CHAPTER 1

INTRODUCTION

Cardiovascular disease (CVD) remains a leading cause of mortality globally, responsible for nearly 18 million deaths annually, as reported by the World Health Organization (WHO). The burden of CVD is not only medical but also societal, as it contributes to substantial economic costs and diminished productivity. Early detection and effective risk stratification are critical for addressing this challenge. Conventional risk assessment tools, such as the Pooled Cohort Equation (PCE), QRISK3, and the Framingham Risk Score, rely on clinical parameters like blood pressure, cholesterol levels, and lifestyle factors. However, these methods often require invasive procedures, frequent monitoring, and access to healthcare infrastructure, which are not universally available.

Recent advancements in artificial intelligence (AI) and medical imaging offer the promise of noninvasive, scalable, and efficient alternatives, enabling broader access to risk prediction and preventive care. The retina provides a unique, non-invasive window into systemic cardiovascular health. Retinal imaging enables the assessment of microvascular changes, such as vessel caliber and tortuosity, which correlate strongly with cardiovascular conditions like hypertension, atherosclerosis, and heart disease. By leveraging these insights, AI-driven approaches, particularly deep learning, have emerged as powerful tools for analyzing retinal images.

DenseNet-121, a state-of-the-art convolutional neural network (CNN), has shown exceptional capability in capturing complex visual patterns, making it well-suited for cardiovascular risk prediction. Integrating retinal imaging with AI provides a costeffective and accessible solution, especially in resource-limited settings. This technology has the potential to bridge healthcare disparities, offering preventive screening to underserved populations and reducing the global burden of CVD.

A crucial dataset contributing to the development of such technologies is the Diabetic Retinopathy Detection dataset, available on Kaggle. This dataset includes a large set of high-resolution retinal images taken under a variety of imaging conditions. Each subject is represented by left and right eye images labeled with a subject ID (e.g., 1_left.jpeg for the left eye of patient ID 1). Clinicians have graded the presence of diabetic retinopathy (DR) in these images on a scale from 0 to 4, where 0 indicates no DR and 4 indicates proliferative DR. The dataset is challenging due to the variation in camera models and imaging conditions, with some images shown anatomically and others inverted, as seen through a condensing lens. Additionally, the dataset contains real-world noise such as artifacts, focus issues, and exposure inconsistencies, requiring the development of robust algorithms capable of performing reliably in the presence of such variation.

Beyond the healthcare benefits, AI-driven retinal imaging has significant social and environmental implications. By offering non-invasive, needle-free screening methods, this technology could increase participation in health programs, particularly among individuals with trypanophobia or those in regions with limited access to traditional medical facilities. Additionally, it reduces the dependency on invasive tests and disposable medical supplies, such as syringes and test tubes, contributing to a reduction in medical waste. These environmental benefits align with global efforts to promote sustainable healthcare practices. Furthermore, the wide adoption of such technologies could alleviate the strain on overburdened healthcare systems by enabling earlier detection and reducing the need for costly late-stage interventions.

This paper examines two key contributions to this transformative field. First, a systematic scoping review highlights the current landscape of research integrating retinal imaging and deep learning for cardiovascular risk prediction, identifying advancements and gaps. Second, a validation study evaluates Reti-CVD, a novel AI-driven retinal biomarker, against traditional tools like the PCE, QRISK3, and Framingham Risk Score. Together, these studies illustrate the potential of retinal imaging to revolutionize cardiovascular care, not only through improved medical outcomes but also by addressing social inequities and promoting environmentally sustainable healthcare practices. As the field advances, rigorous validation and thoughtful integration into healthcare systems will be essential to fully realize the societal and environmental benefits of these innovations.

CHAPTER 2

PROBLEM DEFINITION

Cardiovascular disease (CVD) remains one of the leading causes of death globally, accounting for millions of deaths annually. Early identification of high-risk individuals is crucial for effective prevention and timely intervention. Traditional methods of assessing CVD risk, such as the use of clinical data including blood pressure, cholesterol levels, and lifestyle factors, often rely on invasive procedures or laboratory tests. These assessments are not always easily accessible, especially in resource-limited settings, and can delay the detection of cardiovascular conditions. Furthermore, many of these conventional methods are based on aggregated risk factors, which can miss subtle but important changes in an individual’s cardiovascular health.

In contrast, retinal images provide a unique, non-invasive window into the vascular system, enabling the observation of microvascular changes that can indicate systemic cardiovascular health. Retinal vessels, due to their direct connection to the body’s vascular network, can reflect the underlying condition of the cardiovascular system, offering valuable insights into diseases such as hypertension, atherosclerosis, and diabetic retinopathy.

This study aims to address this gap by applying deep learning techniques, specifically the DenseNet-121 convolutional neural network (CNN), to analyze retinal images for cardiovascular risk prediction. DenseNet-121 is a state-of-the-art model known for its efficiency in extracting and processing intricate visual features from medical images. This approach leverages the power of deep learning to automate the analysis of retinal images, identifying relevant features that are indicative of cardiovascular risk. By doing so, it aims to provide a highly accurate, non-invasive, and cost-effective solution for CVD risk assessment that could be easily deployed in routine clinical practice.

CHAPTER 3

LITERATURE REVIEW

The use of retinal images for predicting cardiovascular disease (CVD) risk has gained significant traction in recent years, particularly with the advent of deep learning models that can automate and improve the accuracy of analysis. Retinal imaging offers a noninvasive, accessible method to examine the vascular health of an individual, which is closely linked to cardiovascular conditions. Various studies have explored how deep learning can be leveraged to predict cardiovascular risk through the analysis of retinal images, utilizing advanced neural networks and machine learning algorithms.

One notable approach is presented by Zhang et al. (2023), where a deep learning model is used to enhance stability in cardiovascular disease risk prediction by leveraging retinal images. Their work emphasizes the potential of using deep learning techniques to refine the predictability of CVD, highlighting the stability and accuracy improvements over traditional methods [1]. Similarly, Kujalambal et al. (2023) utilized a neural network algorithm to predict heart disease using retinal images, demonstrating that deep learning models can detect subtle retinal features that correlate with cardiovascular risk factors such as hypertension and atherosclerosis [2].

The integration of deep learning with existing risk factors has also been explored in studies like Mellor et al. (2023), where deep learning on retinal images is combined with known risk factors to augment CVD prediction in diabetic patients. Their prospective cohort study found that adding retinal image analysis could enhance the predictive power of traditional risk models in diabetes [3]. A broader review by Hu et al. (2023) systematically examined the application of deep learning techniques in cardiovascular risk prediction, synthesizing findings from multiple studies and confirming that retinal imaging is a promising tool for CVD risk assessment. This review also provided a metaanalysis of the effectiveness of these models across various populations [4].

In addition to predictive models, research also focuses on identifying abnormalities in retinal images that may signal cardiovascular risk. For example, Prakash et al. (2024) concentrated on identifying retinal abnormalities to predict CVD, using deep learning algorithms to classify retinal features linked to cardiovascular conditions [5]. Other works, such as those by Shaikh et al. (2023), have emphasized the broader application of retinal imaging for heart disease prediction, proposing that this method could become a standard screening tool due to its ease of use and cost-effectiveness [6].

Recent advancements have also compared deep learning-based retinal biomarkers with existing clinical risk scores. Yi et al. (2023) conducted a study assessing the effectiveness of a deep-learning-based retinal biomarker (Reti-CVD) in comparison with established CVD risk scores, such as the Pooled Cohort Equations (PCE) and Framingham Risk Scores. Their findings suggest that retinal biomarkers can significantly aid in identifying high-risk individuals, especially in diverse populations [7]. Barriada and Masip (2023) provide an overview of various deep-learning methods for cardiovascular risk assessment, underscoring the growing interest and progress in this field [8]. Additionally, Li et al. (2024) performed a systematic review of studies using deep learning to predict cardiovascular markers from retinal fundus images, contributing valuable insights into how these technologies are evolving and their potential to complement traditional CVD risk assessments [9].

Moreover, Alagona Jr. and Ahmad (2015) provided a foundational perspective on cardiovascular disease risk assessment and prevention by outlining the current guidelines and limitations in ASCVD management. Their review highlights the significant public health impact of ASCVD, emphasizing that it accounts for over one-third of all U.S. deaths and remains a leading cause of disability [10]. Key insights from their work include the identification of major ASCVD risk factors through observational studies and the transformative role of statin therapy in reducing cardiovascular events. They also stress the importance of patient-centered care, integrating evidence-based clinical recommendations with personalized prevention strategies. While these guidelines have led to substantial progress in CVD management, significant gaps remain in identifying atrisk individuals and preventing recurrent cardiovascular events [10].

Muhammad Mateen et al. (2023) conducted a comprehensive survey on the identification of diabetic retinopathy (DR) based on nearly 150 research articles. Their work summarizes the collection of retinal datasets, various methodologies adopted for DR detection, and the performance evaluation metrics used to represent outcomes. They discuss retinal datasets initially, followed by explanations of approaches for detecting retinal abnormalities, including neovascularization, hemorrhages, microaneurysms, and exudates. Furthermore, the study emphasizes the role of evaluation metrics in computeraided diagnosis (CAD) systems and provides a detailed discussion on the significance of deep learning-based approaches. The authors also offer future research directions to address challenges in DR detection [11].

K. Shankar et al. (2023) proposed the HPTI-v4 model, which incorporates segmentation through feature extraction processes based on histograms and Inception v4. Bayesian optimization is employed for hyperparameter tuning in Inception v4, followed by classification using an MLP. Experimental results reveal that the HPTI-v4 model achieves exceptional performance, with accuracy, sensitivity, and specificity rates of 99.49%, 98.83%, and 99.68%, respectively. This model is positioned as an automated diagnostic tool for DR image classification. The study also underscores the broader health implications of diabetes, including kidney failure, retinal infection, nerve damage, and increased risk of heart attacks and strokes due to diabetic neuropathy and retinopathy [12].

The European Society of Cardiology and the European Atherosclerosis Society proposed the SCORE2 risk prediction algorithms, derived from individual-participant data encompassing 45 cohorts in 13 countries. These algorithms account for variables such as age, sex, smoking status, systolic blood pressure, and lipid profiles, recalibrated to reflect four European risk regions using population-specific CVD incidence and risk factor distributions. In external validation, the models demonstrated robust performance with Cindices ranging from 0.67 to 0.81 across different cohorts. Notably, predicted 10-year CVD risk varied significantly among regions, highlighting the importance of localized adjustments in cardiovascular risk prediction models [13].

Huang et al. (2020) investigated the association between cardiac structure and retinal vascular geometry in a study involving 50 participants without cardiovascular disease. Using transthoracic echocardiography, cardiac structure indices such as left ventricular internal diameter end diastole index, left ventricular internal diameter end systole index, left ventricular mass index, and left atrial volume index were measured. Retinal imaging was used to analyze vascular geometric indices, including branching angle, curvature tortuosity, and fractal dimension. Multiple linear regressions, adjusted for variables like age, sex, BMI, and comorbidities, revealed significant associations. For instance, each unit increase in cardiac structure indices corresponded to larger retinal arteriolar branching angles. The findings suggest that retinal vascular geometry could provide insights into cardiac structural changes, underscoring the potential of retinal imaging in broader cardiovascular assessments [14].

These studies collectively demonstrate the potential of deep learning in transforming cardiovascular risk prediction. By automating the analysis of retinal images, deep learning models offer a scalable, non-invasive, and cost-effective alternative to traditional risk assessment tools. The growing body of research highlights not only the accuracy and reliability of these methods but also their capacity to improve early detection, allowing for timely intervention and better patient outcomes.

CHAPTER 4

PROJECT DESCRIPTION

This project focuses on predicting cardiovascular disease (CVD) risk using retinal images, harnessing the power of deep learning, specifically the DenseNet-121 convolutional neural network (CNN). Retinal images, which reflect the condition of the body's vascular system, offer valuable insights into a person's cardiovascular health. By analyzing these images, it is possible to detect signs of CVD risk factors such as hypertension, atherosclerosis, and microvascular changes, often before clinical symptoms manifest.

The proposed solution uses the DenseNet-121 model, a state-of-the-art deep learning architecture known for its efficiency in feature extraction through dense connectivity. This model is trained to recognize patterns and abnormalities in retinal images that correlate with CVD risk. The system will categorize individuals into risk groups based on their retinal scans, providing a non-invasive, rapid, and scalable alternative to traditional risk assessment methods. By automating this process, the project aims to make CVD risk prediction more accessible, particularly in resource-limited settings, improving early detection and prevention of cardiovascular diseases.

The primary dataset used in this project is the Diabetic Retinopathy Detection dataset, available on Kaggle, which provides a rich collection of retinal images. Although primarily designed for diabetic retinopathy detection, the dataset is highly relevant for cardiovascular disease (CVD) prediction as both conditions share common risk factors and physiological markers. The dataset consists of a variety of high-resolution retinal images, each captured under different imaging conditions, and is annotated with severity labels that reflect the presence of abnormal vascular changes in the retina.

The dataset contains images from a diverse patient population, capturing both left and right eye scans of individuals with varying degrees of diabetic retinopathy. These images are classified into five categories based on the severity of retinopathy, which also correlates with cardiovascular risks.

The categories are as follows:

- 0: No Diabetic Retinopathy – No visible abnormalities are present in the retinal image, indicating a low risk of cardiovascular disease.
- 1: Mild Diabetic Retinopathy – Mild signs of retinopathy are present, which may suggest early-stage vascular changes linked to cardiovascular risks.
- 2: Moderate Diabetic Retinopathy – Clear signs of vascular abnormalities, such as microaneurysms and hemorrhages, which are indicative of increasing cardiovascular risk.
- 3: Severe Diabetic Retinopathy – Significant retinal damage is visible, often linked with advanced microvascular changes, which correlate with heightened cardiovascular risk.
- 4: Proliferative Diabetic Retinopathy – The most severe stage of retinopathy, with significant growth of abnormal blood vessels, which often occurs in individuals with longstanding hypertension or heart disease.

Key Features of the Dataset:

- Image Format and Size: The retinal images in the dataset are provided in JPEG format with a resolution of 512x512 pixels. These images offer sufficient detail to capture minute vascular features and abnormalities, essential for predicting cardiovascular risk factors.
- Data Imbalance: A key consideration in this dataset is the imbalance in the distribution of the severity labels, where a higher proportion of images may fall into the "no diabetic retinopathy" (category 0) and lower-risk categories. This imbalance must be addressed during model training to ensure that the classifier does not bias its predictions towards the majority class.

- Multiple Imaging Conditions: The dataset includes retinal images taken under varying conditions—different camera models, lighting variations, and image quality. Some images may have noise, such as blurriness or artifacts, which adds complexity to the task of training a deep learning model capable of accurately identifying subtle signs of cardiovascular risk.
- Anatomical Variations: The dataset features images of both left and right eyes, and as mentioned earlier, anatomical differences may lead to inversions in image orientation. The system needs to be able to handle these variations to ensure that the features being extracted are consistent, regardless of the eye being analyzed.
- Annotations: Each retinal image is annotated with a severity grade (ranging from 0 to 4) for diabetic retinopathy. Although this grading system is specific to diabetic retinopathy, these severity labels are useful for the CVD risk prediction task, as the vascular changes associated with diabetic retinopathy often overlap with the signs of cardiovascular disease, such as hypertension and atherosclerosis.

By leveraging this dataset, the project aims to train and evaluate the DenseNet-121 model to predict CVD risk based on retinal features. The approach will provide a transformative, AI-driven solution for early cardiovascular risk assessment, bridging the gap in healthcare accessibility and offering a scalable tool for routine clinical use.

CHAPTER 5

REQUIREMENTS

5.1 Hardware Requirements

- Processor:

  Intel Core i5 or i7 (recommended for faster processing); Intel Core i9 or AMD Ryzen 7/9 for optimal performance with large datasets and complex models.

- GPU:

  Dedicated GPU with CUDA support (e.g., NVIDIA GTX 1660 or higher; RTX 3060/3070 recommended for deep learning tasks).

- RAM:

  At least 8 GB (16 GB or more recommended for handling large image datasets and parallel processing).

- Cooling System:

  Adequate cooling system for maintaining optimal hardware performance during prolonged model training tasks.

- Power Supply:

  A reliable power supply unit to support high-performance GPUs and processors.

- Display:

  Full HD monitor (1920x1080) or higher resolution for data visualization and analysis.

- Other Peripherals:

  Keyboard, mouse, and optionally a second monitor to facilitate multitasking during development and analysis.

5.2 Software Requirements

Operating System:

Windows 10/11, Ubuntu 20.04 or higher, macOS (with Python and TensorFlow support).

Programming Language:

Python 3.7 or higher.

Libraries/Frameworks:

- TensorFlow/Keras for deep learning model development.
- NumPy for numerical computations.
- OpenCV or PIL for image preprocessing.
- Matplotlib/Seaborn for data visualization.
- Pandas for dataset management.

IDE:

Jupyter Notebook or VS Code (with Jupyter extensions).

Other Tools:

Git for version control.

5.3 Functional Requirements

Image Upload and Preprocessing:

Allow users to upload retinal images individually or in batches.Resize and normalize images to meet model input requirements.

Feature Extraction:

Use DenseNet-121 to extract features from the input images.

Classification/Regression Model

Perform regression for predicting continuous variables like CVD risk scores.

Model Training and Fine-Tuning:

Fine-tune DenseNet-121 with domain-specific datasets.

Visualization:

Display preprocessed images and extracted features. Show performance metrics such as accuracy, precision, and recall.

Data Augmentation and Validation:

Apply data augmentation to improve model generalization. Split the dataset into training, validation, and test sets.

5.4 Non-Functional Requirements

Performance:

Real-time or near-real-time feature extraction and predictions. Efficient utilization of GPU for deep learning computations.

Scalability:

Ability to handle large datasets without significant performance degradation.

Reliability:

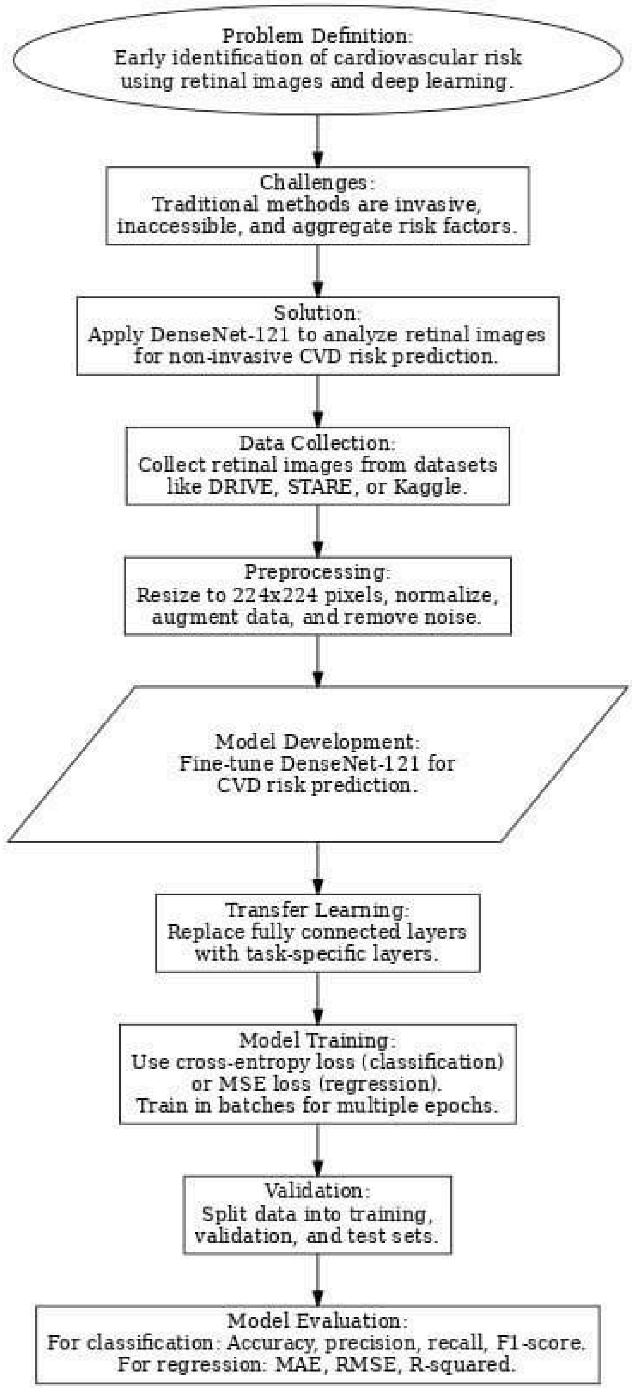
Ensure accurate and consistent predictions. Handle incomplete or lowquality input images gracefully.

Usability:

Provide a simple interface for users to upload images and view results.

CHAPTER 6

DESIGN



6.1(a)Proposed Block Diagram

The proposed design begins with defining the problem and addressing challenges with traditional methods. The proposed solution involves leveraging DenseNet-121, a deep learning model, for retinal image analysis. The design includes steps for data collection from public datasets, preprocessing, and fine-tuning DenseNet-121 using transfer learning by replacing fully connected layers with task-specific ones. The model is trained with appropriate loss functions and validated through dataset splitting into training, validation, and test sets. This pipeline aims to develop an accurate, efficient, and noninvasive tool for CVD risk prediction

- Problem Definition:
  - This block defines the objective of the project: "Early identification of cardiovascular risk using retinal images and deep learning."
  - The focus is on leveraging advanced technologies to predict cardiovascular disease (CVD) risk in a non-invasive and efficient manner.
- Challenges:
  - This block highlights the limitations of traditional methods for assessing cardiovascular risk:
    - They are often invasive (e.g., blood tests or imaging procedures).
    - Accessibility is limited, particularly in resource-constrained environments.
    - Conventional methods typically rely on aggregated risk factors, which may miss individual-level nuances.
- Solution:
  - The proposed solution is outlined here: applying the DenseNet-121 convolutional neural network (CNN) to analyze retinal images. This provides a non-invasive, scalable alternative for cardiovascular risk prediction.
- Data Collection:
  - This block specifies the source of retinal images:
    - Datasets such as DRIVE, Kaggle are used for collecting the required data.

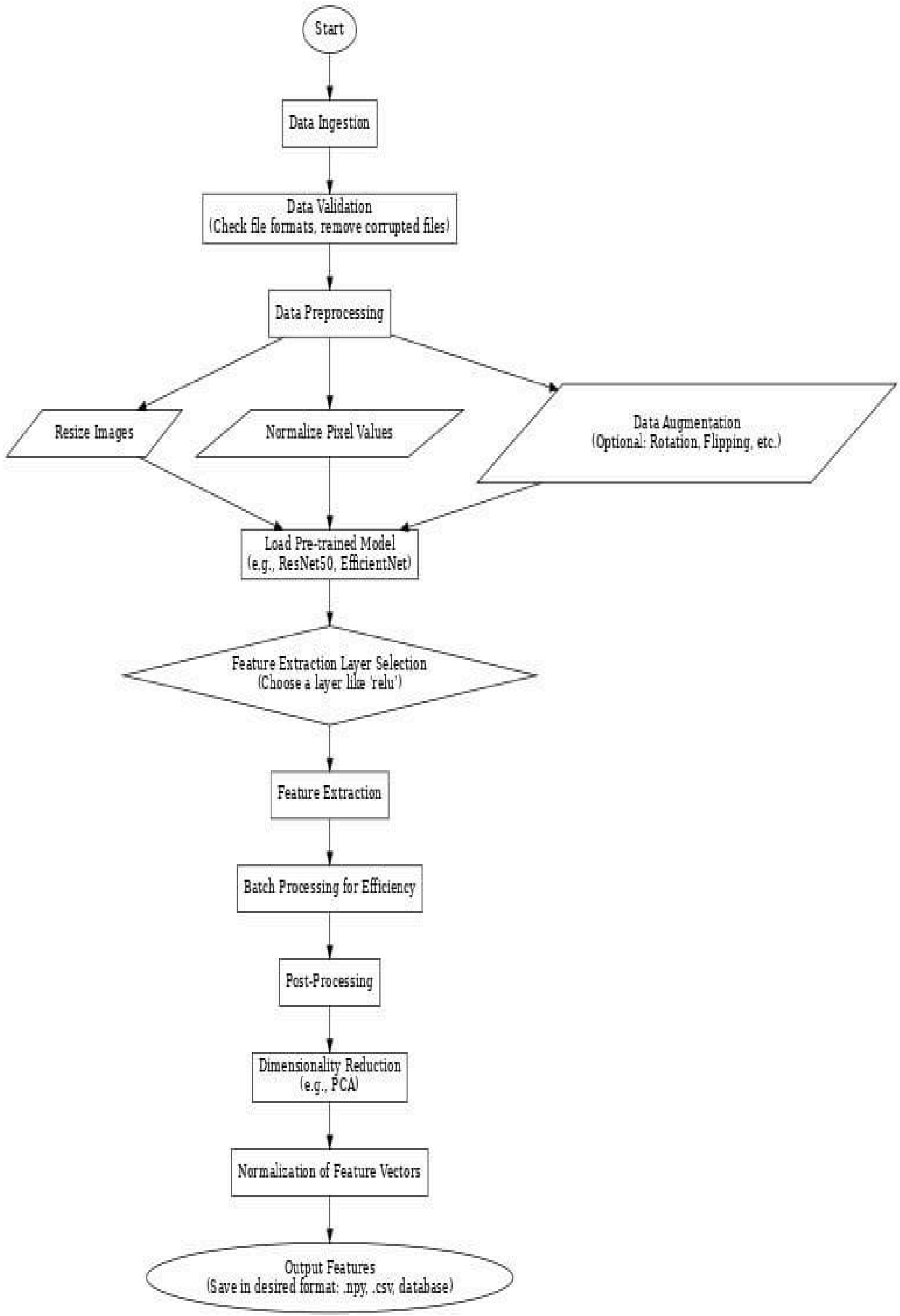
These datasets typically include annotated retinal images needed for model development and evaluation.

Preprocessing:
This step involves preparing the raw retinal images for training:

- Resizing images to a standard size (e.g., 224x224 pixels) for compatibility with DenseNet-121.
- Normalizing the pixel values to enhance model performance.
- Augmenting the dataset by applying transformations (e.g., rotation, flipping) to increase diversity.

Removing noise and artifacts to improve data quality.

- Model Development:
  - o Fine-tuning DenseNet-121, a pre-trained CNN, to suit the specific task of CVD risk prediction using retinal images.
- Transfer Learning:
  - o The pre-trained model's fully connected layers are replaced with taskspecific layers suited for either classification or regression tasks.
- Model Training:
  - o Loss functions like cross-entropy (for classification) or mean squared error (MSE, for regression) are used. o The model is trained in batches over multiple epochs, allowing it to learn features from the retinal images.
- Validation:
  - o Training set: Used for learning the model parameters. o
  - o Validation set: Used for tuning hyperparameters and avoiding overfitting.
  - o Test set: Used for final performance evaluation.
- Model Evaluation:
  - o This block describes the metrics used to assess the model’s performance:
    - ▪ For classification tasks: Accuracy, precision, recall, and F1-score are used.


Start
Data Ingestion
Data Validation
(Check file formats, remove corrupted files)
Data Preprocessing
Resize Images
Normalize Pixel Values
Data Augmentation
(Optional: Rotation, Flipping, etc.)
Load Pre-trained Model
(e.g., ResNet50, EfficientNet)
Feature Extraction Layer Selection
(Choose a layer like 'relu')
Feature Extraction
Batch Processing for Efficiency
Post-Processing
Dimensionality Reduction
(e.g., PCA)
Normalization of Feature Vectors
Output Features
(Save in desired format: .npy, .csv, database)

6.1(b) Design

This flowchart represents a typical workflow for feature extraction in a machine learning or deep learning pipeline. Here's a detailed explanation of each stage in the process:

The process begins with Data Ingestion, where raw data is collected and brought into the system for processing. This stage could involve loading image datasets from local storage, cloud storage, or external databases. The ingested data forms the foundation for all subsequent steps in the pipeline.

Once the data is ingested, it undergoes Data Validation to ensure its quality and usability. This involves verifying the file formats (e.g., .jpg, .png) to confirm they match the required specifications. Additionally, corrupted or unreadable files are identified and removed to prevent errors in downstream processes. This step is critical to maintaining the integrity of the pipeline.

The next stage is Data Preprocessing, which prepares the data for feature extraction. Several key operations are performed here. First, images are resized to ensure they have consistent dimensions that match the input requirements of the chosen model. Second, pixel values are normalized, often scaling them to a range between 0 and 1 or -1 and 1, which helps stabilize and accelerate the training or feature extraction process.

Optionally, Data Augmentation can be applied during this step to artificially increase the size of the dataset by introducing variations such as rotations, flips, or other transformations. These augmentations improve the model's ability to generalize to unseen data.

After preprocessing, a Pre-trained Model is loaded. Models like ResNet50 or EfficientNet, which have been pre-trained on large datasets like ImageNet, are commonly used for feature extraction. These models provide a powerful foundation as they have already learned to identify complex patterns and features in images.

The user must then select a Feature Extraction Layer within the pre-trained model. This is often an intermediate layer (e.g., a layer named "ReLU") where meaningful features are encoded. The choice of layer depends on the specific application and the type of features required for the task.

Once the layer is chosen, the process of Feature Extraction begins. The model processes the input data and extracts high-level feature representations. These features capture essential information about the images, such as edges, textures, and shapes, which are crucial for downstream tasks like classification or clustering.To improve efficiency, Batch Processing is employed. Instead of processing each image individually, data is divided into batches, allowing the model to process multiple images simultaneously. This reduces computation time and optimizes the use of system resources.

After feature extraction, the features undergo Post-Processing to refine and optimize them for further use. One common step in this stage is Dimensionality Reduction, where techniques like Principal Component Analysis (PCA) are used to reduce the number of dimensions in the feature vectors. This eliminates redundant or less informative components, making the features more compact and computationally efficient.

Finally, the processed and normalized features are saved during the Output Features step. The features can be stored in various formats such as .npy (NumPy arrays), .csv files, or even databases, depending on the intended use. These features are now ready to be utilized for tasks like training machine learning models, performing clustering, or serving as input to other systems.

CHAPTER 7

METHODOLOGY

7.1. SYSTEM MODEL

This stage is the basic stage in moving from issue to the strategy space. Appropriately, beginning with what is obliged; chart takes us to run after how to satisfy those prerequisites. Framework plot depicts all the basic information structure, record game-plan, yield & veritable modules in the structure & their description is picked. This accepts a fundamental part in light of the fact that it will give the keep going yield on which it was being working. In our work we are utilizing a few modules, these modules are recorded underneath.

1. Input Image

The input picture is given to the application utilizing the site page that is created utilizing python dash. The server runs in the local machine and tunes in at 8050 port number. We can select picture utilizing the file dialog & select them from the document system directory.

2. Preprocessing & segmentation

Image processing is fundamental for image upgrades. During Preprocessing RGB picture to change over into HSV shading space. This progression was taken on the grounds that HSV shading space was less delicate to enlightenment changes contrasted with RGB. At that point, it was separated, smoothened & lastly, the greatest parallel-connected object was being thought about to stay away from thought of skin-hued protests other than the hand. To get a decent outcome, smoothing and separating are finished. Image segmentation is essentially performed to find the hand object in the picture.

3. Feature Extraction:

Feature Extraction stage is important in light of the fact that specific highlights must be removed so they are novel for each signal. After the choice is made that a sign is available, at that point the last edge is mulled over & features.

4. Classification:

Classification of hand is finished with the assistance of different highlights determined already. The five-bit paired succession is hence created to exceptionally perceive and use these perceived glaucoma sickness. By the component extraction, a significant pinnacle is encoded as 1 while an insignificant pinnacle is encoded as 0 dependent on the crossing point to the limit line. The deep learning model CNN is save the trained model in cnn.h5 model file.

5.Prediction:

Various pictures were tested & tracked down that the new procedure of classification was found to show 97% precision. A few pictures tried with other data set pictures are given in the examination of the outcomes. In the expectation part, the input picture is prepared & includes are removed. The anticipated outcome is shown on the web page that is planned utilizing a dash

7.2. ALGORITHMS USED:

- **CNN Algorithm overview**:

Convolutional Neural Network (CNN) were used to achieve some breakthrough results and win well-known contests. The application of convolutional layers consists in convolving a signal or an image with kernels to obtain feature maps. So, a unit in a feature map is connected to the previous layer through the weights of the kernels. The weights of the kernels are adapted during the training phase by back propagation, in order to enhance certain characteristics of the input. Since the kernels are shared among all units of the same feature maps, convolutional layers have fewer weights to train than dense FC layers, making CNN easier to train and less prone to overfitting. Moreover, since the same kernel is convolved over all the image, the same feature is detected independently of the locating—translation invariance. By using kernels, information of the neighborhood is taken into account, which is an useful source of context information. Usually, a non-linear activation function is applied on the output of each neural unit. If we stack several convolutional layers, the extracted features become more abstract with the increasing depth. The first layers enhance features such as edges, which are aggregated in the following layers as motifs, parts, or objects.

The following concepts steps are important in the context of CNN:

1)Initialization:

It is important to achieve convergence. We use the Xavier initialization. With this, the activations and the gradients are maintained in controlled levels, otherwise back-propagated gradients could vanish or explode.

2)Activation Function:

It is responsible for non-linearly transforming the data. Rectifier linear units (ReLU), defined as

$$f(x)= \max(0, x),$$

were found to achieve better results than the more classical sigmoid, or hyperbolic tangent functions, and speed up training. However, imposing a constant 0 can impair the gradient flowing and consequent adjustment of the weights. We cope with these limitations using a variant called leaky rectifier linear unit (LReLU) that introduces a small slope on the negative part of the function. This function is defined as

$$f(x)= \max(0, x)+\alpha\min(0,x)$$

where is the leakyness parameter. In the last FC layer, we use softmax.

3)Pooling:

It combines spatially nearby features in the feature maps. This combination of possibly redundant features makes the representation more compact and invariant to small image changes, such as insignificant details; it also decreases the computational load of the next stages. To join features it is more common to use max-pooling or average-pooling.

4)Regularization:

It is used to reduce overfitting. We use Dropout in the FC layers. In each training step, it removes nodes from the network with probability. In this way, it forces all nodes of the FC layers to learn better representations of the data, preventing nodes from co-adapting to each other. At test time, all nodes are used. Dropout can be seen as an ensemble of different networks and a form of bagging, since each network is trained with a portion of the training data.

5)Data Augmentation:

It can be used to increase the size of training sets and reduce overfitting. Since the class of the patch is obtained by the central voxel, we restricted the data augmentation to rotating operations.

Network Architecture:

Image-Input Layer:

An imageInput Layer is the place you initialize the size of input image, here, 128-by-128-by-1 is used. These numbers represent height, width, and the number of channels. In this case, input data is a grayscale image, hence the number of channel is 1.

Convolutional Layer:

Input arguments for this layer are filtering size, the number of filters, and padding. Here, the filter of size 10 is used, which determines 10 x 10 filter. The number of channels used is 10, means 10 neurons are connected. Padding of 1 specifies that the size of the output image is same as that of an input image.

ReLU Layer:

ReLU (rectified linear unit) layer is a batch normalization layer, which is placed after initializing a nonlinear activation function. Importance of this layer is to decrease the sensitivity and increase the pace of the training.

Max Pooling Layer:

Max pooling layer is one of the down sampling technique which is used for convolutional layers. In this architecture, poolSize is set to 3 and training function's step size is 3.

Fully Connected Layer:

Fully connected layers follow max pooling layer. In this layer, all the neurons of all layers are interconnected to the previous layer. The given input argument for this layer is 10, which indicate 10 classes.

Softmax Layer:

Fully connected layers are followed by softmax layer, which is normalization technique. This layer generates positive numbers as output such that the sum of numbers is one. Classification layer uses these numbers for classification.

Classification Layer:

Classification layer is the final layer of the architecture. This layer classifies the classes based on probabilities obtained from softmax layer and also calculate cost function.

Training Options:

The maximum number of epochs set to 100 and initial learning rate is 0.001.

Architecture (AlexNet):

This architecture was one of the first deep networks to push ImageNet Classification accuracy by a significant stride in comparison to traditional methodologies. It is composed of 5 convolutional layers followed by 3 fully connected layers, as depicted in Figure.

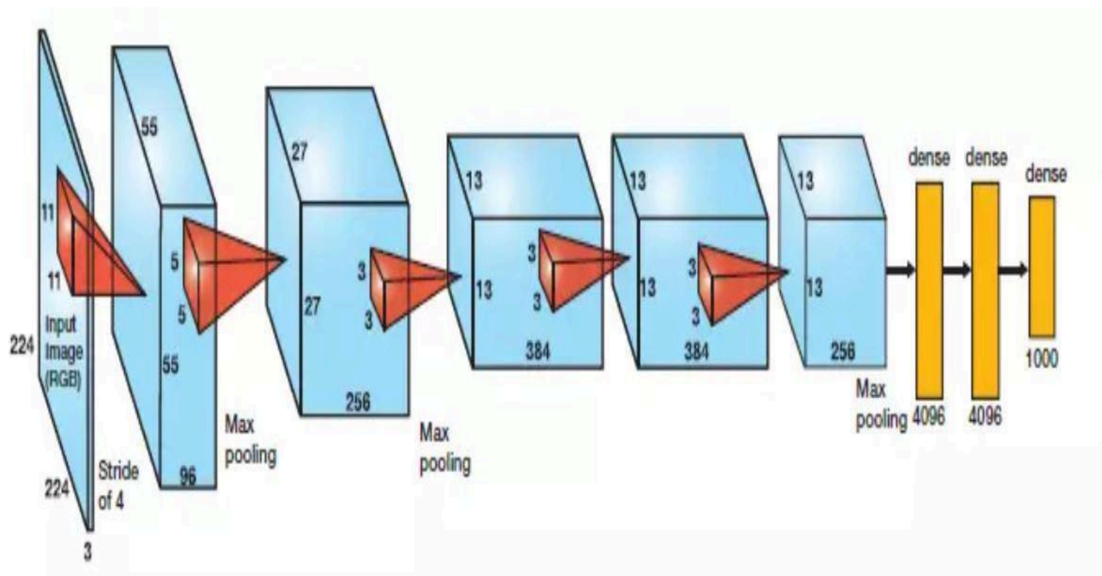


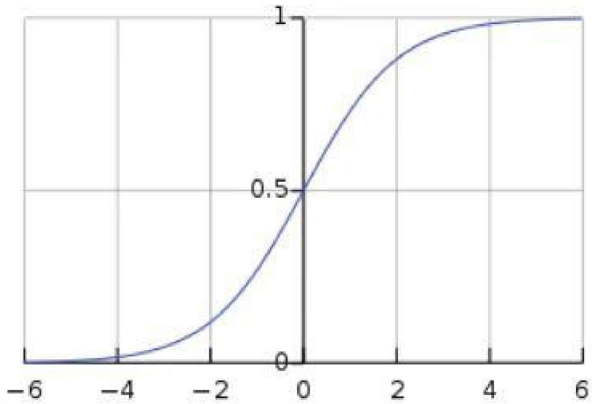
Fig3.1: Architecture of Neural network (AlexNet)

AlexNet, proposed by Alex Krizhevsky, uses ReLu(Rectified Linear Unit) for the non-linear part, instead of a Tanh or Sigmoid function which was the earlier standard for traditional neural networks. ReLu is given by:

$f(x) = \max(0,x)$

The advantage of the ReLu over sigmoid is that it trains much faster than the latter because the derivative of sigmoid becomes very small in the saturating region and therefore the updates to the weights almost vanish. This is called vanishing gradient problem.

In the network, ReLu layer is put after each and every convolutional and fully-connected layers (FC).



Another problem that this archsvitecture solved was reducing the over-fitting by using a Dropout layer after every FC layer. Dropout layer has a probability,(p), associated with it and is applied at every neuron of the response map separately. It randomly switches off the activation with the probability p, as can be seen in figure.

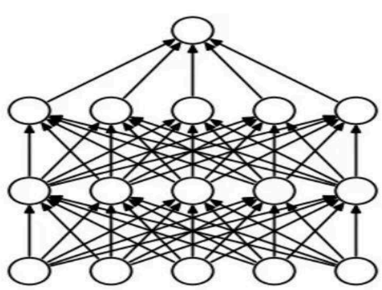


Fig3.2: Standard Neural Net

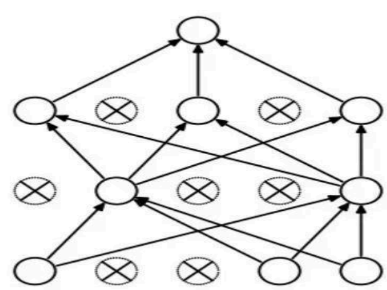


Fig3.3: After applying dropout

DropOut work:

The idea behind the dropout is similar to the model ensembles. Due to the dropout layer, different sets of neurons which are switched off, represent a different architecture and all these different architectures are trained in parallel with weight given to each subset and the summation of weights being one. For n neurons attached to DropOut, the number of subset architectures formed is 2^n. So it amounts to prediction being averaged over these ensembles of models. This provides a structured model regularization which helps in avoiding the over-fitting. Another view of DropOut being helpful is that since neurons are randomly chosen, they tend to avoid developing co-adaptations among themselves thereby enabling them to develop meaningful features, independent of others.

Advantages of algorithm:

- Minimize computation compared to a regular neural network.
- Convolution simplifies computation to a great extent without losing the essence of the data.
- They are great at handling image classification.
- They use the same knowledge across all image locations.
- Execution time is less.
- Achieves a much better performance using AlexNet.
- High prediction accuracy.

7.3. Algorithm works Steps:

Step1: Convolutional Neural Networks:

Convolutional Neural Networks have a different architecture than regular Neural Networks. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer — the output layer — that represent the predictions.

Convolutional Neural Networks are a bit different. First of all, the layers are organised 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.

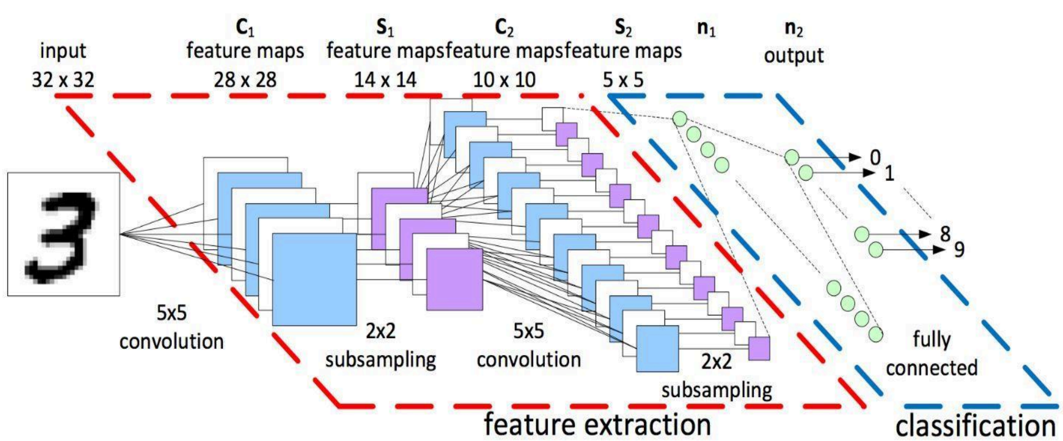


Fig3.4: Convolutional Neural Networks architecture

There are squares and lines inside the red dotted region which we will break it down later. The green circles inside the blue dotted region named classification is the neural network or multi-layer perceptron which acts as a classifier. The inputs to this network come from the preceding part named feature extraction.

Feature extraction is the part of CNN architecture from where this network derives its name. Convolution is the mathematical operation which is central to the efficacy of this algorithm. Lets understand on a high level what happens inside the red enclosed region. The input to the red region is the image which we want to classify and the output is a set of features. Think of features as attributes of the image, for instance, an image of a cat might have features like whiskers, two ears, four legs etc. A handwritten digit image might have features as horizontal and vertical lines or loops and curves. Later we'll see how do we extract such features from the image.

Step2: 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:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Fig3.5: Input image and the filter

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:

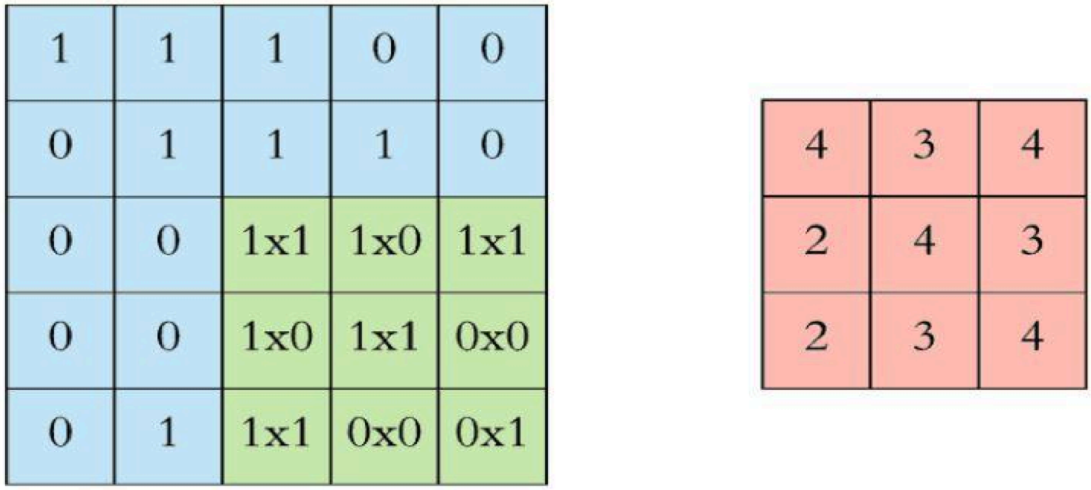


Fig3.6: Filter (green) slides over the input image

In the above Fig3.6 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:

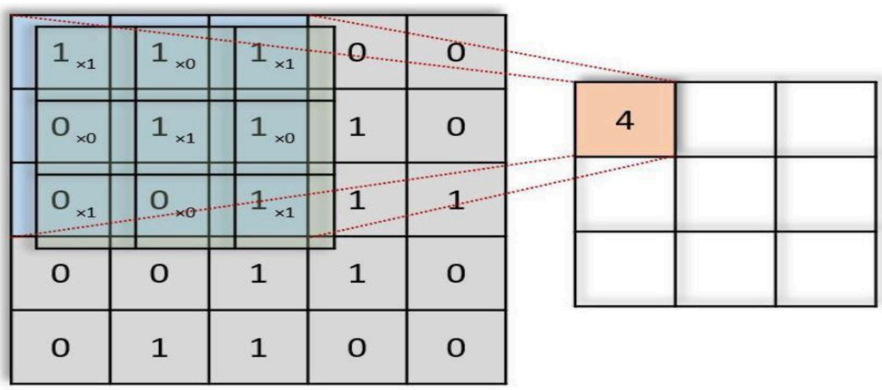


Fig3.7: 1st step of convolution

(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:

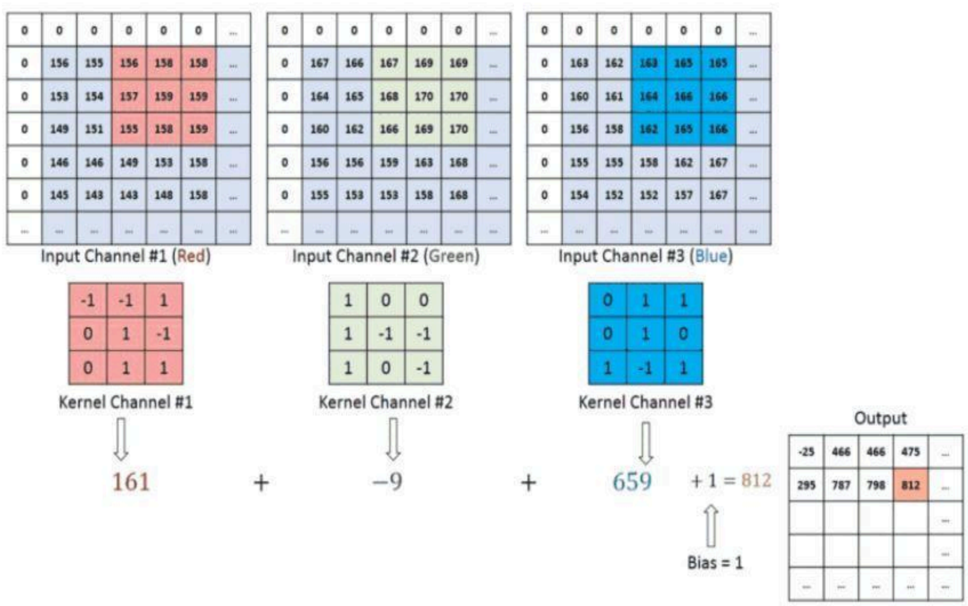


Fig3.8: Squashed one-depth channel convolution feature

Each neuron in the output matrix has overlapping receptive fields. The Fig6 below will give you a better sense of what happens in convolution.

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

With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the data-set, similar to how we would.

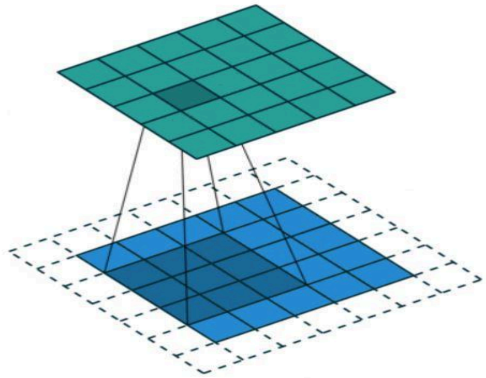


Fig3.9: Convolution example

Step3: 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. In above example our padding is 1.

In our example when we augment the 5x5x1 image into a 7x7x1 image and then apply the 3x3x1 kernel over it, we find that the convoluted matrix turns out to be of dimensions 5x5x1. It means our output image is with same dimensions as our output image (Same Padding).

On the other hand, if we perform the same operation without padding, in the output we'll receive an image with reduced dimensions. So our (5x5x1) image will become (3x3x1).

Feature Extraction: example:

Lets say we have a handwritten digit image like the one below. We want to extract out only the horizontal edges or lines from the image. We will use a filter or kernel which when convoluted with the original image dims out all those areas which do not have horizontal edges:

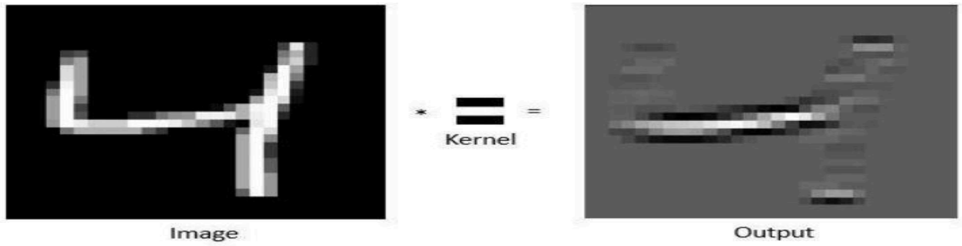


Fig3.10: Horizontal filter example

Notice how the output image only has the horizontal white line and rest of the image is dimmed. The kernel here is like a peephole which is a horizontal slit. Similarly for a vertical edge extractor the filter is like a vertical slit peephole and the output would look like:

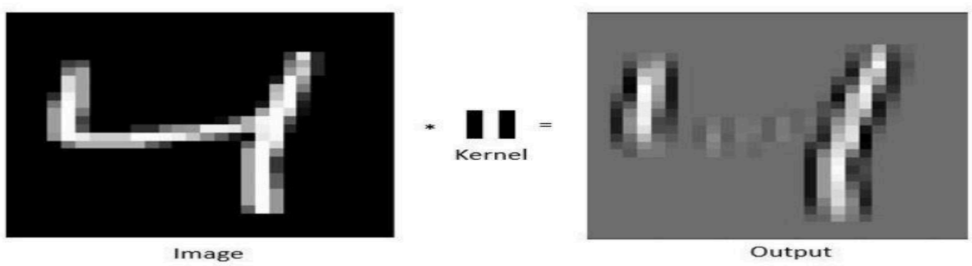


Fig3.11: Vertical filter example

Step4: Feature Extraction: Non-Linearity:

After sliding our filter over the original image the output which we get is passed through another mathematical function which is called an activation function. The activation function usually used in most cases in CNN feature extraction is ReLu which stands for Rectified Linear Unit. Which simply converts all of the negative values to 0 and keeps the positive values the same:

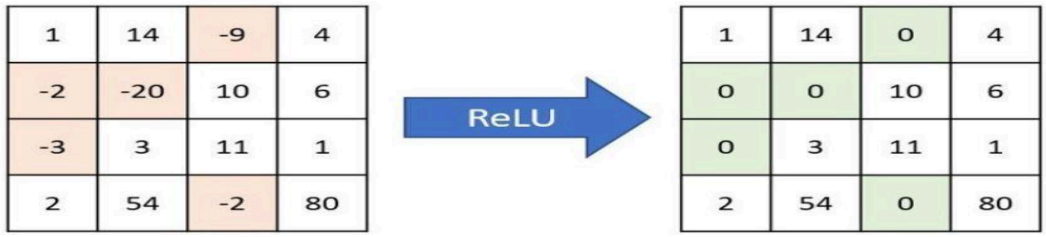


Fig3.12: CNN feature extraction with ReLu

After passing the outputs through ReLu functions they look like:

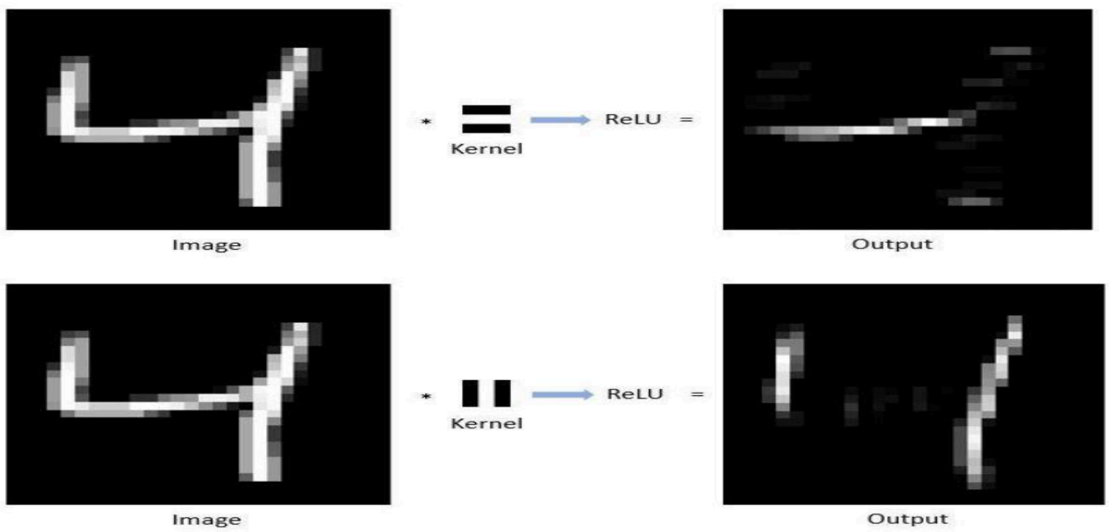


Fig3.13: Input image after filters with ReLu example

So for a single image by convolving it with multiple filters we can get multiple output images. For the handwritten digit here we applied a horizontal edge extractor and a vertical edge extractor and got two output images. We can apply several other filters to generate more such outputs images which are also referred as feature maps.

Step5: 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:

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.

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.

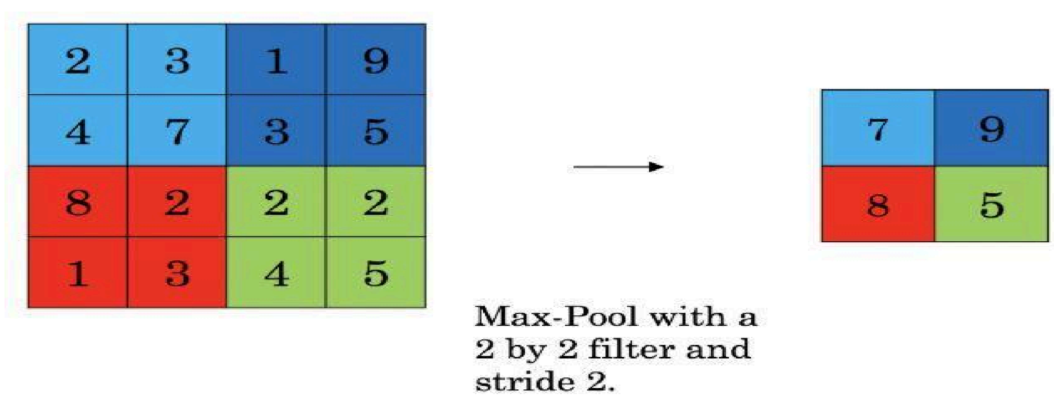


Fig3.14: Max Pooling example

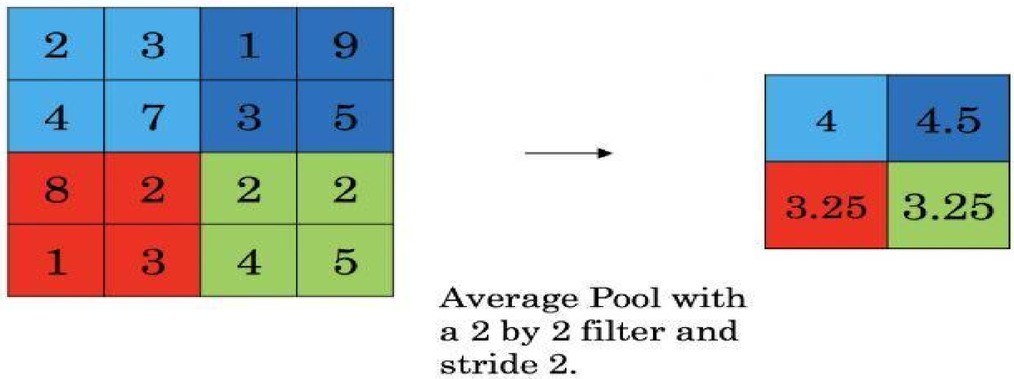


Fig3.15: Average Pooling example

Step6: 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:

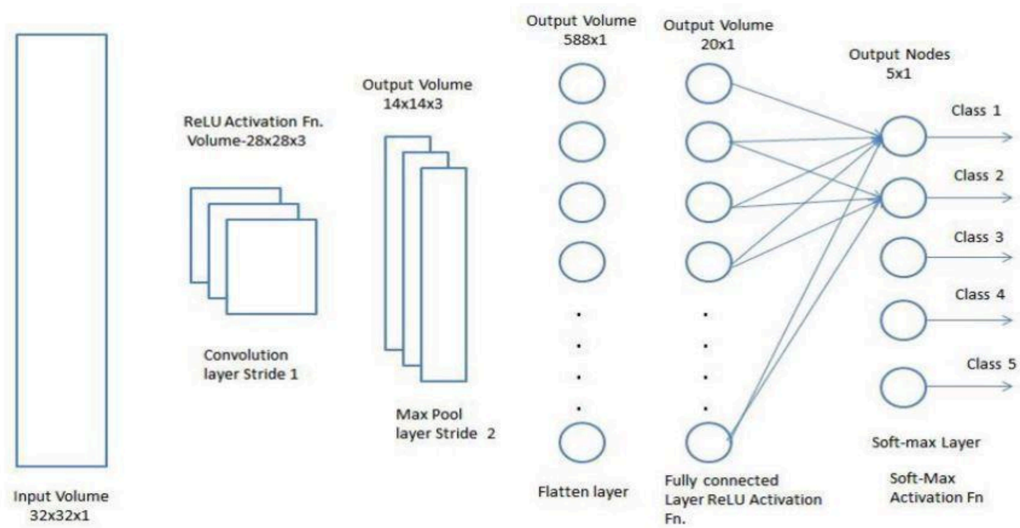


Fig3.16: Fully Connected model

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 Softmax Classification technique.

So now we have all the pieces required to build a CNN. Convolution, ReLU and Pooling. The output of max pooling is fed into the classifier which is usually a multi-layer perceptron layer. Usually in CNNs these layers are used more than once i.e. Convolution -> ReLU -> Max-Pool -> Convolution -> ReLU -> Max-Pool and so on.

CHAPTER 8

IMPLEMENTATION

8.1 Introduction

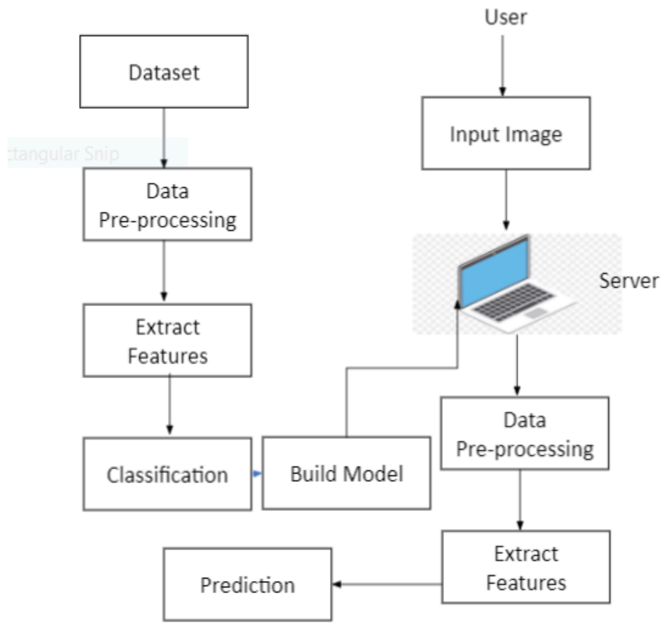
- The system setup plan creates general structure building plot. Programming chart fuses tending to the thing framework works in a shape that may be changed into in any event one envisions. The basic exhibited by the end client must be set in a systematical way. Chart is an imaginative framework; an exceptional plan is the best way to deal with sensible structure. The structure "Plan" is portrayed as "The technique of applying specific frameworks & rules with an authoritative objective of depicting a system or a system in sufficient explanation basic to permit its genuine confirmation".Different plan sections ȧre taken after to add to the framework. The plan detail portrays the fragments of the framework, the segments or sections of the structure & their appearance to end-clients.

8.2 Design Consideration

- The clarification behind the arrangement is to organize the strategy of the issue directed by the necessities report. This stage is the fundamental stage in moving from issue to the blueprint space. In light of everything, start with what is obliged; outline takes us to pursue how to fulfill those necessities. The plan of the framework is maybe the most basic section influencing the method of the thing & note commendably influences the later stages, especially testing & upkeep. Framework chart portrays all the enormous information structure, report blueprint, yield & authentic modules in the framework & their description is picked.

8.3 SYSTEM ARCHITECTURE

- The engineering setup strategy is worried about working up a crucial fundamental framework for a system. It incorporates perceiving the genuine pieces of the structure & exchanges between these sections. The starting arrangement method of perceiving these subsystems & working up a structure for subsystem control & correspondence is called development demonstrating layout & the yield of this diagram system is a depiction of the item basic arranging. The proposed engineering for this framework is given beneath. It shows the manner in which this framework is planned & brief working of the framework.



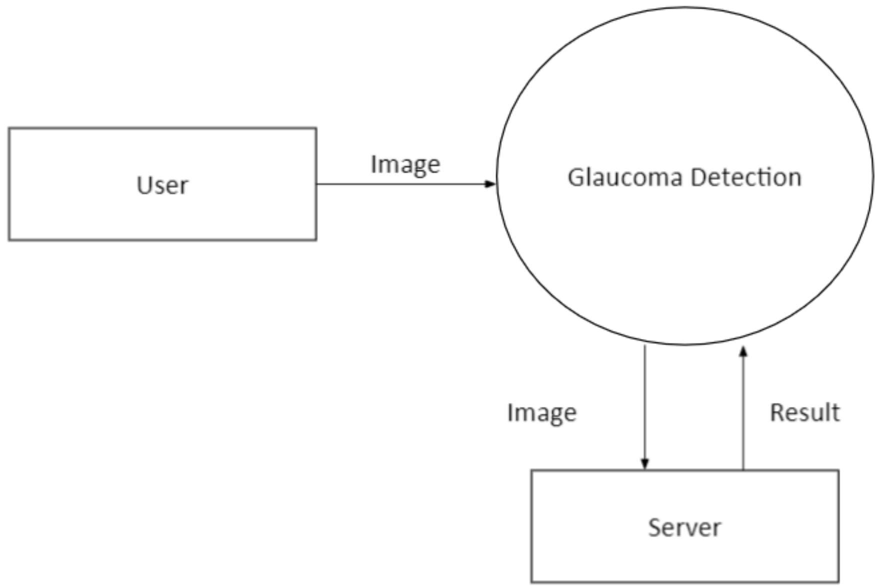
8.4 Data Flow Diagrams

- DFD graphically speaking to the capacities, or processes, which seize, command, stock, & disseminate information between a framework & its current circumstance & between parts of a framework. The visual portrayal makes it a decent specialized instrument among User & System fashioner. Structure of DFD permits beginning from a wide review & grow it to a progressive system of definite graphs. DFD has regularly been utilized because of the accompanying reasons
- Logical data brook of the framework
- Assurance of physical framework development necessities
- Simplicity of documentation
- Foundation of manual & computerized frameworks essentials

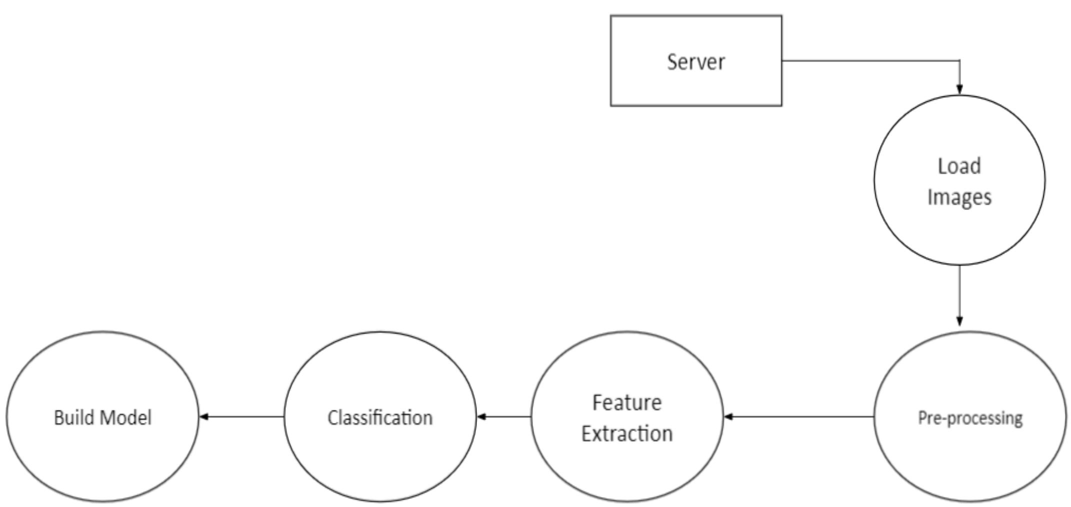
DFD Components

- DFD deals with Source, destination, storage & brook of information utilizing the accompanying arrangement of segments –
- **Entities** - An external entity is an individual, department, outside association, or other data framework that gives information to the framework.
- **Process** - any process that substitute the information, creating a end-result. It may execute calculations, or sort information dependent on rationale, or direct the information stream dependent on work edicts.
- **Data Storage** - records or stores that hold data for sometime in the future, e.g, an database table or an enlistment structure. Every information supply gets a straightforward name, e.g, " Orders."
- **Data Flow** - the course that information considered between the outer elements, processes & information supply

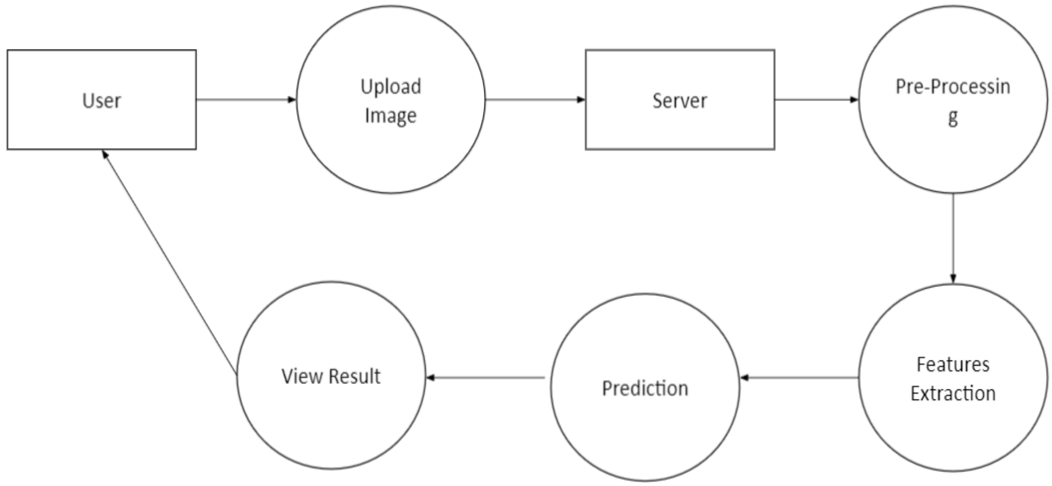
- **DFD-L**


User
Image
Glaucoma Detection
Image
Result
Server

- **DFD-L1:**


Server
Load Images
Build Model
Classification
Feature Extraction
Pre-processing

- **DFD-L2:**



8.5 Use Case Diagram

- The motivation behind this is to catch the dynamic part of a framework. Notwithstanding, this definition is too conventional to even consider describing the reason, as other 4 graphs likewise have a similar reason. We will discover some particular cause, which will identify it from other four charts

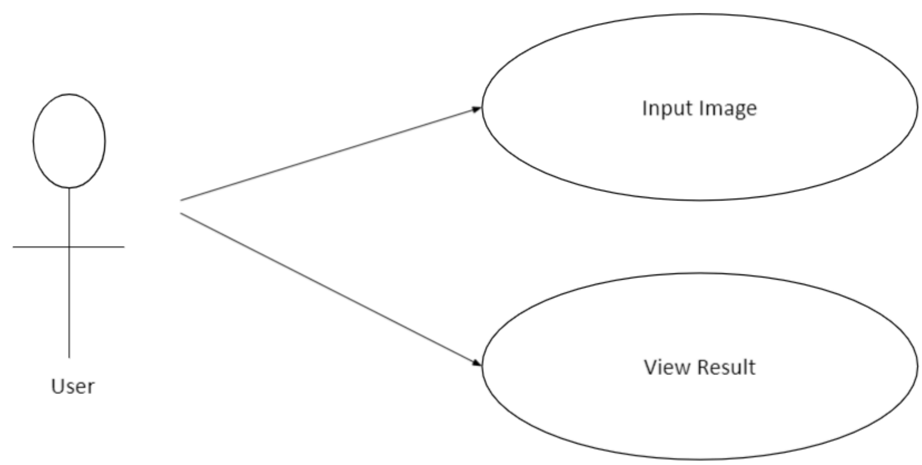
- Use case diagrams are utilized to assemble the essential framework including interior & outer impacts. These prerequisites are generally plan necessities. Henceforth, when a framework is breaking down to assemble its functionalities, use cases are readied & actors are distinguished.

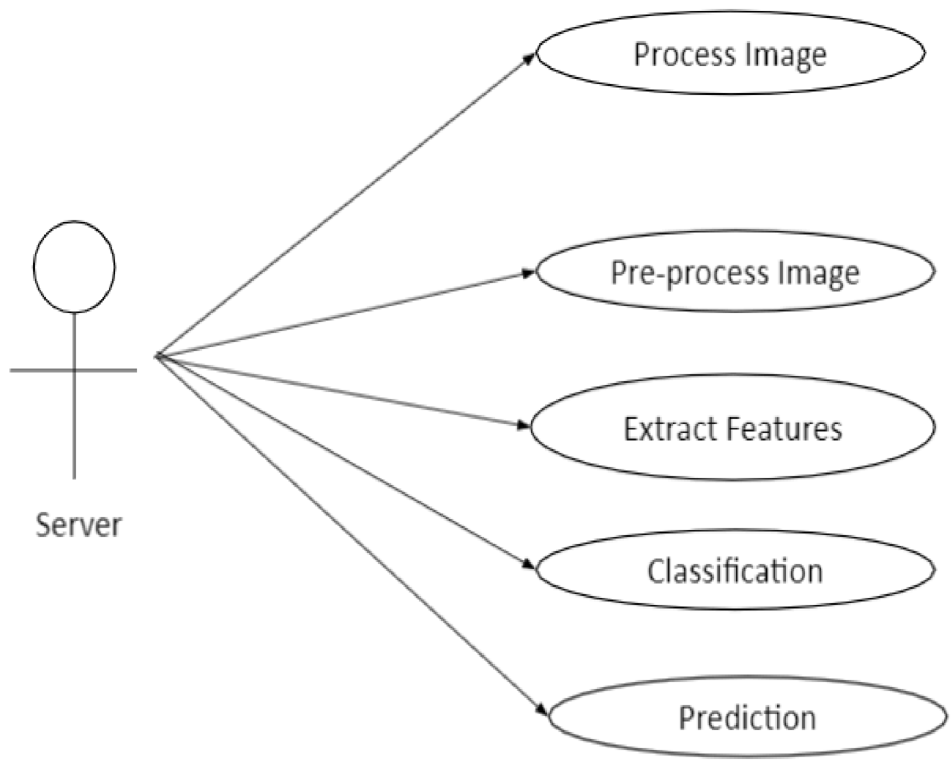
- At the point when the underlying undertaking is finished, use case diagrams are demonstrated to introduce the external view.

- To sum things up, the reasons for use case diagrams can be supposed to be as per the following

- Used to accumulate the prerequisites of a framework

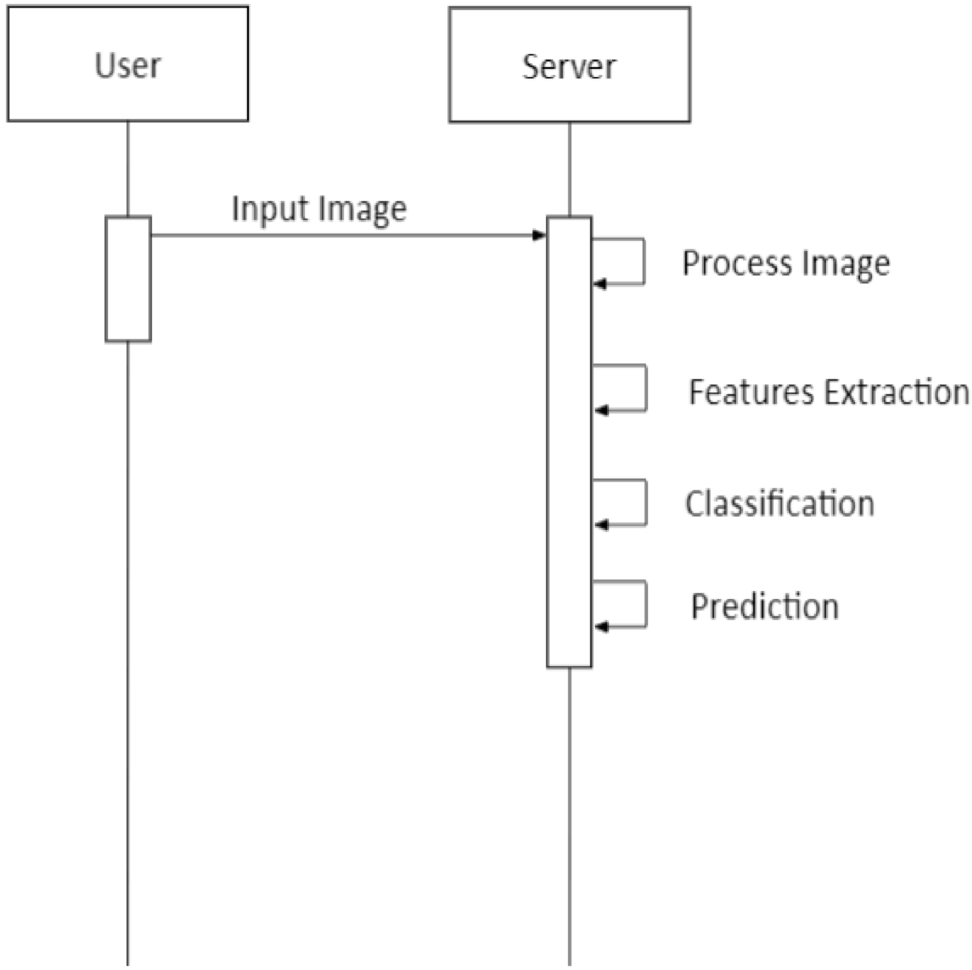
- Utilized to get an external outlook on a framework.
- Find the outside & inward factors impacting the framework.
- Show the cooperation among the requirements are actors.





8.6 Sequence Diagram

- A sequence diagram is a framework is a communication diagram that shows how process works with one another & in what request. It's a build of a message grouping diagram. A sequence diagram presents object connections orchestrated in time succession. It portrays the items & classes associated with the situation & grouping of messages trade between the objects expected to complete the usefulness of the situation. Sequence diagram are here & there called event diagrams or event scenarios.

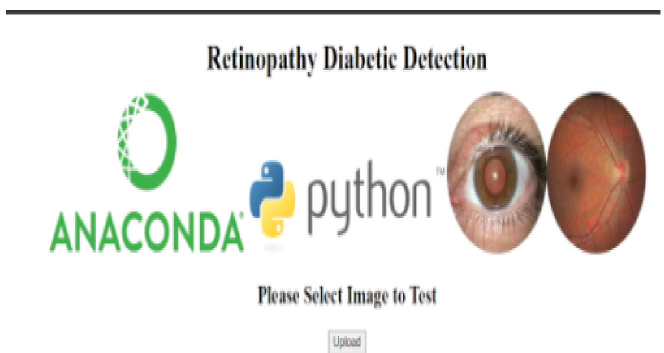

User
Server
Input Image
Process Image
Features Extraction
Classification
Prediction

CHAPTER 10

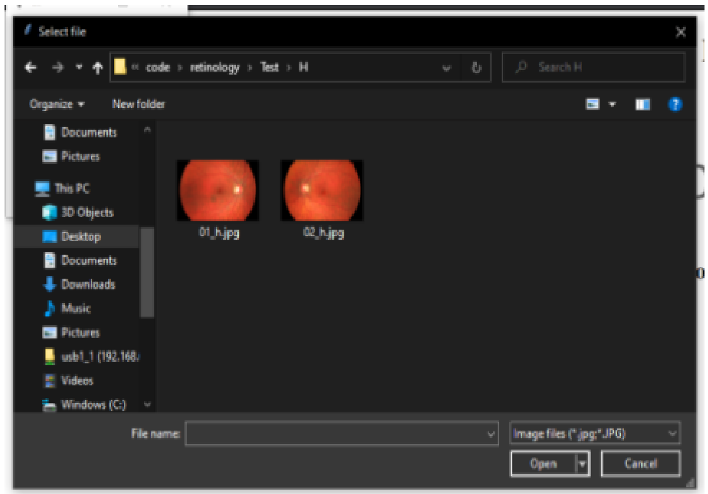
EXPERIMENTAL RESULTS & ANALYSIS

The Machine Learning based algorithms åre used to identify the peculiarities & five sorts of assaults åre recorded in the method.

Output-1:

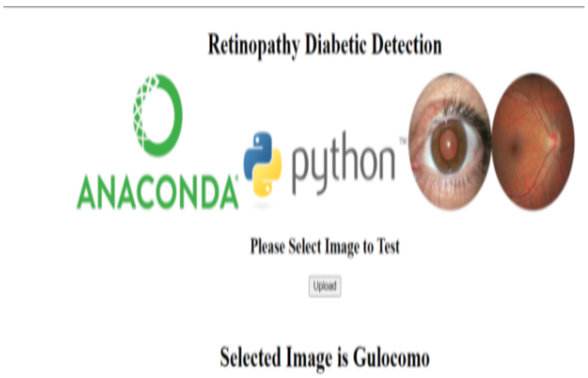


Output-2:



Output-3:

Output-3:



CHAPTER 10

DELIVERABLES

- Deep learning model: a fully trained and validated deep learning model capable of analyzing retinal images to predict cardiovascular disease risk with high accuracy. The model should be optimized for performance and tested across diverse datasets.
- Comprehensive research report: a detailed report documenting the project, including the problem statement, methodology, data preprocessing, model architecture, training process, evaluation metrics, and results. The report will also discuss limitations and future directions.

- Retinal image dataset: a curated and preprocessed dataset of retinal images used in training, validation, and testing, ensuring proper labeling and data augmentation techniques are applied to improve model performance.
- User interface (optional): a prototype or simple user interface that allows users (clinicians/researchers) to upload retinal images and receive cardiovascular risk predictions, making the model accessible for practical use.
- Publication or research paper: a ready-to-publish paper summarizing the findings, methodology, and contributions to the field of cardiovascular disease prediction using retinal images, suitable for submission to relevant conferences or journals.

REFERENCES

[1].W. Zhang, Z. Tian, F. Song, P. Xu, D. Shi, and M. He, "Enhancing stability in cardiovascular disease risk prediction: A deep learning approach leveraging retinalimages," Informatics in Medicine Unlocked, vol. 2023, 2023.

[2]. A. Kujalambal, K. Abhirami, and A. Aarthi, "Prediction of Heart Disease Using Retinal Images by Neural Network Algorithm," International Research Journal of Modernization in Engineering Technology and Science, vol. 5, no. 10, Oct. 2023.

[3]. J. Mellor, W. Jiang, A. Fleming, S. J. McGurnaghan, L. Blackbourn, C. Styles, A. J. Storkey, P. M. McKeigue, and H. M. Colhoun, "Can deep learning on retinal images augment known risk factors for cardiovascular disease prediction in diabetes? A

prospective cohort study," International Journal of Medical Informatics, vol. 2023, Apr. 2023.

[4]. W. Hu, F. S. L. Yii, R. Chen, X. Zhang, X. Shang, K. Kiburg, E. Woods, A. Vingrys, L. Zhang, Z. Zhu, and M. He, "A Systematic Review and Meta-Analysis of Applying Deep Learning in the Prediction of the Risk of Cardiovascular Diseases from Retinal Images," Translational Vision Science & Technology, vol. 2023, Jul. 2023.

[5]. Y. P. Prakash, B. Siri, and J. Jabez, "Identifying the Abnormalities in Retinal Images Towards the Prediction of Cardiovascular Disease Using Deep Learning," International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), 2024.

[6]. S. Shaikh, M. B. Mujawar, I. Wagh, D. Surve, and V. Zaveri, "Heart Disease Prediction Using Eye Retinal Images," in 6th International Conference on Advances in Science and Technology (ICAST), 2023.

[7]. J. K. Yi, T. H. Rim, S. Park, S. S. Kim, H. C. Kim, and C. J. Lee, "Cardiovascular disease risk assessment using a deep-learning-based retinal biomarker: a comparison with existing risk scores," European Heart Journal - Digital Health, vol. 4, no. 3, pp. 236-244, 2023. Available: https://doi.org/10.1093/ehjdh/ztad023.

[8]. R. G. Barriada and D. Masip, "An Overview of Deep-Learning-Based Methods for Cardiovascular Risk Assessment with Retinal Images," Diagnostics, vol. 13, no. 1, p. 68, Jan. 2023. Available: https://doi.org/10.3390/diagnostics13010068.

[9]. L. Y. Li, A. A. Isaksen, B. Lebiecka-Johansen, K. Funck, V. Thambawita, S. Byberg, T. H. Andersen, O. Norgaard, and A. Hulman, "Prediction of Cardiovascular Markers and Diseases Using Retinal Fundus Images and Deep Learning: A Systematic Scoping Review," European Heart Journal - Digital Health, vol. 4, no. 3, pp. 236-244, 2024.

Available: https://doi.org/10.1093/ehjdh/ztae068.

[10] Alagona Jr P, Ahmad TA. Cardiovascular disease risk assessment and prevention: current guidelines and limitations. Med Clin 2015;99(4):711–31.

[11].He, Along, et al. "CABNet: category attention block for imbalanced diabetic retinopathy grading." IEEE Transactions on Medical Imaging 40.1 (2020): 143-153.

[12] Kaushik, Harshit, et al. "Diabetic retinopathy diagnosis from fundus images using stacked generalization of deep models." IEEE Access 9 (2021): 108276-108292.

[13] Group Sw, Collaboration ESCCr. SCORE2 risk prediction algorithms: new models to estimate 10-year risk of cardiovascular disease in Europe. Eur Heart J. 2021;
42(25): 2439–2454

[14] Huang L, Aris IM, Teo LLY, et al. Exploring associations between cardiac structure and retinal vascular geometry. J Am Heart Assoc. 2020; 9(7): e014654.