

AI POWERED CANCER DETECTION

A Project Report Phase - II

Submitted in partial fulfilment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

By

SHOBAN PRAKASH S - (39130427)

HARI MANISH K - (39130161)



**DEPARTMENT OF ELECTRONIC AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICALS AND ELECTRONICS**

SATHYABAMA

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

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL-2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC | Approved by AICTE
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of SHOBAN PRAKASH S-(39130427) AND HARI MANISH K-(39130161) who carried out the project entitled "**AI POWERED CANCER DETECTION**" under our supervision June 2022 to April 2023.

INTERNAL GUIDE

Dr.A. ARANGANATHAN M.Tech.,PhD.

HEAD OF THE DEPARTMENT

Dr. T. Ravi M.E., Ph.D.

Submitted for Viva Voce Examination held on 26.04.23

Internal Examiner

External Examiner

DECLARATION

SHOBAN PRAKASH S-(39130427) AND HARI MANISH K-(39130161) hereby declare that the Project Report entitled “**AI POWERED CANCER DETECTION**” done by us under the guidance of **Dr.A. ARANGANATHAN M.Tech.,Ph.D.**, is submitted in partial fulfilment of the requirements for the award of **Bachelor of Engineering degree in Electronics and Communication**.

DATE: 26.04.23

PLACE: chennai

SIGNATURE OF THE CANDIDATES

1. *prakash*

2. *hari*

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N.M. NANDHITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept of Electronics and Communication Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.A. ARANGANATHAN, M.Tech.,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Cancer is the leading cause of death in the world. In 2019, annual cancer cases grew to 23.6 million people and 10 million deaths worldwide. The most common types of cancer are lung cancer, breast cancer, prostate cancer, cervical cancer, colorectal cancer, and stomach cancer. Early detection of cancer plays a vital role in survival of the patient. Imaging tests are used most for identifying the cancer. Imaging tests used in diagnosing cancer may include a computerised tomography (CT) scan, bone scan, magnetic resonance imaging (MRI), positron emission tomography (PET) scan, ultrasound and X-ray.

AI medical diagnosis is becoming effective day by day. By this approach patients can get their cancer scan reports faster. In this paper we are introducing AI assisted web applications which can predict cancer. With help of CNN (Convolutional neural networks) deep learning model. In this project we are going to detect some types of cancer. Apart from deep learning a user friendly interface using react and API using Flask (a python backend framework) is used. The web app is built to add or update existing models for future scenarios with ease. This project is open sourced to help those who are in need.

TABLE OF CONTENT

CHAPTER NO:	TITLE	PAGE NO:
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	10
	1.1 DEEP LEARNING	2
	1.2 CNN	3
	1.3 CNN ARCHITECTURE	4
	1.3.1 Convolution Layer (Kernel):	5
	1.3.2 Pooling layer	8
	1.3.3 Types of Pooling	9
	1.3.4 Classification: Fully Connected Layer (FC Layer)	10
	1.4 WEBSITE DEVELOPMENT	11
	1.4.1 ReactJS	12
	1.4.2 Firebase	14
	1.4.3 Tailwind CSS	16
	1.4.4 NodeJS	16
	1.4.5 ExpressJS	18
	1.4.6 ViteJS	19
	1.4.7 Firebase functions	19
	1.4.8 Firebase storage	20
	1.4.9 Firebase hosting	21
2	LITERATURE SURVEY AND SUMMARY	22
	2.1 LITERATURE SURVEY	23
	2.2 SUMMARY OF SURVEY	29

3	METHODOLOGY	32
	3.1 EXISTING WORK	32
	3.2 CONCLUSION OF EXISTING WORK	34
	3.3 DATA COLLECTION	36
4	IMPLEMENTATION DETAILS	39
	4.1 MODEL TRAINING	39
	4.1.1 Data Augmentation	39
	4.1.2 Building an CNN Model	40
	4.1.3 Transfer Learning	42
	4.1.4 Test the Model	43
	4.1.5 Save the Model	44
	4.2 DEVELOPMENT AND DEPLOYMENT SETUP	45
	4.2.1 Firebase function development	45
	4.2.2 React development	45
	4.2.3 Deployment	46
5	CONCLUSION AND FUTURE IMPLEMENTATIONS	49
	5.1 CONCLUSION	49
	5.2 FUTURE IMPLEMENTATION	50
6	REFERENCES	52

LIST OF FIGURES

FIGURE NO:	NAME OF THE FIGURE	PAGE NO:
1.1	CNN workflow for mnist data	4
1.2	Kernel convolution to find the feature in image	5
1.3	Binary algorithm for kernel	7
1.4	Stride Movement	7
1.5	Kernel padding	8
1.6	Difference between Max and average pooling	10
1.7	CNN Workflow	11
1.8	Firebase	14
1.9	NodeJS (starting a web server)	18
1.10	ViteJS	19
4.1	Frequency Distribution of classes	39
4.2	Data Augmentation Steps	40
4.3	Built CNN MODEL (with 5 convolution layer)	42
4.4	Training and validation loss of the Model	44
4.5	Deployment stages	48

CHAPTER 1

INTRODUCTION

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 tumour types and identification of markers for each cancer such that we can build a machine learning to identify specific metastatic tumour 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. Apart from deep learning a user friendly interface using react. The web app is built to add or update existing models for future scenarios with ease.

1.1 DEEP LEARNING

Deep learning (DL), a branch of Artificial Intelligence, is a family of multi-layer neural network models that excel at the problem of learning from big data. Similar to other machine learning methods, DL consists of the training step where the estimation of network parameters from a given training dataset is carried out, and the testing step that utilises the trained network to predict outputs of new input data. The accumulation of whole transcriptomic profiling of tumour samples enabled the pursuit of the DL model for better accuracy and innovative interpretability for cancer type prediction. One prominent resource of cancer transcriptomic profiling is The Cancer Genome Atlas (TCGA) which consists of more than 11,000 tumours from 33 most frequent cancer types. Several DL models have been developed for cancer diagnosis. Ahn, et al.,

designed a fully connected deep neural network (DNN) trained using a dataset of 6703 tumour and 6402 normal samples, and provided an initial assessment of individual gene's contribution to the final classification. Lyu et al. and Li et al. extended such an effort to classifying individual tumour types. Li et al. proposed a k-nearest neighbours (KNN) algorithm coupled with a genetic algorithm for gene selection and achieved > 90% accuracy for predicting 31 cancer types. Lyu et al. proposed a CNN model with 2D mapping of the gene expression samples as input matrices and achieved > 95% accuracy for all 33 TCGA cancer types. Lyu et al., also provided a data interpretation approach based on Guided Grad-Cam. GeneCT is another attempt which constrains the input genes to 2 categories: oncogenes and tumour suppressors (1076 genes in total) to determine the cancerous status, and transcription factors (1546 genes) to classify samples to the tissue of origin. The paper reported an overall accuracy of 97.8% with the 10-fold cross-validation. Instead of using transcriptomic data, DeepCNA, a CNN based classifier, utilised ~ 15,000 samples with copy number aberrations (CNAs) from COSMICS and the HiC data from 2 human cell-lines and achieved an accuracy ~ 60% to discern 25 cancer types. While all these attempts achieved high accuracy to some extent, these methods ignore the existence of tissue of origin within each cancer type. Without removing the influence of normal tissues during cancer classification, the implementation of a data interpretation scheme will be unlikely to differentiate tissue-specific genes or cancer-type-specific genes. Thus, it is impossible to perform functional analysis or select biomarkers for cancer detection from such models. Moreover, none of these studies systematically evaluated different CNN model constructions and their impact on the classification accuracy.

In one of our earlier attempts, Chen et al constructed an autoencoder system (GSAE) with embedded pathways and functional gene-sets at each input node to reduce the number of weights to be estimated. They applied the GSAE to classify breast cancer subtypes. Here we presented a study of different CNN models constructed for different input data formats. These models systematically interrogate the capacity of the convolution kernels. Utilising the entire collection of TCGA gene expression data sets, covering all 33 cancer types and nearly 700 normal samples from

various tissues of origin, we examined the accuracies of tumour type prediction before and after removing the influence of tissue-specific genes' expression. In addition, we proposed a unique model interpretation scheme to examine the impact of all key genes that participated in the DL prediction machinery, and we demonstrated the unique characteristics of the proposed CNN models and the feasibility of extracting diagnostic markers for future validation studies.

1.2 CNN

CNN also known as ConvNets, consist of multiple layers and are mainly used for image processing and object detection. Yann LeCun developed the first CNN in 1988 when it was called LeNet. It was used for recognizing characters like ZIP codes and digits.

CNNs are widely used to identify satellite images, process medical images, forecast time series, and detect anomalies. A neural network is a mathematical model based on neural units-artificial neurons-similar to biological neural networks. Typically, neurons are organised in layers and the connections are set up between neurons only from adjacent layers. The Input feature vector is put into the first layer and, moving from layer to layer, is transformed to a high level features vector. The output layer neurons finally leads to classification and their number is equal to the number of classifying classes. Each convolutional layer consists of a set of trainable filters and computes dot production between these filters and layer input to obtain an activation map. Filters are also called kernels and allow detecting same features in different locations. The activation function used is RELU function and Sigmoid function.

1.3 CNN ARCHITECTURE

CNN takes an image as an input, distinguishes its objects based on three colour planes, and identifies various colour spaces. It also measures the image dimensions. In order to explain this process, we will give an example of an RGB image given below.

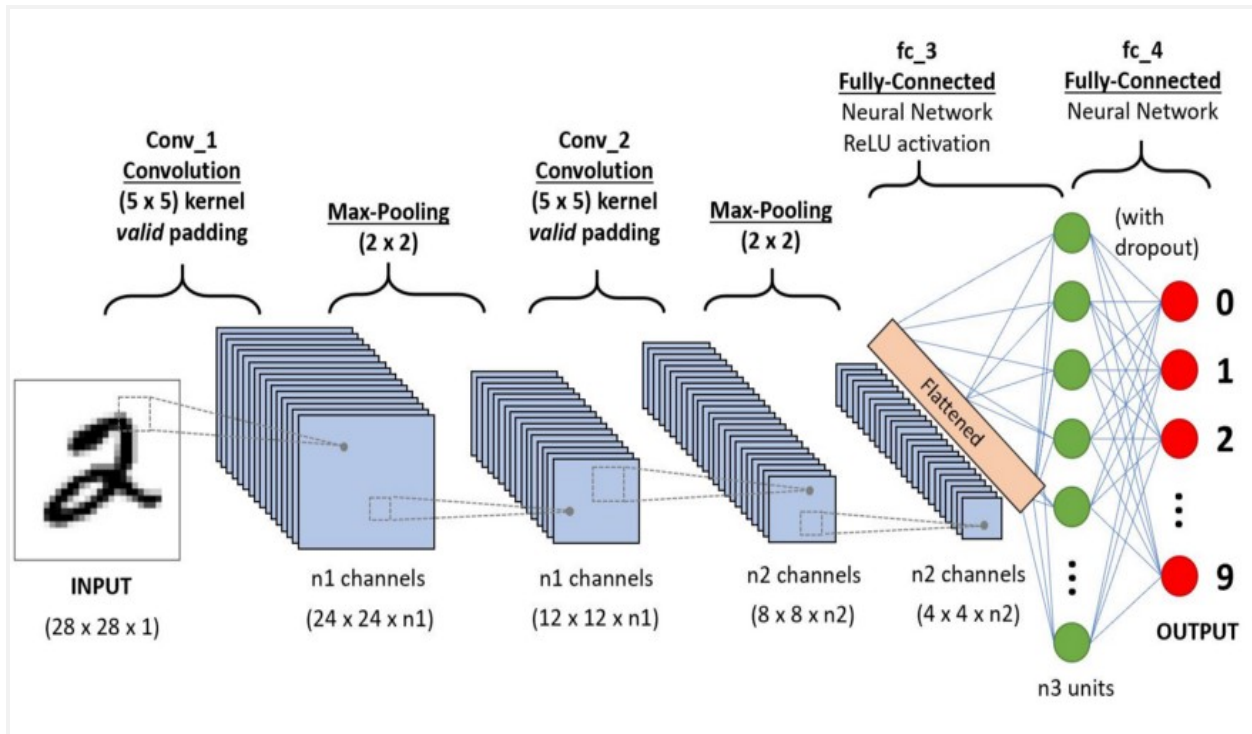


Fig 1.1 CNN workflow for mnist data

In this image, various colours are based on the three-colour plane that is Red, Green, and Blue, also known as RGB. The various colour spaces are then identified in which images are found, such as RGB, CMYK, Grayscale, and many more. It can become a tedious task while measuring the image dimensions as an example if the image is perse 8k (*7680x4320*). Here comes one of the handy capabilities of CNN: it reduces the image's dimension to the point that it is easier to process, while also maintaining all of its features in one piece. This is done so that a better prediction is obtained. This ability is critical when designing architectures having not only better learning features but also can work on massive datasets of images.

1.3.1 Convolution Layer (Kernel):

The Kernel of CNN works on the basis of the following formula.

$$\text{Image Dimensions} = n1 \times n2 \times 1$$

where $n1$ = height, $n2$ = breadth, and 1 = Number of channels such as RGB.

So, as an example, the formula will become $I D = 5 \times 5 \times 1$. We will explain this using the image given below.

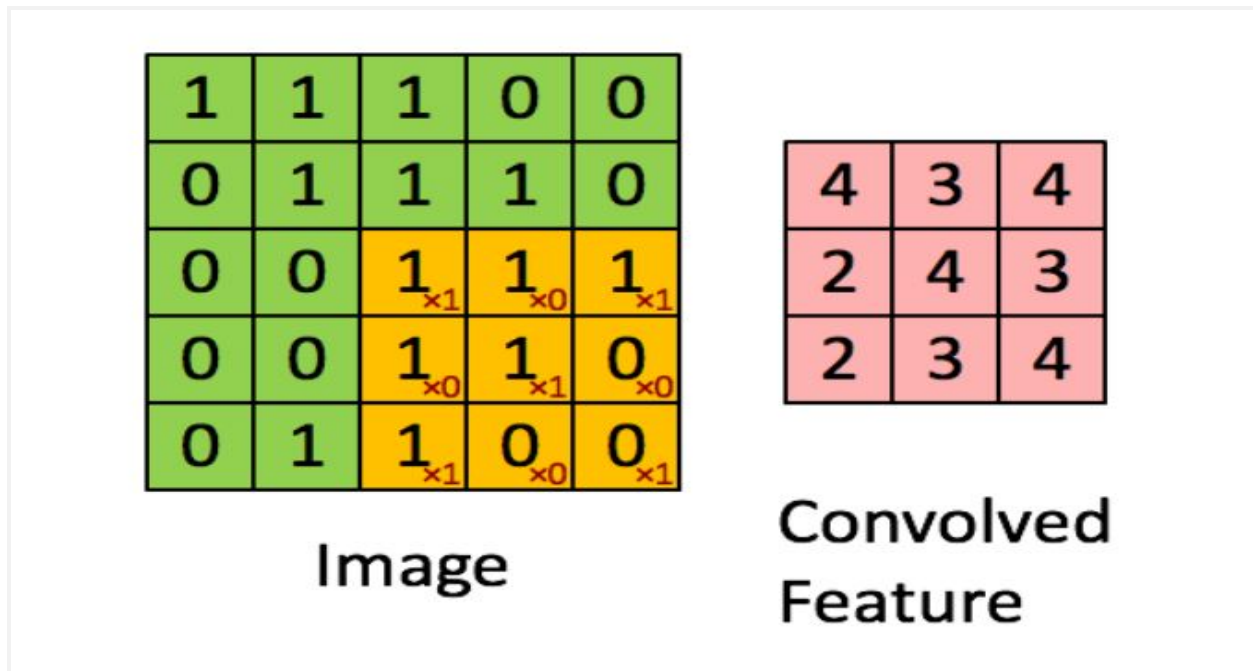


Fig 1.2 Kernel convolution to find the feature in image

In this image, the green section shows the $5 \times 5 \times 1$ formula. The yellow box evolves from the first box till last, performing the convolutional operation on every 3×3 matrix. This operation is called Kernel (K) and works on the basis of the following binary algorithm.

K =		
1	0	1
0	1	0
1	0	1

Fig 1.3 Binary algorithm for kernel

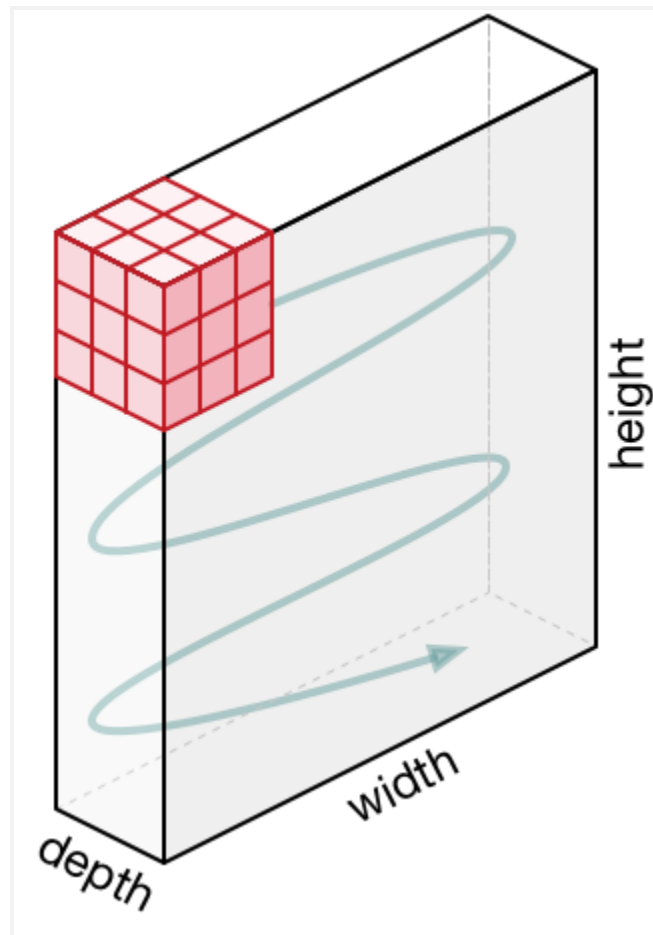


Fig 1.4 Stride Movement

In the above figure, the Kernel moves to the right with a defined value for “Stride.” Along the way, it parses the image objects until it completes the breadth. Then it hops down to the second row on the left and moves just as in the top row till it covers the whole image. The process keeps repeating until every part of the image is parsed.

If there are multiple channels such as found in RGB images, then the kernel contains the same depth as found in the input image. The multiplication of the matrix is implemented based on the number of Ks. The procedure is followed in stack format, for example, {K1, I1}, {K2, I2}, and so on. The results are generated based on the summation of bias. The result is in the form of a squeezed “1-depth channel” of convoluted feature output.

The goal of this convolution operation is to obtain all the high-level features of the image. The high-level features can include edges of the image too. This layer is not just limited to high-level features; it also performs an operation on low-level features, such as colour and gradient orientation. This architecture evolves to a new level and thus includes two more types of layers. The two layers are known as Valid padding and the Same padding.

The objective of these layers is to reduce the dimensionality of the image that is found in the original input image and to increase dimensionality or, in some cases, to leave it unchanged, depending on the required output. The same padding is applied to convolute the image to different dimensions of the matrix, while valid padding is applied when there is no need to change the dimension of the matrix.

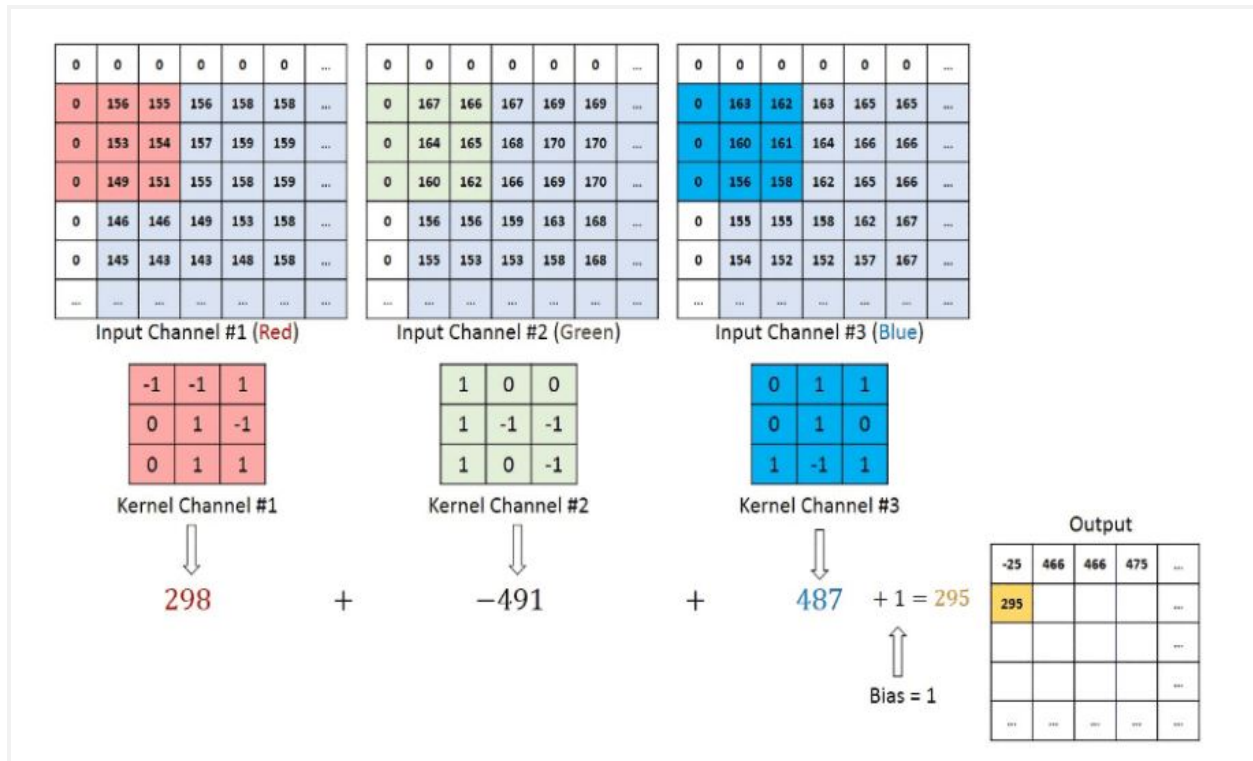


Fig 1.5 Kernel padding

1.3.2 Pooling layer

As identical to the recognized layer “convolutional,” the foremost aim of the Pooling layer is essential to decrease the spatial size of the Convolved Feature. So, in short words, it works for decreasing the required computational power for the processing of data by the method of dimensionality reduction. Moreover, it is also beneficial for the extraction of the dominant features, which are basically rotational as well as positional invariant, so the maintenance of the process effectively is needed.

1.3.3 Types of Pooling

There are mainly two different types of Pooling which are as follows:

Max Pooling: The Max Pooling basically provides the maximum value within the covered image by the Kernel.

Average Pooling: The Average Pooling provides and returns the average value within the covered image by the Kernel.

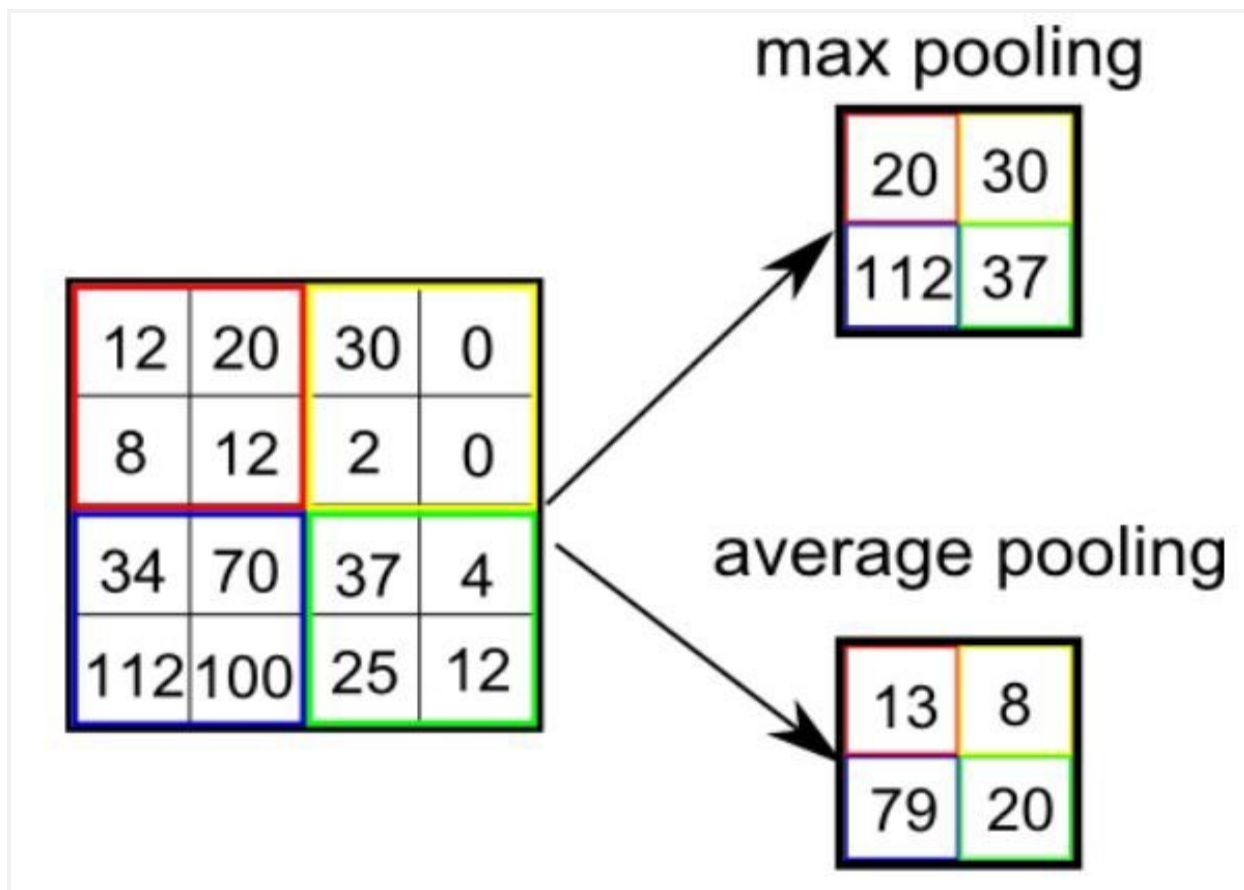


Fig 1.6 Difference between Max and average pooling

The other functionality of Max Pooling is also noise-suppressing, as it works on discarding those activations which contain noisy activation. And on the other side, the Average Pooling simply works on the mechanism of noise-suppressing by dimensionality reduction. So, in short words, we can conclude that Max Pooling works more efficiently than Average Pooling.

The Convolutional Layer, altogether with the Pooling layer, makes the “i-th layer” of the Convolutional Neural Network. Entirely reliant on the image intricacies, the layer counts might be a rise-up for the objective of capturing the details of the detailed level, but also needs to have more computational power. After analysing the above-described information about the process, we can easily execute the model for understanding the features. Moreover, here we are about to get the output and then provide it as an input for the regular Neural Network for further classification reasons.

1.3.4 Classification Fully Connected Layer (FC Layer)

The addition of the FC layer is mostly the easiest way for the learning purpose of the non-linear combinations of the abstract level structures, as it is also revealed by the output of the convolutional layer. The FC layer provides the space for learning non-linear functions. As we have now achieved our task to convert our image output into a specific form of Multi-layer Perceptron, now we must flatten the output image into a form of a column vector. Over the different eras of epochs, the model is basically succeeded for the distinguishing function between the dominating and low-level features.

Here are some impressive examples of CNN architectures:

- AlexNet
- GoogLeNet
- ZFNet
- LeNet
- ResNet

CNN's have multiple layers that process and extract features from data:

Convolution Layer

- CNN has a convolution layer that has several filters to perform the convolution operation.

Rectified Linear Unit (ReLU)

- CNN's have a ReLU layer to perform operations on elements. The output is a rectified feature map.

Pooling Layer

- The rectified feature map next feeds into a pooling layer. Pooling is a down-sampling operation that reduces the dimensions of the feature map.
- The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into a single, long, continuous, linear vector by flattening it.

Fully Connected Layer

- A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the images.

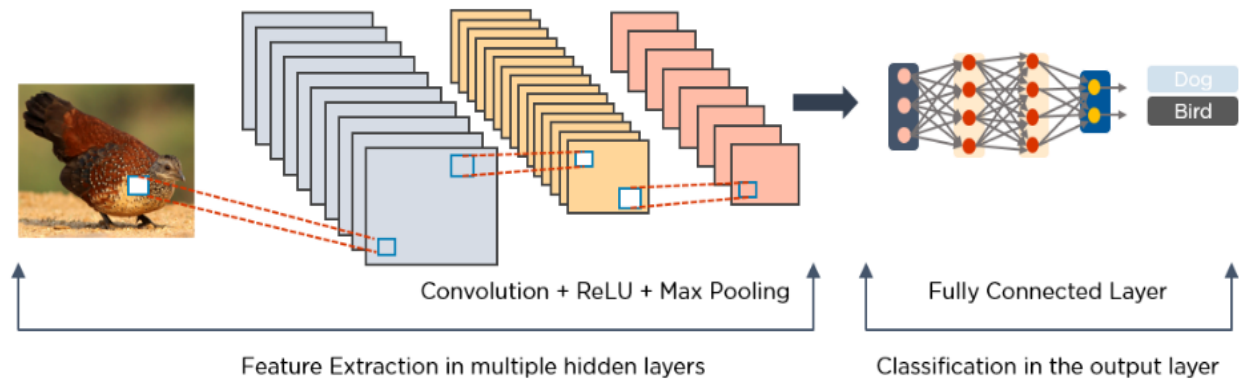


Fig 1.7 CNN Workflow

1.4 WEBSITE DEVELOPMENT

Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

It can be classified into:

- Frontend development
- Backend development
- Deployment

Frontend can be created with HTML, CSS and Javascript but it is time consuming and tedious as it requires creation of all the required modules and libraries ourselves. So a frontend framework or library can be used to create websites faster with less bugs.

1.4.1 ReactJS

React is a popular JavaScript library for building user interfaces. Developed by Facebook, React is a powerful and efficient tool for creating fast, interactive, and scalable web applications. React is based on the concept of components, which are small, self-contained units of code that can be reused throughout an application.

React allows developers to create complex user interfaces by breaking them down into smaller, more manageable components. These components can be reused across different parts of an application, making it easier to maintain and update the codebase. React components are written in JavaScript and can be used in conjunction with other libraries and frameworks to build complete applications.

One of the key features of React is its virtual DOM (Document Object Model). The virtual DOM is a lightweight representation of the actual DOM that React uses to update the user interface. When a user interacts with a React application, React updates the virtual DOM instead of the actual DOM. React then compares the virtual DOM with the actual DOM to determine what needs to be changed, and updates the actual DOM accordingly. This process is much faster than directly manipulating the actual DOM, making React applications more efficient and faster than traditional web applications.

Another key feature of React is its focus on declarative programming. With React, developers can define what they want the application to look like, and React takes care of the details of how it should be implemented. This approach makes it easier to reason about the code and to create reusable components.

React also includes a powerful set of tools and libraries that make it easy to build complex applications. These tools include React Router for client-side routing, Redux for state management, and React Native for building native mobile applications. React also integrates with other popular libraries and frameworks, such as Angular and Vue, making it a flexible tool for building web applications.

React is also known for its strong community and ecosystem. There are many resources available online for learning React, including documentation, tutorials, and forums. The React community is also constantly creating new libraries and tools to extend the functionality of React and make it easier to build complex applications.

In conclusion, React is a powerful and popular JavaScript library for building user interfaces. Its focus on component-based architecture, virtual DOM, and declarative

programming makes it efficient, scalable, and easy to maintain. With its strong community and ecosystem, React is an excellent choice for building fast and interactive web applications.

1.4.2 Firebase

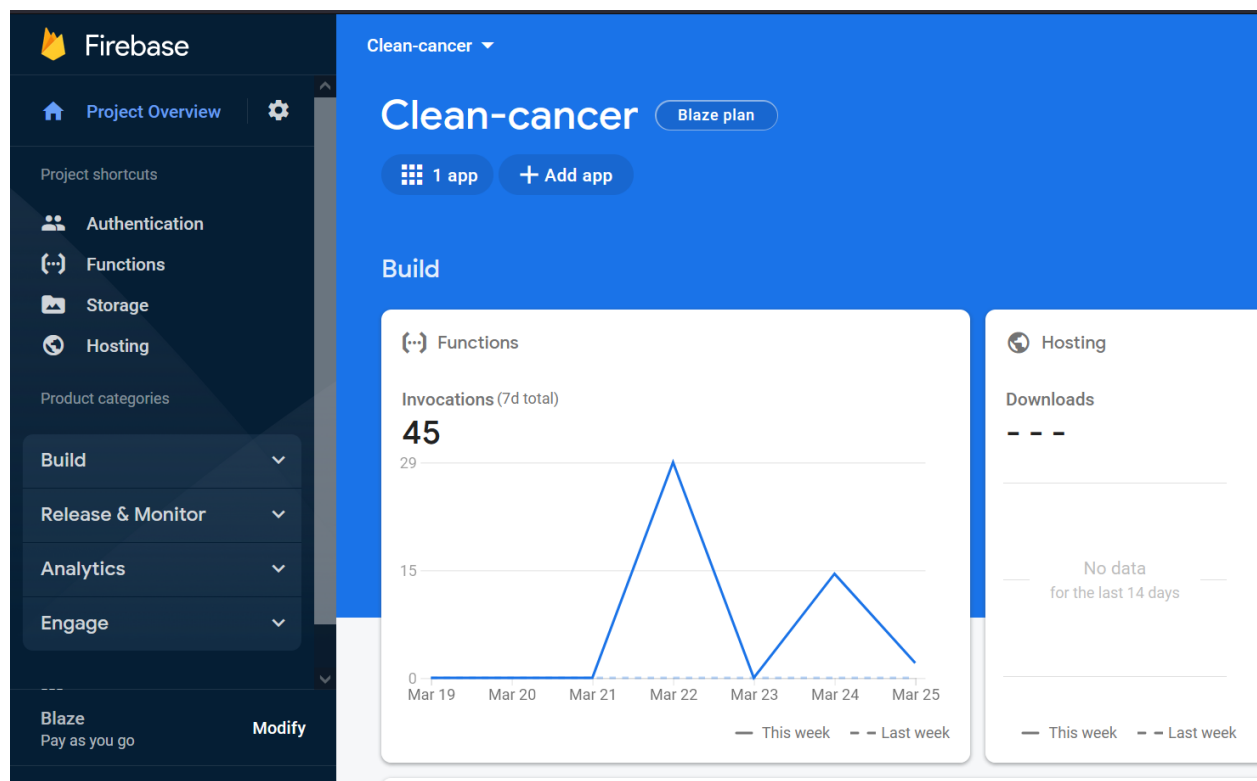


Fig 1.8 Firebase

Firebase is a comprehensive mobile and web application development platform that provides a wide range of tools and services for building high-quality and feature-rich applications. Developed by Google, Firebase provides a suite of backend services that can be integrated into any application, allowing developers to focus on creating great user experiences without worrying about the infrastructure behind the scenes.

Firebase offers a variety of features that can help developers build better applications, including real-time database, authentication, cloud messaging, storage, hosting, and more. These features can be used individually or together to build a robust and scalable application.

One of the key features of Firebase is its real-time database. This database is a NoSQL cloud-hosted solution that allows developers to store and sync data in real-time. This means that any changes made to the database are immediately propagated to all connected clients, ensuring that all users have access to the most up-to-date information. This real-time feature makes Firebase an excellent choice for building real-time applications such as chat apps, collaboration tools, and multiplayer games.

Firebase also includes a comprehensive authentication system that supports various sign-in methods, such as email and password, social media accounts, and phone numbers. This authentication system enables developers to add user authentication to their applications quickly, securely, and with minimal effort. Firebase authentication also includes built-in support for user management, such as user account creation, password resets, and more.

Firebase's cloud messaging service is another powerful feature that enables developers to send push notifications to their users on various platforms, including Android, iOS, and web. The cloud messaging service can be used to send targeted notifications based on user behaviour or to broadcast messages to all users of an application.

Firebase also provides cloud storage that allows developers to store and retrieve user-generated content such as images, videos, and audio files. This feature provides developers with a scalable and secure way to store and serve user-generated content without the need for additional infrastructure.

Firebase also offers hosting services that enable developers to deploy and host their web applications quickly and easily. Firebase hosting supports HTTPS, custom domains, and CDN integration, providing developers with a reliable and fast hosting solution for their applications.

In addition to these features, Firebase also includes a wide range of tools and SDKs that make it easy for developers to integrate Firebase services into their applications. Firebase supports multiple programming languages, including Java,

JavaScript, Swift, and Kotlin, and provides documentation and support for developers to get started quickly.

In conclusion, Firebase is an excellent platform for building mobile and web applications that require real-time data synchronisation, user authentication, push notifications, cloud storage, and hosting services. Its comprehensive set of features and easy-to-use tools make it a popular choice among developers who want to build robust, scalable, and feature-rich applications quickly and efficiently.

1.4.3 Tailwind CSS

Tailwind CSS is a popular utility-first CSS framework for building responsive and customizable user interfaces. It provides a comprehensive set of pre-designed CSS classes that developers can use to style HTML elements quickly and easily. Tailwind is designed to be highly customizable, with an emphasis on utility classes rather than pre-designed components, giving developers complete control over the look and feel of their applications.

One of the key benefits of Tailwind is its simplicity and flexibility. With its comprehensive set of utility classes, Tailwind makes it easy to create complex layouts and responsive designs without writing custom CSS code. Tailwind also integrates with popular front-end frameworks such as React, Vue, and Angular, making it easy to use in a wide variety of projects.

Overall, Tailwind is an excellent choice for developers who want to quickly create responsive and customizable user interfaces without sacrificing flexibility or control.

1.4.4 NodeJS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment built on the V8 engine. It allows developers to use JavaScript to build server-side applications that can be run on a variety of platforms, including Windows, macOS, and Linux.

One of the key benefits of Node.js is its performance. Node.js is designed to be fast and efficient, with a non-blocking, event-driven architecture that allows it to handle large numbers of simultaneous connections. This makes it an ideal choice for building real-time applications such as chat applications, multiplayer games, and financial applications.

Another benefit of Node.js is its scalability. Node.js applications can be scaled horizontally across multiple machines or vertically by adding more resources to a single machine. This makes it possible to build applications that can handle large amounts of traffic without slowing down or crashing.

Node.js also has a large and active community, with a wide range of libraries and modules available for developers to use. These libraries and modules can be easily installed using the Node Package Manager (NPM), which is built into Node.js. This makes it easy for developers to add functionality to their applications without having to write all the code from scratch.

In addition to being a great choice for building back-end applications, Node.js can also be used for front-end development. It can be used to build server-side rendered applications, which can improve the performance of web applications by reducing the amount of client-side JavaScript that needs to be downloaded and executed.

Node.js also includes several built-in modules that make it easy to work with common web technologies such as HTTP, WebSocket, and TCP. These modules can be used to build web servers, web applications, and APIs.

Node.js is used by many large companies, including Netflix, PayPal, LinkedIn, and Walmart, to power their back-end systems. It is also popular among startups and individual developers due to its ease of use, performance, and scalability.

In conclusion, Node.js is a powerful and versatile back-end JavaScript runtime environment that is used by many companies and developers to build fast, scalable, and efficient web applications. Its non-blocking, event-driven architecture, scalability, and large community make it an excellent choice for building real-time and high-traffic applications. Additionally, its built-in modules and NPM make it easy to work with common web technologies and add functionality to applications.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Fig 1.9 NodeJS (starting a web server)

1.4.5 ExpressJS

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

While Node.js is used to create a backend runtime environment and serves the server, request express is used to create API faster with the included set of tools .

This is a built-in middleware function in Express. It parses incoming requests with JSON payloads and is based on body-parser.

1.4.6 ViteJS

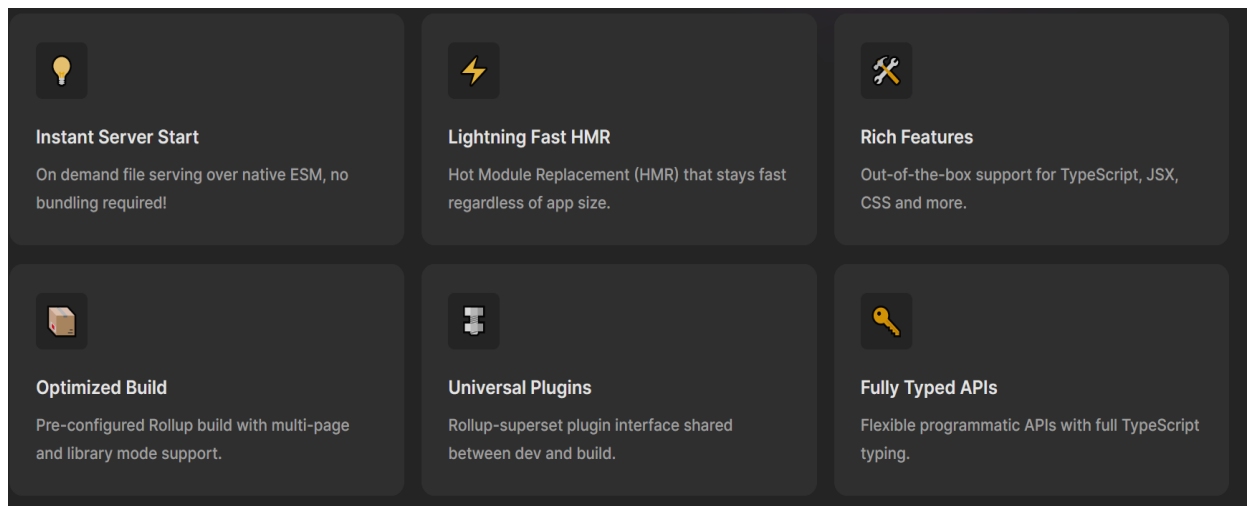


Fig 1.10 ViteJS

Vite is a build tool that aims to provide a faster and leaner development experience for modern web projects. It consists of two major parts:

- A dev server that provides rich feature enhancements over native ES modules, for example extremely fast Hot Module Replacement (HMR).
- A build command that bundles your code with Rollup, pre-configured to output highly optimized static assets for production.

Vite is opinionated and comes with sensible defaults out of the box, but is also highly extensible via its Plugin API and JavaScript API with full typing support.

1.4.7 Firebase functions

Firebase Functions is a serverless computing platform provided by Firebase, a mobile and web application development platform owned by Google. It allows developers to write server-side logic in JavaScript or TypeScript that can be executed in response to events triggered by Firebase services, such as changes to a database or the upload of a file to storage.

One of the key benefits of Firebase Functions is its ease of use. Developers can write functions using familiar languages and tools, such as Node.js and npm, without having to manage server infrastructure or worry about scalability. Firebase Functions automatically scales the infrastructure to handle the workload and charges based on usage, making it cost-effective for small and large applications alike.

Firebase Functions can be used to perform a wide range of tasks, such as sending emails, processing payments, and running machine learning models. Developers can also use it to integrate with third-party APIs, such as Stripe or Twilio, to extend the functionality of their applications.

Firebase Functions can be deployed with a single command using the Firebase CLI, and can be tested locally using the Firebase emulator suite. This makes it easy to develop, test, and deploy functions without having to set up and manage a development environment.

Firebase Functions also integrates with other Firebase services, such as Authentication, Firestore, and Cloud Storage, making it easy to build complete serverless applications with Firebase. Functions can be triggered by events from these services, and can interact with them using Firebase SDKs, allowing developers to build complex workflows without having to manage the infrastructure.

In summary, Firebase Functions is a powerful and flexible serverless computing platform that allows developers to write server-side logic in JavaScript or TypeScript that can be executed in response to events triggered by Firebase services. With its ease of

use, scalability, and integration with other Firebase services, Firebase Functions is an excellent choice for building serverless applications with Firebase.

1.4.8 Firebase storage

Firebase Storage is a cloud-based storage solution provided by Firebase, a mobile and web application development platform owned by Google. It allows developers to store and serve user-generated content such as images, videos, and audio files in a scalable and secure manner.

Firebase Storage provides a simple and intuitive API for uploading and downloading files, as well as managing file metadata and access controls. It integrates seamlessly with other Firebase services such as Authentication, Firestore, and Functions, allowing developers to build complete applications with Firebase.

Firebase Storage also provides a range of features such as resumable uploads, automatic transcoding of images and videos, and client-side SDKs for easy integration with mobile and web applications. It is designed to scale to handle large amounts of data and traffic, making it a reliable choice for applications of all sizes.

Overall, Firebase Storage is a powerful and flexible cloud storage solution that allows developers to easily store and serve user-generated content in a scalable and secure manner.

1.4.9 Firebase hosting

Firebase Hosting is a cloud-based hosting solution provided by Firebase, a mobile and web application development platform owned by Google. It allows developers to host static websites and web applications with ease, and provides a simple and intuitive interface for managing deployments, custom domains, and SSL certificates.

Firebase Hosting also provides a range of features such as CDN integration, automatic caching, and easy integration with other Firebase services such as Authentication, Firestore, and Functions. It is designed to scale to handle large amounts of traffic, making it a reliable choice for applications of all sizes.

Firebase Hosting also provides a fast and secure hosting solution, with automatic SSL certificate provisioning and renewal, and DDoS protection included. It is also easy to set up, with simple CLI tools and a web-based interface for managing deployments.

Overall, Firebase Hosting is a powerful and flexible cloud hosting solution that allows developers to easily host and scale static websites and web applications, with a range of features and integrations designed to make development and deployment as easy as possible.

CHAPTER 2

LITERATURE SURVEY AND SUMMARY

2.1 LITERATURE SURVEY

Wadood Abdul et al(2020).conducted lung cancer detection using CNN with data contained in the LIDC-IDRI database in which 1405 three-dimensional nodules were collected (1011 benign and 394 malignant). By doing data preprocessing the data was shrunk about 7000 scan reports (3500 of each, malignant and benign). And created a CNN architecture, consisting of a first convolutional layer with 32 filters and ReLU activation. The second convolutional layer consists of 16 filters followed by ReLU activation. Then the pooling layer with kernel size of 2 is used. Then there is a dropout layer. This is followed by an FC 16-neuron layer, a ReLU activation and a dropout layer. Dropout layer is used to prevent the overfitting problem. and obtained 97.2% accuracy.

Surbhi Gupta,Anish Gupta and Yogesh Kumar et al(2021). This team did bagging and boosting to an already available CNN model which predicts cancer.Bagging, also known as Bootstrap aggregating, is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model. Bagging avoids overfitting of data and is used for both regression and classification models, specifically for decision tree algorithms.This strategy however fails to discover complex information from different classification models.Traditional methods like Kaplan-Meier technique is used for analysing the effect of several risk factors on survival time. Another retrospective study proposed gene expression data can be better investigated through Cox Regression (Proportional Hazards Model) for survival analysis. Ensemble Techniques should be used for classification as these take advantage of multiple classifiers, also ensemble based on Stacking And did analysis on affected people due to cancer and created charts such as heatmap and pie charts.

Priya Bhardwaj, Yogesh Kumar and Gaurav Bhandari et al(2021). did analysis of cancer affected person details and used some graphs to show the result of the analysis.

- The most prevalent in 2020 among male as given in figure is prostate cancer and lung cancer.
- while the most prevalent cancer among females is breast cancer and lung cancer.

These are some results of the analysis. Data analysis is as important as creating a predictive model. Data analysis gives a broader view of the data. It is also an essential part of data preprocessing. Using Pandas, Matplotlib and Seaborn they created a heatmap and pie chart of the data.

Narayana Darapaneni, Bhawna Sahni, Anwesh Reddy Paduri, Sachin Jain, Syed Mohamed, Abin Banerjee and Santhanu Chakrabarti et al(2022). This team has built a business model around skin cancer detection like connecting doctors and customer care to give end to end experience for the user. They also used CNN to classify only skin cancer. They named their service model as HAIS (Health Care AI Solutions).

The step wise process flow of HAIS:

- Patient visits to the hospital.
- Patient profile is created along with past medical reports and records.
- Doctor captures the patient's skin image.
- Skin images are uploaded on HAIS IT platform.
- HAIS IT platform reads medical reports via robotic process automation module and associates these additional details as a meta-data to the images.
- Transformed images are fed into the AIML image processing module.
- AIML image processing module returns the outcome of the skin cancer prediction, along with risk levels.
- Doctors analyse the outcome to provide more accurate feedback to the patient.

The classification systems available have been evaluated using the large public skin lesion testing dataset, containing approx 11000 testing images, and 24000 training images, and has obtained performance results around 89.1%.

M. Ghaderi, G. Gupta, L. Tamil, M. Abdallah and K. Qaraqe et al(2020). In this paper they presented a mHealth platform that helps users to do automated breast cancer risk analysis and access to doctors for interpretation, support and further action. Their user interface is heavily based on AngularJS. AngularJS is a JavaScript library that simplifies development of the client-side application. Our UI (User Interface) is a combination of ASP.NET views and AngularJS single page application. Pages outside of user panels (pages before login) are like regular web pages and have to have a routing inside of the server-side part. After login, all pages are routed and handled using AngularJS routing to give the feeling of a native application to the user. Website is hosted on an Amazon EC2 (Elastic Cluster) node. It's a Windows 2013 Server machine with IIS (Internet Information Server) to execute ASP.NET applications. They have planned to create an app for mobile phones in the next phase of their project.

Syed Amer Ali, Nikhil Vallapureddy, Sridivya Mannem, Yashwanth Gudla, Dr V Malathy et al(2022). They proposed diagnosing lung cancer from CT images using SVM classification and comparing the survival rate of patients using 3D Convolutional neural network(CNN) on lung nodules data set in this study, and they discovered that the survival rate of lung cancer patients has improved. Deep learning has a significant advantage over conventional machine learning algorithms in that it can undertake feature engineering on its own. This analyses the data in order to discover and include related features into the learning process. During the training and testing phase, images are pre-processed, and feature selection and extraction are conducted. The CNN algorithm classifies the input lung picture as normal or abnormal and presents the result when the training and testing phases are successfully completed. As a result, in order to detect cancer, a Deep CNN network is used to classify lung images.

Min Lu, Xia Xiao et al(2022). This article proposes a CNN-LSTM framework based on 560 UWB backscatter signals for breast tumour detection and localization. The received microwave signals of three channels are 562 directly fed into the proposed network. The task of tumour 563 detection and localization can be realised at once, and the 564 whole process is completely automatic without empirical and 565 manual feature extraction. With F1 score and accuracy both 566 above 97%, the proposed network performs well on the two 567 simulation datasets. The testing results based on the experimental system also show superior performance and overall 569 accuracy reaches 89.50%. Tumour detection and localization 570 based on UWB microwave detection and deep learning frame 571 work is helpful for early warning and timely diagnosis of 572 breast cancer. Future work will focus on establishing a more 573 comprehensive UWB breast signal dataset and further option 574 mixing the proposed deep learning algorithm for promoting 575 this technology to practical applications.

Wadhah Ayadi, Wajdi Elhamzi and Mohamed Atri et al(2022). They present a new scheme using CNN for multi-brain tumour classification. It is automatic and a minimum preprocessing is needed. The suggested technique was tested on a public dataset. Several performance metrics were calculated with the goal to evaluate the model. The proposed approach yields a better performance compared to some previous methods, which exploit the same dataset. The experimental results illustrate the suggested technique effectiveness. The new method can be used for some other MRI classification since it does not require handcrafted features.

Soumen Basu, Mayank Gupta, Pratyaksha Rana, Pankaj Gupta and Chetan Arora et al(2022). Their paper addresses Gallbladder Cancer detection from Ultrasound images using deep learning and proposes a new supervised learning framework (GBCNet) based on ROI selection and multi-scale second-order pooling. The proposed design helps the classifier focus on the crucial GB region predicted by the region selection network. They propose a visual acuity-based curriculum to make our design resilient to texture bias and improve its specificity. Extensive experiments show that GBCNet, combined with curriculum learning, improves performance over the

baseline deep classification and object detection architectures. They hope our work will generate interest in the community towards this important but hitherto overlooked problem of GBC detection.

K.Srilatha, P.Chitra, M.Sumathi, Mary Sajin Sanju.I and F.V.Jayasudha et al(2022). The proposed method has developed a computerised process in support of the segmentation and classification of different kinds of tumours within brain MRI imagery by FCM and GLCM texture features. This system is to be effectively tested on MRI brain images for brain tumours. The brain area interconnected to a lesion is precisely broken up from the MRI brain image. The proposed method assists the general practitioner to make out with reference to the kind of brain tumours, for more dealing. The Precision, Sensitivity, Specificity and F-Score among the outcomes obtained using the proposed technique and the GT (groundtruth) are 98.65, 97.46, 96.98 and 98.01 respectively. A classification with accuracy of 92.76 percentages has been obtained by the Convolutional Neural Network (CNN) classifier. The proposed system can be considered to classify other kinds of tumour as well by a small amount of variations.

Imen Labiadh, Hassene Seddik, Larbi Boubchir et al(2022). In this paper, they have studied LBP algorithm, skewness coefficients for asymmetry detection and correction, morphological transformations, and CNN models for detecting prostate cancer tumours from microscopic cells and MRI imaging. Proposed methods based on these algorithms and techniques with DL architecture to achieve powerful image recognition. The experimental results carried out have shown the effectiveness of the proposed methods using DL architectures, in particular, AlexNet. Our future work will focus on extending this work by studying the detection of tumour surfaces in blurred images.

M. Iswarya, Sivakami A, Aravindha Mira. R and Karpagam. A et al(2022). Their idea is to provide an algorithm to diagnose leukaemia in a cost-effective method and with greater evaluation accuracy. The proposed non-invasive method detects the

occurrence of leukaemia at an early stage. Furthermore, the results can be extended for therapeutic applications. The overall accuracy of the project is calculated by finding accuracy vs epoch graph where the calculation is done for both training and testing data. The accuracy is found to be greater than 90%.

Jessnia Fernandes, Nikeeta Simoes, Dominic Vaz, Saurav Tiwari, Amrita Naik and Damodar Reddy Edla et al(2022). The proposed work selects the most relevant feature from CNN using an extra tree feature selection algorithm and classifies the selected feature using an extreme learning machine to classify the nodules of the lung into either malignant or benign. The LUNA16 dataset was utilised to evaluate our model. When we used only CNN the accuracy, precision, sensitivity and specificity were 91.11%, 82.08%, 91.79% and 89.52% respectively , whereas using the proposed model we achieved an accuracy of 93.74% precision of 86.75 % and the sensitivity and specificity are 94.03% and 93.07% respectively. Our study shows that use of feature selection algorithms on CNN features has improved the accuracy of classification systems. Extra tree feature selection algorithm has performed better than other feature selection algorithms they tried to utilise.

T.Gayathri, T.Madhavi and K.Ratna Kumari et al(2022). In their study, the learning rate of CNN is optimised by using the Mayfly algorithm. The experiments are carried out on two different datasets such as MIAS and CBIS-DDSM datasets in terms of various parameters for the analysis of segmentation and classification performance. From the results, it is proved that the proposed MFA-CNN achieved better performance for the MIAS dataset than CBIS-DDSM dataset in terms of all parameters. In future work, the presents study needs the improvement of new meta-heuristics algorithm and deep learning techniques for achieving better classification accuracy

Necip Çınar, Buket Kaya and Mehmet Kaya et al(2022). In their study, the success of the ResNet101 architecture on the test dataset was determined by accuracy, recall, precision and F1-score criteria. As a result of the training and test experiments, it has been seen that the ResNet101 model provides a higher accuracy rate than the

VGG19, InceptionV3, DenseNet169 and AlexNet. The main differences that distinguish the ResNet101 architecture from other CNNs architectures are the number and structure of the layers in its architecture. It can be said that the loss and activation functions used in architecture also have a direct effect on performance. The filters used and the size of these filters are other factors that affect the success of the model. It can also be said to be a suitable model for high-resolution images such as MR images. The accuracy value of the ResNet101 method was compared with previous similar studies and it was seen that the performance of the ResNet101 method was high. In future studies, it is aimed to modify ResNet architecture to see the impact of the layers on model's success.

Amgoth Harichandar Lal, E. Sreenivasulu, M. Aravind Kumar and Srinivas Bachu et al(2022). Their work developed a computational methodology for detecting and classifying Brain tumours from MRI data using a CNN-based deep learning strategy. Initially, Gaussian filters are used for preprocessing, which remove any undesirable noise components or artefacts introduced during brain tumour acquisition. The ROI is extracted and malignant cells are detected using K-means clustering segmentation. Then, the GLCM, DWT-based technique was created for extracting statistical, colour, and texture information from segmented images. Finally, using a trained network model, CNN was used to categorise the kind of tumour as benign or malignant. As a result, we find that CNN outperforms the traditional SVM approach when compared to state-of-the-art studies. This study may be expanded in the future by including a larger number of network layers into the CNN, and it can also be used to treat different types of benign and malignant tumours.

2.2 SUMMARY OF SURVEY

We conducted a survey of existing projects that use convolutional neural networks (CNNs) to predict cancer from medical images. We found that CNNs are the

most efficient way of achieving high accuracy and performance in this domain. However, we also noticed some limitations and gaps in the current state-of-the-art:

- Most of the projects focused on a single type of cancer, such as breast, lung, or skin cancer. This limits the generalizability and applicability of the models to other cancer types and scenarios.
- Some of the projects did not use any ensemble methods, such as boosting or bagging, to improve the robustness and stability of the models. This may lead to overfitting or underfitting problems and reduce the reliability of the predictions.
- Some of the projects did not deploy their models as user-friendly applications that can be accessed by patients or clinicians. This reduces the impact and usefulness of the models in real-world settings.
- Only one of the projects proposed a business model around their idea, but it was a business-to-business (B2B) solution that targeted hospitals and clinics as customers. This leaves out the potential market of business-to-customer (B2C) solutions that can directly benefit individual users.

Based on these findings, we identified some opportunities and goals for our project that can address these challenges and provide a novel and valuable solution for cancer prediction using CNNs.

Our goal is to develop a web-based application that can predict multiple types of cancer from medical images using CNNs. Our application will have the following features and advantages:

- User experience: We aim to provide an end-to-end user experience that is easy, intuitive, and informative. Our application will allow users to upload their own images, select the type of cancer they want to check, and receive a prediction with a confidence score and an explanation. Our application will also provide brief information about each type of cancer, such as its symptoms, causes, risk factors, and prevention methods.

- Accuracy: We plan to increase the accuracy of our model by using ensemble methods, such as boosting, bagging, and stacking. We will train our model on a large and diverse dataset of images from different sources and labels. We will also use data augmentation techniques to increase the variability and quality of the data. We aim to achieve at least 97.2% accuracy for our model, which is the highest among our surveyed projects.
- Business model: We propose a business-to-customer (B2C) solution that can offer our service for free to individual users. By doing so, we can reach a wider audience and create social value by raising awareness and facilitating early detection of cancer. We can also generate revenue by collecting data from our users, such as their location and health condition, and using it to create personalised recommendations for nearby hospitals and health centres that offer diagnosis and treatment services for cancer.
- Light and fast: We will use client-side rendering and running models to reduce the load on the server and improve the speed and responsiveness of our application. We will also use viteJs to bundle our source code and make it load faster in the browser.
- More types of cancer: We will not limit our model to a single type of cancer, but rather include as many types as possible. We will use a multi-class classification approach that can handle multiple labels for each image. We will also use transfer learning techniques to leverage pre-trained models that have learned features from other domains and tasks.
- Open source: Our project will be an open source project that anyone can access, use, or contribute to. We will publish our code, data, and documentation on GitHub and other platforms. We will also welcome feedback and suggestions from other developers and researchers who are interested in this field.

CHAPTER 3

METHODOLOGY

3.1 EXISTING WORK

Cancer is a deadly disease that affects millions of people worldwide. Early detection of cancer is crucial in increasing the chances of successful treatment and improving patient outcomes. In recent years, there has been a growing interest in using machine learning techniques, particularly Convolutional Neural Networks (CNNs), for cancer detection.

CNNs are a type of deep learning algorithm that can learn features directly from images, making them particularly suitable for image-based cancer detection. These algorithms work by analysing the pixels in an image and identifying patterns that are associated with the presence of cancer. CNNs have shown remarkable performance in a variety of image classification tasks, including cancer detection, and have outperformed traditional machine learning methods.

The current state of the art in cancer detection using CNNs is still in the proposed stage, and it is not yet available for public use. The system is undergoing further development and refinement to ensure its reliability and effectiveness in real-world scenarios. Although the system has shown promising results in detecting various types of cancer accurately, it requires further improvements and testing to ensure its accuracy and performance in clinical settings.

The proposed CNN-based cancer detection system works by analysing medical images such as X-rays, CT scans, and MRIs. The system uses deep learning algorithms to detect patterns in the images that are associated with the presence of cancer. Once the patterns are identified, the system uses them to classify the image as either cancerous or non-cancerous. The system can detect various types of cancer,

including lung cancer, breast cancer, and prostate cancer, with a high degree of accuracy.

One of the main advantages of using CNNs for cancer detection is their ability to learn features directly from the images without the need for manual feature engineering. This means that the system can detect patterns that are not easily visible to the naked eye, allowing for early detection of cancer. CNNs can also detect patterns that are unique to different types of cancer, allowing for accurate diagnosis and personalised treatment plans.

Despite the promise of CNN-based cancer detection systems, several challenges must be addressed before they can be deployed for public use. One of the main challenges is the need for large amounts of annotated data to train the system. Annotated data refers to images that are labelled as either cancerous or non-cancerous. The system needs access to a large dataset of annotated images to learn the patterns associated with cancer accurately.

Another challenge is the need for robust validation and testing to ensure the system's reliability and effectiveness in real-world scenarios. The system must undergo extensive testing to ensure its accuracy and performance in clinical settings. The system must also be validated on diverse datasets to ensure its generalizability across different populations.

The proposed CNN-based cancer detection system is not yet deployed as a web application, which limits its accessibility to a broader audience. Deploying the system as a web application would make it more accessible to patients and healthcare professionals, allowing for faster and more accurate diagnosis of cancer. A web-based system would also allow for remote diagnosis and monitoring, enabling patients to receive timely treatment regardless of their location.

In conclusion, the proposed CNN-based cancer detection system holds immense potential to revolutionise cancer diagnosis and improve patient outcomes. Although the

system is still in the proposed stage and is not yet available for public use, it has shown promising results in detecting various types of cancer accurately. However, several challenges must be addressed before the system can be deployed for public use, including the need for large amounts of annotated data, robust validation and testing, and deployment as a web application. With continued development and refinement, CNN-based cancer detection systems have the potential to transform cancer diagnosis and improve patient outcomes.

3.2 CONCLUSION OF EXISTING WORK

In conclusion, the existing cancer detection system based on Convolutional Neural Networks (CNNs) is currently in the proposed state and is far from being available for public use. While the system has shown promising results in detecting various types of cancer accurately, it still requires further improvements and testing to ensure its reliability and effectiveness in real-world scenarios. Moreover, the system is not yet deployed as a web application, which limits its accessibility to a broader audience.

The proposed CNN-based cancer detection system holds immense potential to revolutionise cancer diagnosis and improve patient outcomes. One of the main advantages of using CNNs for cancer detection is their ability to learn features directly from the images without the need for manual feature engineering. This means that the system can detect patterns that are not easily visible to the naked eye, allowing for early detection of cancer. CNNs can also detect patterns that are unique to different types of cancer, allowing for accurate diagnosis and personalised treatment plans.

However, there are several challenges that must be addressed before CNN-based cancer detection systems can be deployed for public use. One of the main challenges is the need for large amounts of annotated data to train the system. Annotated data refers to images that are labelled as either cancerous or non-cancerous. The system needs access to a large dataset of annotated images to learn the patterns associated with cancer accurately.

Another challenge is the need for robust validation and testing to ensure the system's reliability and effectiveness in real-world scenarios. The system must undergo extensive testing to ensure its accuracy and performance in clinical settings. The system must also be validated on diverse datasets to ensure its generalizability across different populations.

Deploying the system as a web application would make it more accessible to patients and healthcare professionals, allowing for faster and more accurate diagnosis of cancer. A web-based system would also allow for remote diagnosis and monitoring, enabling patients to receive timely treatment regardless of their location. Therefore, it is essential to focus on developing a web-based system for CNN-based cancer detection to make it more accessible and convenient for the public.

Furthermore, collaboration with healthcare professionals and medical experts is crucial to ensure that CNN-based cancer detection systems are developed with real-world scenarios in mind. Medical experts can provide valuable insights into the types of cancer and the patterns that are unique to them, which can help improve the accuracy of the system. Healthcare professionals can also provide feedback on the system's usability and effectiveness in clinical settings.

In conclusion, the proposed CNN-based cancer detection system holds immense potential to transform cancer diagnosis and improve patient outcomes. Although the system is still in the proposed stage and is not yet available for public use, continued development and refinement are needed to address the challenges and ensure its reliability and effectiveness in real-world scenarios. The deployment of a web-based system and collaboration with medical experts and healthcare professionals can further enhance the system's accuracy and accessibility, making it an invaluable tool in the fight against cancer.

3.3 DATA COLLECTION

Data is collected from the kaggle website. The name of the data is Skin Cancer MNIST: HAM10000

Glossary of Terms:

- **Nv** Melanocytic nevi are benign neoplasms of melanocytes and appear in a myriad of variants, which all are included in our series. The variants may differ significantly from a dermatoscopic point of view.
- **Mel** Melanoma is a malignant neoplasm derived from melanocytes that may appear in different variants. If excised in an early stage it can be cured by simple surgical excision. Melanomas can be invasive or non-invasive (in situ). We included all variants of melanoma including melanoma in situ, but did exclude non-pigmented, subungual, ocular or mucosal melanoma.
- **Bkl** "Benign keratosis" is a generic class that includes seborrheic keratoses ("senile wart"), solar lentigo - which can be regarded a flat variant of seborrheic keratosis - and lichen-planus like keratosis (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression. The three subgroups may look different dermatoscopically, but we grouped them together because they are similar biologically and often reported under the same generic term histopathologically. From a dermatoscopic view, lichen planus-like keratosis are especially challenging because they can show morphologic features mimicking melanoma and are often biopsied or excised for diagnostic reasons.
- **Bcc** Basal cell carcinoma is a common variant of epithelial skin cancer that rarely metastasizes but grows destructively if untreated. It appears in different morphologic variants (flat, nodular, pigmented, cystic, etc), which are all included in this set.
- **Akiec** Actinic Keratoses (Solar Keratoses) and intraepithelial Carcinoma (Bowen's disease) are common, non-invasive, variants of squamous cell carcinoma that can be treated locally without surgery. Some authors regard them as precursors of squamous cell carcinomas and not as actual carcinomas. There is, however, agreement that these lesions may progress to invasive squamous

cell carcinoma - which is usually not pigmented. Both neoplasms commonly show surface scaling and commonly are devoid of pigment. Actinic keratoses are more common on the face and Bowen's disease is more common on other body sites. Because both types are induced by UV-light the surrounding skin is usually typified by severe sun damage except in cases of Bowen's disease that are caused by human papillomavirus infection and not by UV. Pigmented variants exist for Bowen's disease and for actinic keratoses. Both are included in this set.

We will use the 'Breast Histopathology Images' dataset. This dataset consists of 277,524 image patches of size 50x50 (198,738 IDC negative and 78,786 IDC positive). The images are in png format.

Glossary of Terms:

- **Histopathology** This involves examining glass tissue slides under a microscope to see if disease is present. In this case that would be examining tissue samples from lymph nodes in order to detect breast cancer.
- **Whole Slide Image (WSI)** A digitised high resolution image of a glass slide taken with a scanner. The images can be several gigabytes in size. Therefore, to allow them to be used in machine learning these digital images are cut up into Patches.
- **Patch** A patch is a small, usually rectangular, piece of an image. For example, a 50x50 patch is a square patch containing 2500 pixels, taken from a larger image of size say 1000x1000 pixels.
- **Lymph Node** This is a small bean shaped structure that's part of the body's immune system. Lymph nodes filter substances that travel through the lymphatic fluid. They contain lymphocytes (white blood cells) that help the body fight infection and disease.

- **Sentinel Lymph Node** A blue dye and/or radioactive tracer is injected near the tumour. The first lymph node reached by this injected substance is called the sentinel lymph node. The images that we will be using are all of tissue samples taken from sentinel lymph nodes.
- **Invasive Ductal Carcinoma** This is the most common subtype of all breast cancers. Almost 80% of diagnosed breast cancers are of this subtype.
- **Metastasis** The spread of cancer cells to new areas of the body, often via the lymph system or bloodstream.
- **Metastatic Cancer** A metastatic cancer or metastatic tumour is one that has spread from the site where it started into different area/s of the body.

CHAPTER 4

IMPLEMENTATION DETAILS

4.1 MODEL TRAINING

EDA (Exploratory data analysis) is important for training of the model. It is used to get an insight about our data before the training of the data.

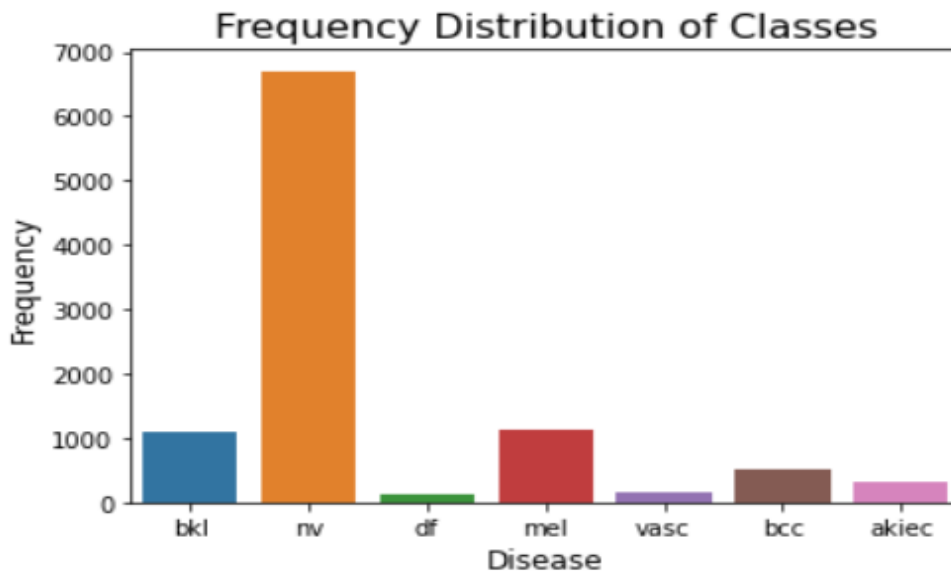


Fig 4.1 Frequency Distribution of classes

4.1.1 Data Augmentation

Contemporary advances in deep learning models have been largely associated with large quantities and diversity of data. Having large data helps crucially in improving the performance of machine learning models. But obtaining such vast quantities of data is cost-intensive and tedious. Hence, we use the technique of Data Augmentation. It is a technique that enables us to considerably increment the diversity and quantity of data available, without actually aggregating new data. In order to generate new data through

augmentation of images, various techniques such as cropping, padding, adding noise, brightness changing and horizontal flipping are commonly used to train large neural networks.

In this project, the training images are augmented in order to make the model robust to new data which in turn helps in increasing the testing accuracy.

Table 5 shows the augmentation techniques used on the dataset.

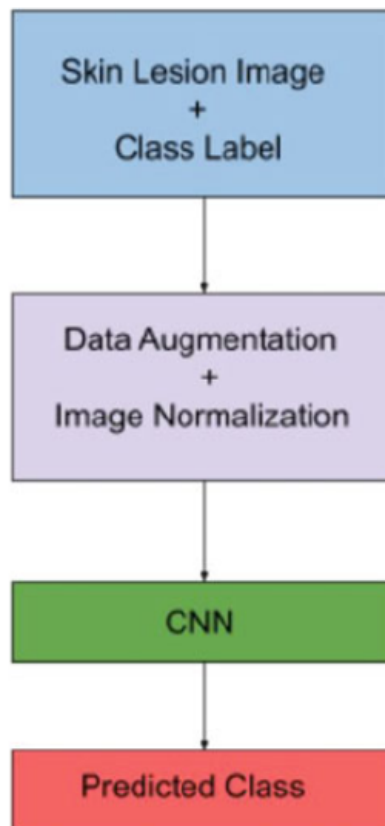


Fig 4.2 Data Augmentation Steps

4.1.2 Building an CNN Model

Now we will start developing a convolutional neural network to classify images for correct labels. CNN is best to work with image data.

The architecture of our CNN model

- Conv2D layer – we will add 5 convolutional layers of 32 filters, size of 5*5, and activation as Relu
- Max Pooling – MaxPool2D with 2*2 layers.
- Dropout with a rate of (0.25).
- Flatten layer to squeeze the layers.
- Dense layer(nodes=9, activation="softmax")
- MaxPool2D – Maximum pooling layer is used to reduce the size of images
- Dropout – Dropout is a regularisation technique to reduce overfitting
- Flatten – to convert the parallel layers to squeeze the layers
- Dense – for feed-forward neural network

The last layer will have an activation function as softmax for Multi-class classification.

```

''' Model '''
n_c = 9

''' preprocessing layer '''
model = Sequential([layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_h, img_w, 3))])

''' Convolutional Layers '''
model.add(Conv2D(filters=32, kernel_size=(5,5), padding='Same', activation='relu', input_shape = (180, 180, 3)))
model.add(Conv2D(filters=32, kernel_size=(5,5), padding = 'Same', activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

model.add(Conv2D(filters=32, kernel_size=(5,5), padding='Same', activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

model.add(Conv2D(filters=32, kernel_size=(5,5), padding='Same', activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

model.add(Conv2D(filters=32, kernel_size=(5,5), padding='Same', activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))

''' adding dropout '''
model.add(Dropout(0.25))
model.add(Flatten())

''' Classification Layer'''
model.add(Dense(n_c, activation = "softmax"))

```

Fig 4.3 Buildd CNN MODEL (with 5 convolution layer)

4.1.3 Transfer Learning

Transfer Learning is a learning method in which a model trained for a particular task is reiterated as the origin for another model on a similar task. This approach is very mainstream in deep learning due to the vast computation resources and time consumed to train neural network models. In the case of problems in the computer vision domain, low-level features, such as shapes, corners, edges and intensity, can be shared across tasks, and thus enable knowledge transfer.

In this project, the CNN is trained by transfer learning from the pretrained weight of ImageNet classification. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is the classification challenge involving 14 million training images of approximately 20,000 classes. Fine-tuning from the pretrained weight significantly increases the classifier training speed, overcomes the limit of small number of training data and makes the convergence more realistic to occur.

The front layers of the CNN serve to detect the edges, points, corners and simple structures in the image and the pretrained weight enables the trained model to have higher adaptation speed. The model is trained and tested on the state-of-art CNNs, namely Inception.

4.1.4 Test the Model

The dataset contains a test folder that has different test images and a test.csv file that contains details related to the image path and respective labels. Again we will load the data using pandas and resize it to the shape of 30*30 pixels and convert it to a NumPy array. After processing test images we will check the accuracy of the model against actual labels.

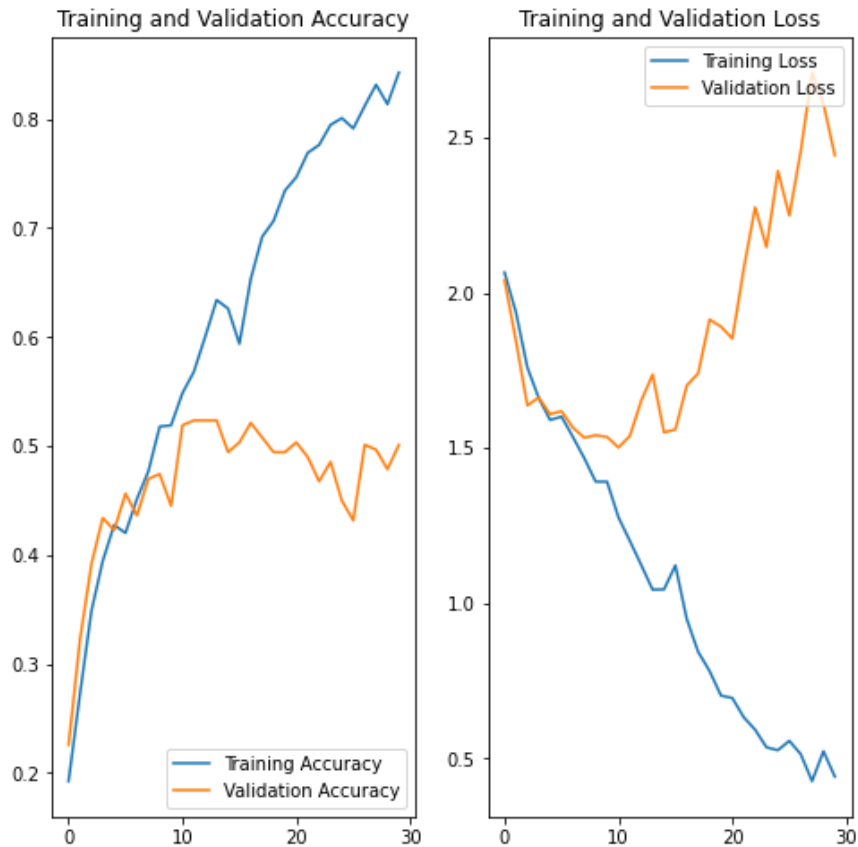


Fig 4.4 Training and validation loss of the Model

4.1.5 Save the Model

Save the model for future use as well, we will use the dump model to create a GUI for Traffic Classification Project.

```
model.save('traffic_classifier.h5')
```

Hence we have successfully built and evaluated our Convolutional neural network for the image classification task. Now we will work on the frontend part and deploy our model on GUI.

4.2 DEVELOPMENT AND DEPLOYMENT SETUP

4.2.1 Firebase function development

This function is built using Node.js application using the Express framework, which provides a server listening on a port for HTTP requests. It uses TensorFlow.js to load a pre-trained machine learning model for image classification, and Multer for handling file uploads. The application responds to POST requests sent to the /predict endpoint. The incoming image URL is retrieved and decoded, and then preprocessed by resizing, casting, and expanding its dimensions before it is fed into the loaded model. The model predicts the probabilities of each class and returns the top 5 most likely classes along with their probabilities in a JSON response. The server logs the top 5 classes to the console, and the response is sent back to the client. The CORS headers allow for cross-origin resource sharing, and Firebase Functions is used for hosting the application in a serverless environment.

4.2.2 React development

The development of this code involved several key steps, including setting up Firebase and the backend API, creating a React component, and implementing the image upload and prediction functionality.

To begin, we created a Firebase project and configured the necessary services, including Authentication and Storage. We also generated a configuration file to connect our React app to Firebase, ensuring secure and efficient communication between the frontend and backend.

Next, we created a backend API using a serverless function in Google Cloud Functions. This function leveraged a machine learning model to process the uploaded image and generate a prediction. We also set up the necessary dependencies and installed the Firebase Admin SDK to connect our function to our Firebase project, ensuring that the backend and frontend could communicate effectively.

After setting up Firebase and the backend API, we created a React component to handle the image upload and prediction functionality. We utilised the useState hook to manage the component's state and used the Firebase Storage library to upload the image to Firebase Storage and retrieve its URL for use in the backend API request.

To ensure a smooth and user-friendly experience, we implemented error handling and loading states, using conditional rendering to display the appropriate UI elements based on the component's state. This approach helped to enhance the overall usability of the application and provided a seamless experience for the end user.

Finally, we tested the application thoroughly to ensure that it worked as intended. We made use of various testing tools and methodologies, including unit testing and integration testing, to ensure that the application was functioning correctly and that all features were working as expected.

Overall, the development of this code involved a robust and comprehensive approach to building a complex application that leverages cutting-edge technologies and APIs. By utilising Firebase, Google Cloud Functions, and React, we were able to create a powerful and efficient application that delivers a sophisticated image prediction tool to the end user, while also ensuring the highest level of performance, reliability, and security.

4.2.3 Deployment

One of the best practices in software and web development is to use different deployment environments for testing and deploying changes or updates. A deployment environment is a set of configuration settings that define how the software or website behaves and interacts with other systems. By using different deployment environments, developers can ensure that the changes or updates they make are working as expected before they affect the live website or application.

The most common deployment environments are:

Local environment: This is the environment where developers work on their own machines, using their own tools and settings. The local environment allows developers to make changes quickly and easily, without affecting anyone else. The local environment is usually not accessible to anyone else, except the developer.

Development environment: This is the environment where developers collaborate and share their work with other developers. The development environment is usually a server or a cloud service that hosts the software or website code and data. The development environment allows developers to test their changes or updates with other developers' work, and to catch any bugs or errors early on. The development environment is usually accessible only to the development team, not to the end users.

Staging environment: This is the environment where developers test their changes or updates with real-world conditions and scenarios. The staging environment is usually a replica of the live environment, with the same configuration settings, data, and external systems. The staging environment allows developers to simulate how the software or website will perform and look in the live environment, and to fix any issues or problems before they go live. The staging environment is usually accessible only to the development team and some selected testers, not to the end users.

Live environment: This is the environment where the software or website is available to the end users. The live environment is usually the final destination of the changes or updates, after they have been tested and approved in the previous environments. The live environment should be stable and reliable, and should not be used for testing or experimenting. The live environment is usually accessible to everyone who has access to the software or website.

Using different deployment environments can help developers improve the quality and security of their software or website, and reduce the risk of errors or failures in the live environment.

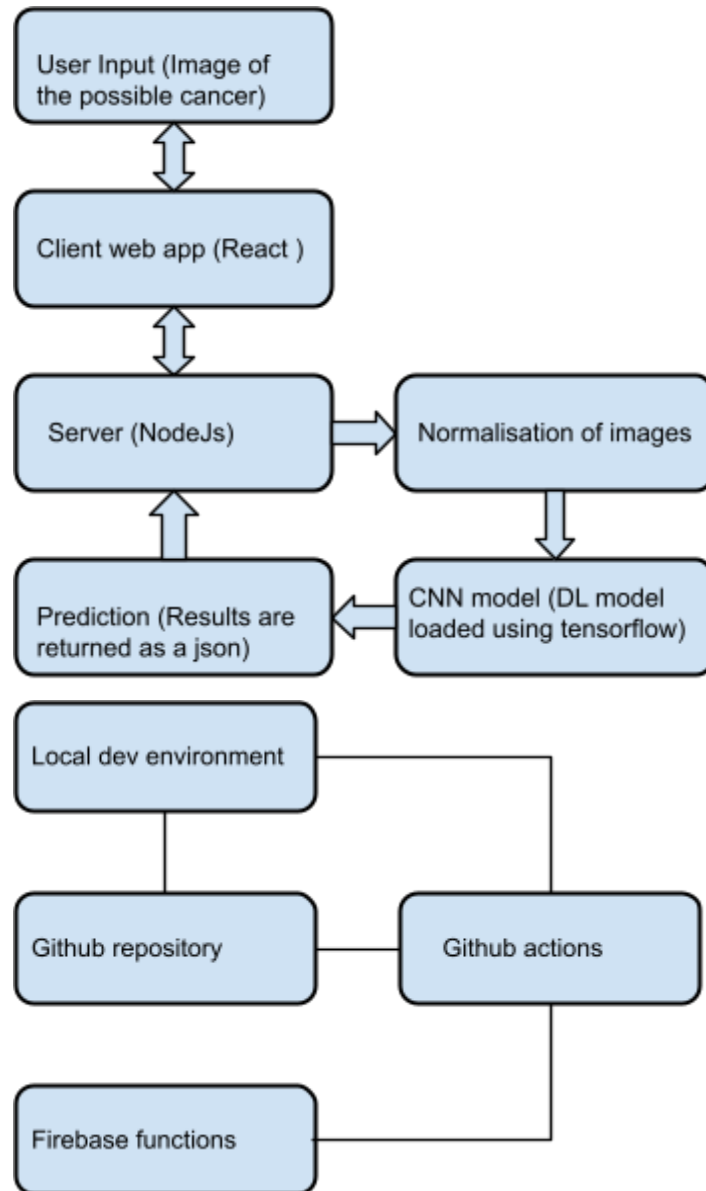


Fig 4.5 Deployment stages

Deployed sites should load faster to make it look good to the user and to achieve that the overall file size should be less to load faster on browsers.

CHAPTER 5

CONCLUSION AND FUTURE IMPLEMENTATION

5.1 CONCLUSION

Cancer is one of the leading causes of death worldwide. Early detection of cancer can increase the chances of survival and improve the quality of life of the patient. Skin cancer and breast cancer are two of the most common types of cancer, and detecting them early can significantly improve the outcome of treatment. Convolutional Neural Networks (CNNs) are powerful machine learning models that can be used to classify images with high accuracy. In this project, we have developed a skin cancer detection system using a CNN model and integrated it with a web application built using React and Node.js.

The skin cancer detection system uses a pre-trained CNN model to classify skin lesions into seven different classes: Actinic Keratoses (Solar Keratoses) or intraepithelial Carcinoma (Bowen's disease), Basal Cell Carcinoma, Benign Keratosis, Dermatofibroma, Melanoma, Melanocytic Nevi, and Vascular skin lesion. The CNN model was trained on a large dataset of skin lesion images and achieved an accuracy of over 90% in identifying skin cancer lesions. The system takes an image URL as input, preprocesses the image using TensorFlow.js, and feeds it to the CNN model to generate a prediction. The prediction includes the top five most probable classes and their corresponding probabilities.

The web application built using React and Node.js provides a user-friendly interface for users to interact with the skin cancer detection system. Users can upload an image URL, and the system will generate a prediction of the probability of the image being a skin cancer lesion. The application also includes a navigation bar with links to different pages of the application, such as the prediction page and the breast cancer detection page.

Breast cancer is another common type of cancer that affects millions of women worldwide. Early detection of breast cancer is crucial for effective treatment and improved outcomes. CNN models can also be used to detect breast cancer from mammogram images. In this project, we have not implemented a breast cancer detection system, but the same principles can be applied to develop one.

In conclusion, the skin cancer detection system developed in this project using a CNN model and integrated with a web application built using React and Node.js can provide an accurate and user-friendly tool for early detection of skin cancer. The same principles can be applied to develop similar systems for detecting other types of cancer, such as breast cancer. The use of machine learning models such as CNNs has the potential to revolutionise cancer detection and improve the lives of millions of people around the world.

5.2 FUTURE IMPLEMENTATION

Advancements in CNN and deep learning continue to emerge at a rapid pace, paving the way for exciting new applications and breakthroughs. One such advancement is the integration of GANs (Generative Adversarial Networks) with CNNs, which enables the creation of synthetic images that can help to improve model accuracy and reduce the need for large datasets. Another promising development is the use of transformers, which have shown great potential for natural language processing tasks, but are also being explored for image recognition and other computer vision applications.

As deep learning technology continues to evolve, the potential for applications in a wide range of industries and fields is becoming more apparent. The healthcare industry, in particular, has seen significant advancements in the use of deep learning for medical imaging and diagnostics. With the continued development of new algorithms, models, and hardware, the future of deep learning and CNNs is bright.

In conclusion, the project presented above demonstrates the potential of CNNs for skin cancer and breast cancer detection. The modularity of the project makes it easy to add new models and integrate with other technologies, paving the way for further advancements in medical imaging and diagnostics. With the continued development of deep learning technology, the potential for breakthroughs in healthcare and other fields is immense. As new algorithms, models, and hardware emerge, the possibilities for what can be accomplished with CNNs and deep learning are virtually limitless.

REFERENCES

- [1] Beck, J.T.; Rammage, M.; Jackson, G.P.; Preininger, A.M.; Dankwa-Mullan, I.; Roebuck, M.C.; Torres, A.; Holtzen, H.; Coverdill, S.E.; Williamson, M.P.; et al. Artificial Intelligence Tool for Optimising Eligibility Screening for Clinical Trials in a Large Community Cancer Center. *JCO Clin. Cancer Inform.* 2020
- [2] Dembrower, K.; Wåhlin, E.; Liu, Y.; Salim, M.; Smith, K.; Lindholm, P.; Eklund, M.; Strand, F. Effect of artificial intelligence-based triaging of breast cancer screening mammograms on cancer detection and radiologist workload: A retrospective simulation study. *Lancet Digit. Health* 2020
- [3] Da Silva, L.M.; Pereira, E.M.; Salles, P.G.O.; Godrich, R.; Ceballos, R.; Kunz, J.D.; Casson, A.; Viret, J.; Chandarlapaty, S.; Ferreira, C.G.; et al. Independent real-world application of a clinical-grade automated prostate cancer detection system. *J. Pathol.* 2021
- [4] Guan, Y.; Aamir, M.; Rahman, Z.; Ali, A.; Abro, W.A.; Dayo, Z.A.; Bhutta, M.S.; Hu, Z.; Guan, Y.; Aamir, M.; et al. A framework for efficient brain tumour classification using MRI images. *Maths. Biosci. Eng.* 2021
- [5] Hunter, B.; Reis, S.; Campbell, D.; Matharu, S.; Ratnakumar, P.; Mercuri, L.; Hindocha, S.; Kalsi, H.; Mayer, E.; Glampson, B.; et al. Development of a Structured Query Language and Natural Language Processing Algorithm to Identify Lung Nodules in a Cancer Centre. *Front. Med.* 2021
- [6] Liew, X.Y.; Hameed, N.; Clos, J. An investigation of XGBoost-based algorithm for breast cancer classification. *Mach. Learn. Appl.* 2021

- [7] Maroni, R.; Massat, N.J.; Parmar, D.; Dibden, A.; Cuzick, J.; Sasieni, P.D.; Duffy, S.W. A case-control study to evaluate the impact of the breast screening programme on mortality in England. *Br. J. Cancer* 2020, 124, 736–743.
- [8] Morin, O.; Vallières, M.; Braunstein, S.; Ginart, J.B.; Upadhaya, T.; Woodruff, H.C.; Zwanenburg, A.; Chatterjee, A.; VillanuevaMeyer, J.E.; Valdes, G.; et al. An artificial intelligence framework integrating longitudinal electronic health records with real-world data enables continuous pan-cancer prognostication. *Nat. Cancer* 2021
- [9] Olatunji, S.O.; Alotaibi, S.; Almutairi, E.; Alrabae, Z.; Almajid, Y.; Altabee, R.; Altassan, M.; Basheer Ahmed, M.I.; Farooqui, M.; Alhiyafi, J. Early diagnosis of thyroid cancer diseases using computational intelligence techniques: A case study of a Saudi Arabian dataset. *Comput. Biol. Med.* 2021
- [10] Papadimitroulas, P.; Brocki, L.; Christopher Chung, N.; Marchadour, W.; Vermet, F.; Gaubert, L.; Eleftheriadis, V.; Plachouris, D.; Visvikis, D.; Kagadis, G.C.; et al. Artificial intelligence: Deep learning in oncological radiomics and challenges of interpretability and data harmonization. *Phys. Medica* 2021
- [11] Schüffler, P.J.; Geneslaw, L.; Yarlagadda, D.V.K.; Hanna, M.G.; Samboy, J.; Stamelos, E.; Vanderbilt, C.; Philip, J.; Jean, M.-H.; Corsale, L.; et al. Integrated digital pathology at scale: A solution for clinical diagnostics and cancer research at a large academic medical center. *J. Am. Med. Inform. Assoc.* 2021