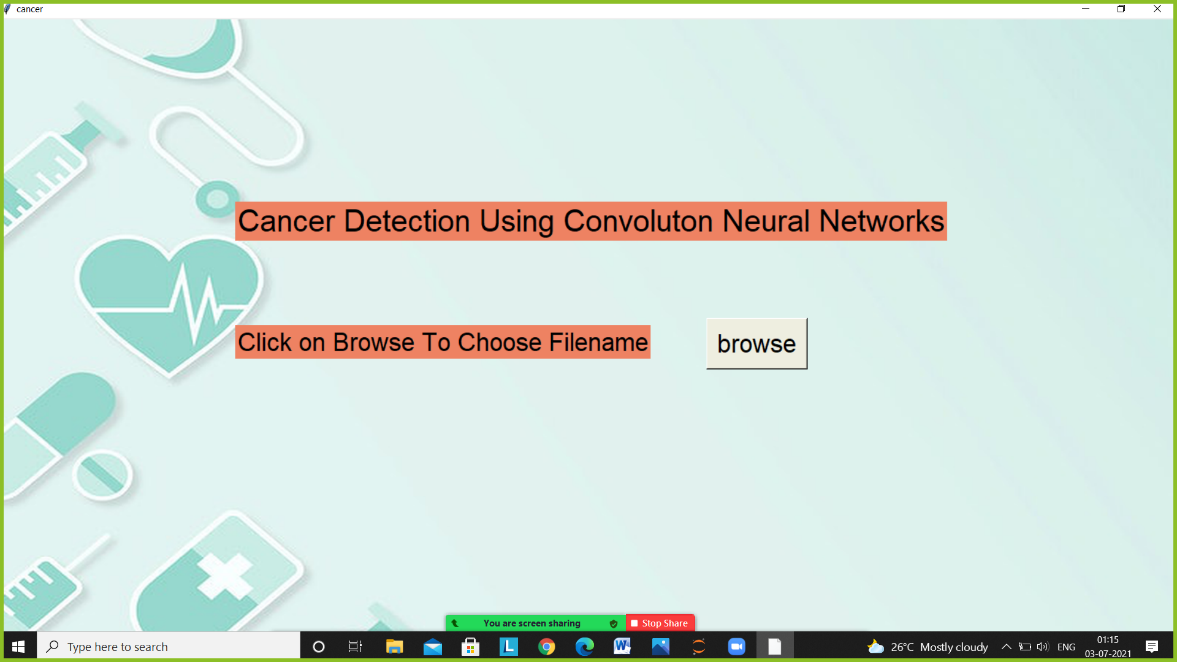


Fig 8.1 : Input Page of The Model

In this page the user needs to select the browse button and then the user can browse and select the image in the computer.

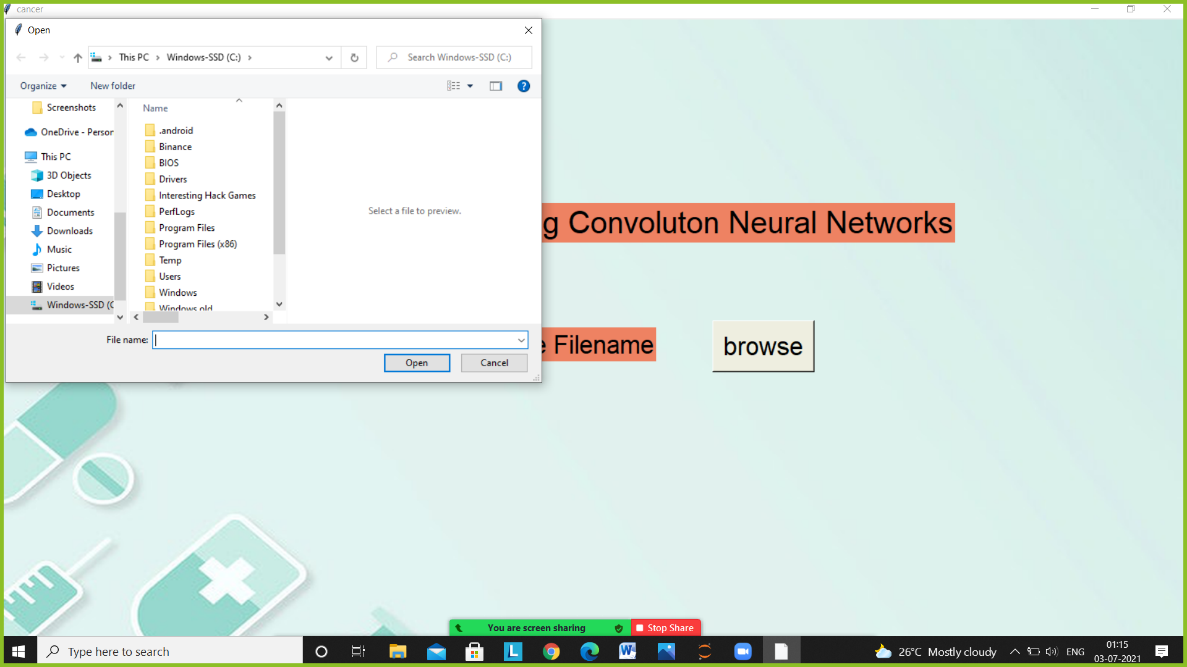


Fig 8.2 : Selecting the image in the model

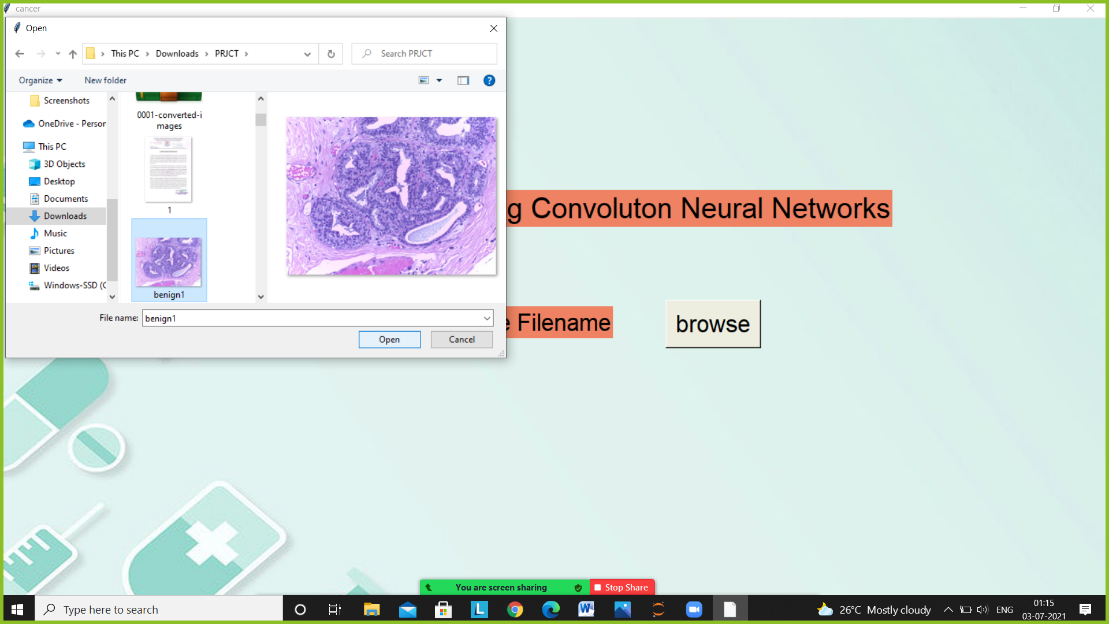


Fig 8.3 : Image Selected

After the image has been selected then it would show the required results related to the cancer.

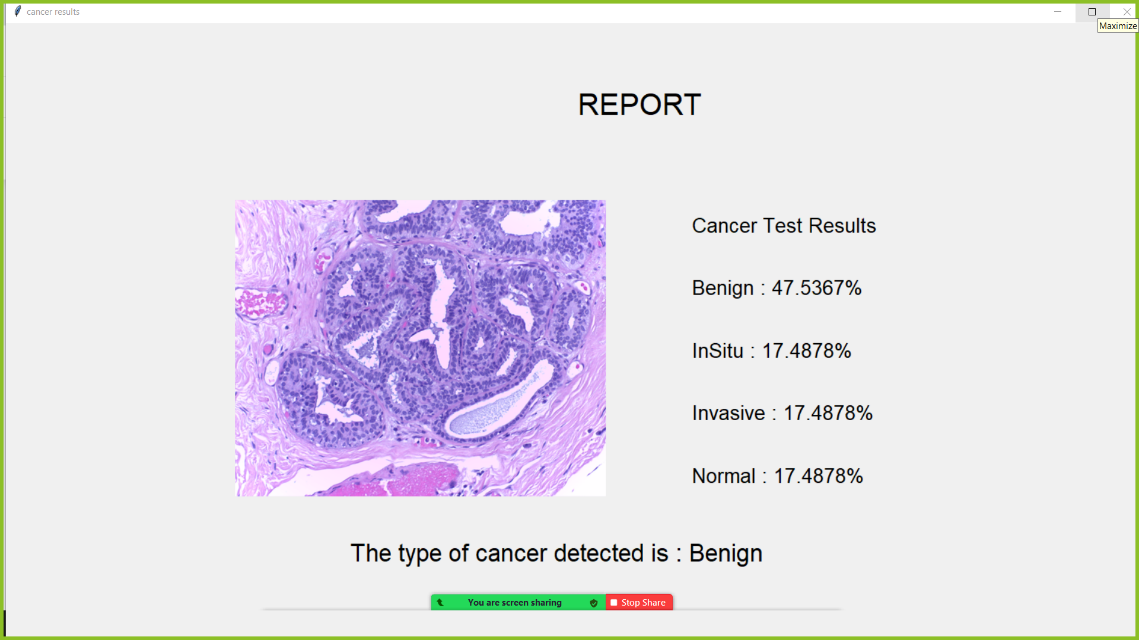


Fig 8.4 : Output of Benign Type Cancer

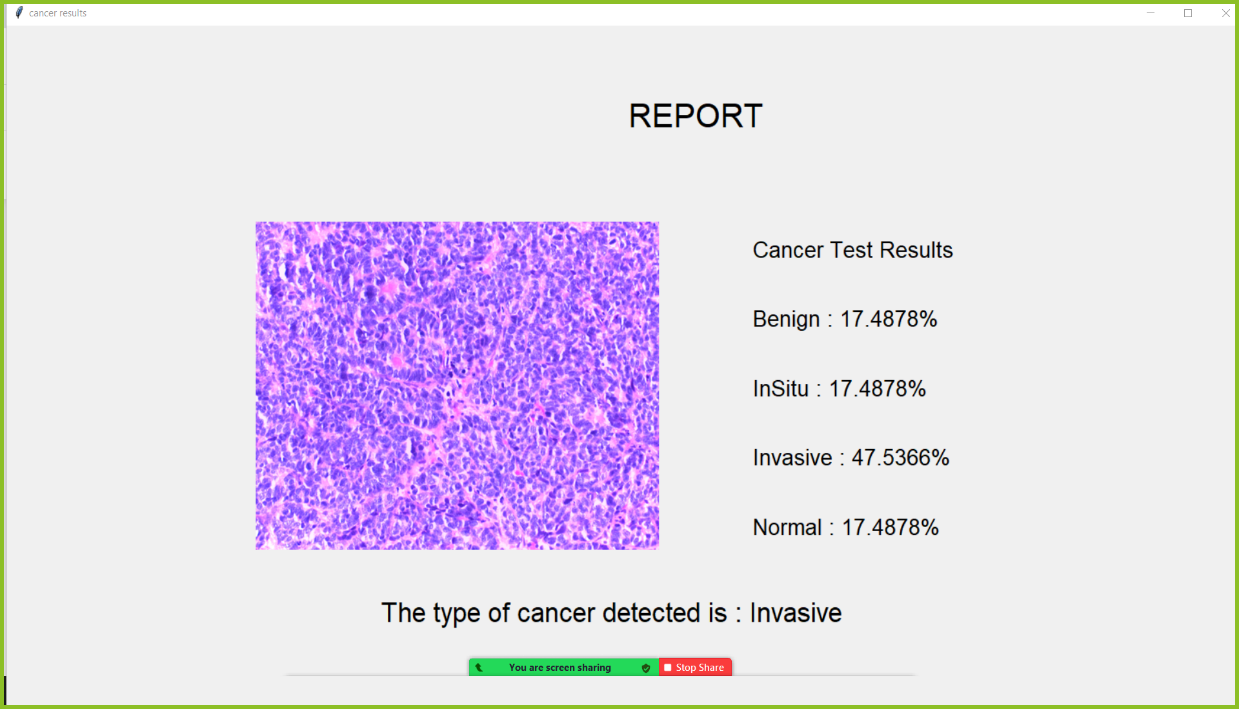


Fig 8.5 Output of Invasive type cancer

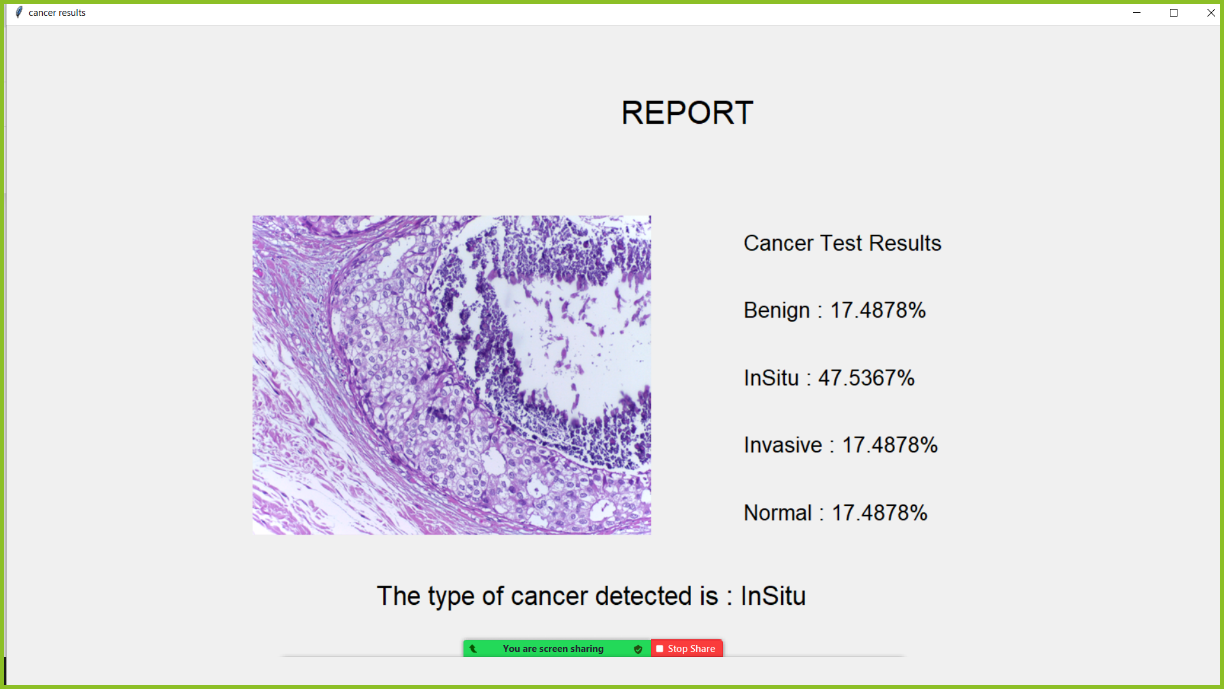


Fig 8.6 : Output of Insitu type cancer

Here the output is displayed of the type of cancer does the patient is suffering from. The percentage of the specific type of cancer is also displayed in the resultant output.

**CHAPTER 9**

**CONCLUSION AND FUTURE SCOPE**

**9.1 Conclusion**

In this project, CNN architecture that take high dimension cancer scanned image inputs and perform cancer type prediction while considering their tissue of origin. Our model achieved an equivalent 95.6% prediction accuracy -comparing to earlier published studies, however with a drastically simplified CNN construction and with a significant reduction from tissue of origin. This allows us to perform a normal interpretation of our CNN model to elucidate cancer markers for each cancer type, with hope in future refinement that will lead to markers for earlier cancer detection.

Convolutional Neural Net is a popular deep learning technique for current visual recognition tasks. Like all deep learning techniques, CNN is very dependent on the size and quality of the training data. Given a well prepared dataset, CNNs are capable of surpassing humans at visual recognition tasks. However, they are still not robust to visual artifacts such as glare and noise, which humans are able to cope. The theory of CNN is still being developed and researchers are working to endow it with properties such as active attention and online memory, allowing CNNs to evaluate new items that are vastly different from what they were trained on. This better emulates the mammalian visual system, thus moving towards a smarter artificial visual recognition system.

**9.2 Future scope**

Convolutional neural networks (CNN) has a big scope in future and the biggest reason for it is that DL doesn’t require any kind of feature engineering. Deep Learning extracts the features from the data itself instead of us giving it the features after extracting it from the data. This way it solves our biggest problem of feature engineering. Also, since features are learned by the model itself, it has a better probability of producing a model which is more generalized than the feature engineered models. These reasons alone are sufficient to prefer DL over other technologies. With the recent development in DL, now we have more state-of-the-art results on various tasks including Natural Language Processing, Language Translation, Automatic Speech Recognition, Multi Label Image Recognition/Classification.

**REFERENCES**

1. Siegel RL, Miller KD, Jemal A: *Cancer statistics*, 2018. CA Cancer J Clin 2018, 68(1):7-30.

2. Cohen JD, Li L, Wang Y, Thoburn C, Afsari B, Danilova L, Douville C, Javed AA, Wong F, Mattox A et al: *Detection and localization of surgically resectable cancers with a multi-analyte blood test*. Science 2018, 359(6378):926-930.

3. Haendel MA, Chute CG, Robinson PN: *Classification, Ontology, and Precision Medicine*. N Engl J Med 2018, 379(15):1452-1462.

4. Phallen J, Sausen M, Adleff V, Leal A, Hruban C, White J, Anagnostou V, Fiksel J, Cristiano S, Papp E et al: *Direct detection of early-stage cancers using circulating tumor DNA*. Sci Transl Med 2017, 9(403):eaan2415.

5. Schiffman JD, Fisher PG, and Gibbs P: *Early detection of cancer: past, present, and future*. Am Soc Clin Oncol Educ Book 2015:57-65.

6. LeCun Y, Bengio Y, Hinton G: *Deep learning. Nature* 2015, 521(7553):436-444.

7. Grossman RL, Heath AP, Ferretti V, Varmus HE, Lowy DR, Kibbe WA, Staudt LM: *Toward a Shared Vision for Cancer Genomic Data*. N Engl J Med 2016, 375(12):1109-1112.

8. Ahn T, Goo T, Lee C-h, Kim S, Han K, Park S, Park T: *Deep Learning-based Identification of Cancer or Normal Tissue using Gene Expression Data*. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM): 2018: IEEE; 2018: 1748-1752.

9. Lyu B, Haque A: *Deep learning based tumor type classification using gene expression data. In*: Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics: 2018: ACM; 2018: 89-96.

10. Li Y, Kang K, Krahn JM, Croutwater N, Lee K, Umbach DM, Li L: *A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data.* BMC Genomics 2017, 18(1):508.