A Project Report on

"FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

By

P. LAKSHMI PRIYA	**20X51A3238**
K. AKHILA	**20X51A3227**
M.V..RUCHITHA	**20X51A3234**
S. AYESHA	**20X51A3246**
S. KARIMUN	**20X51A3249**

Under the Esteemed Guidance of

Mr. P. SUBBA RAO, M. Tech,

Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

SANTHIRAM ENGINEERING COLLEGE::NANDYAL

(AUTONOMOUS)

(Approved by AICTE, New Delhi: Permanently Affiliated to J.N.T University, Anantapur, A.P)

Accredited by NAAC (Grade-A), Accredited by NBA (ECE&CSE)

An ISO 9001:2015 Certified Institution, 2(f) and 12(b) recognition by UGC Act, 1956

Web: www.srecnandyal.edu.in

YEAR: 2020-2024

SANTHIRAM ENGINEERING COLLEGE::NANDYAL

(AUTONOMOUS)

(Approved by AICTE, New Delhi: Permanently Affiliated to J.N.T. University, Anantapur, A.P)

Accredited by NAAC (Grade-A), Accredited by NBA (ECE&CSE)

An ISO 9001:2015 Certified Institution, 2(f) and 12(b) recognition by UGC Act, 1956

Web: **www.srecnandyal.edu.in**

CERTIFICATE

This is to certify that the dissertation entitled "FACIAL PARALYSIS SEVERITY DETECTION USING TRANSGER LEARNING AND ARTIFICAL INTELLIGENCE" is being submitted by P.LAKSHMI PRIYA (20X51A3238), K.AKHILA (20X51A3227), M.V.RUCHITHA (20X51A3234), S.AYESHA (20X51A3246), S.KARIMUN (20X51A3249) for partial fulfillment of the award of the Degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE) in the SANTHIRAM ENGINEERING COLLEGE, Nandyal (Permanently Affiliated to J.N.T. University, Anantapur) is a record of bonafied work carried out by them under our guidance and supervision. The results embodied in this project have not been submitted to any other university for the award of any Degree.

Signature of the Head of the Department:
Smt. N. RAMADEVI M. Tech, (Ph.D)
Associate Professor & HOD

Signature of the Project Guide:
Mr. P. SUBBA RAO M.Tech,
Assistant Professor

Signature of the External Examiner:

Date of Examination:

ACKNOWLEDGEMENT

This will certainly not be complete without acknowledgements paid to all these who have helped us in doing our project work.

We manifest our heartier thankfulness pertaining to our contentment over **Mr. Subba Rao garu,** as project guide with whose adroit concomitance the excellencehas been exemplified in bringing out this project work with artistry.

It is a great pleasure to acknowledge our profound sense of gratitude to our Head of the Department **Smt. N. Ramadevi garu,** and Coordinator **Mr.P.Subba Rao garu,** for his valuable and inspiring guidance, comments and suggestions and encouragement towards the course of this project.

Involuntarily, we are precious to divulge our sincere gratefulness to our principal, **Dr.M.V. Subramanyam garu,** who has been observed posing valance in abundance forwards our individuality to acknowledge our project work tendentiously.

At the outset we thank our Honorable chairman **Dr.M. Santhiramudu garu,** correspondent, Santhiram Engineering College for providing us with good facilities and his moral support throughout the course.

We also express the overall exhilaration and gratitude to all the Teaching and Non- Teaching staff of CSE (DS) Dept., who admitted our project work and accentuated our attitude.

We also extend our heartfelt and sincere gratitude to our beloved parents for their tremendous motivation and moral support.

P. LAKSHMI PRIYA

K. AKHILA

M.V. RUCHITHA

S. AYESHA

S. KARIMUN

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This project introduces a novel deep learning-based approach to detecting facial paralysis, a condition that affects millions worldwide but often goes undiagnosed in its early stages. Leveraging the InceptionV3 algorithm for its robustness and efficiency in real-time object detection, we have developed a model that accurately identifies signs of facial paralysis from both static images and live video feeds. The dataset, sourced from Kaggle's official website, comprises 3,000 images categorized into two classes, enabling the model to learn diverse representations of the condition. Our model achieved an exceptional accuracy of 99.3% during testing, underscoring its potential as a reliable diagnostic tool.

The model was trained and validated using Google Collab, benefiting from its powerful computational resources and collaborative environment. Subsequent deployment in a web application provides an accessible platform for users to register and obtain authorization from an administrator. Upon login, users can upload images or access a live camera feed to detect facial paralysis, facilitating early diagnosis and intervention. The web application also features an administrative module responsible for user authentication, model training, and performance analysis, ensuring the system's integrity and effectiveness.

This project not only showcases the application of deep learning in addressing critical health issues but also sets a precedent for the development of accessible diagnostic tools that can be deployed at scale, offering significant implications for telemedicine and remote healthcare services.

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

Facial paralysis is a medical condition characterized by the loss of voluntary muscle movement in one or both sides of the face, which can significantly affect a person's quality of life through impairments in facial expressions, eating, speaking, and even leading to psychological distress. The causes of facial paralysis are varied, including Bell's palsy, stroke, Lyme disease, and trauma, making its diagnosis complex and necessitating specialized medical expertise. Traditional diagnostic processes are often time-consuming, costly, and not easily accessible to everyone, particularly in less urbanized or resource-limited regions.

In response to these challenges, this project introduces an innovative solution leveraging the advancements in deep learning and computer vision. By adopting the InceptionV3 algorithm, renowned for its speed and accuracy in object detection tasks, we have developed a sophisticated model capable of detecting facial paralysis from both static images and live video streams. This approach not only democratizes access to diagnostic tools but also significantly enhances the efficiency and accuracy of early-stage detection.

The development and deployment of this model within a web-based application further extend its accessibility and usability. Through this platform, users can easily register, upload images, or use a live camera feed to receive immediate diagnostic feedback on the presence of facial paralysis signs. An administrative module ensures the system's integrity by managing user access, overseeing model performance, and facilitating continuous improvement through retraining processes.

This project stands at the intersection of technology and healthcare, aiming to bridge the gap between advanced medical diagnostics and patient accessibility. By providing a scalable, accurate, and user-friendly tool for the early detection of facial paralysis, it paves the way for prompt intervention and treatment, potentially mitigating long-term impacts on patients' lives.

1.2 Objective of the Project

The objective of this project is to develop a deep learning model using the InceptionV3 algorithm for the accurate detection of facial paralysis from static images and live video feeds. It aims to offer an accessible, efficient diagnostic tool through a web application, enabling early detection and intervention for users. The project also includes an administrative module for user management, model training, and performance analysis, enhancing healthcare accessibility and quality. By adopting the InceptionV3 algorithm, renowned for its speed and accuracy in object detection tasks, we have developed a sophisticated model capable of detecting facial paralysis from both static images and live video streams.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Automatic facial paralysis assessment via computational image analysis

Facial paralysis (FP) is a loss of facial movement due to nerve damage. Most existing diagnosis systems of FP are subjective, e.g., the House–Brackmann (HB) grading system, which highly depends on the skilled clinicians and lacks an automatic quantitative assessment. In this paper, we propose an efficient yet objective facial paralysis assessment approach via automatic computational image analysis. First, the facial blood flow of FP patients is measured by the technique of laser speckle contrast imaging to generate both RGB color images and blood flow images. Second, with an improved segmentation approach, the patient's face is divided into concerned regions to extract facial blood flow distribution characteristics. Finally, three HB score classifiers are employed to quantify the severity of FP patients. The proposed method has been validated on 80 FP patients, and quantitative results demonstrate that our method, achieving an accuracy of 90.14%, outperforms the state-of-the-art systems. Experimental evaluations also show that the proposed approach could yield objective and quantitative FP diagnosis results, which agree with those obtained by an experienced clinician.

Objective grading facial paralysis severity using a dynamic 3D stereo photogrammetry imaging system

Facial paralysis is a loss of facial movement due to nerve damage. It is essential for clinicians to diagnose the severity of facial paralysis to treat patients, assess progress and evaluate outcomes. Subjective assessments are common in clinical practices but have their limitations regarding the intra-observer and inter-observer reproducibility. We utilized dynamic 3D stereo photogrammetry technology for the objective grading of facial paralysis by measuring regional facial asymmetries. The correlations between the measured asymmetries and scores of a modified Sunnybrook facial paralysis grading were evaluated to identify the region of interests of objective measurements closely related to the subjective grades. Categorical classifiers were trained to quantify the severity of facial paralysis. Preliminary results showed that the objective asymmetry measurements were highly correlated to the subjective assessments of facial paralysis except the eye region. Machine learning approaches showed the potential of improving the accuracy of severity assessments.

Toward an automatic system for computer aided assessment in facial palsy

Quantitative assessment of facial function is challenging, and subjective grading scales such as House-Brackmann, Sunnybrook, and eFACE have well-recognized limitations. Machine learning (ML) approaches to facial landmark localization carry great clinical potential as they enable high-throughput automated quantification of relevant facial metrics from photographs and videos. However, the translation from research settings to clinical application still requires important improvements.

To develop a novel ML algorithm for fast and accurate localization of facial landmarks in photographs of facial palsy patients and utilize this technology as part of an automated computer-aided diagnosis system.

Portrait photographs of 8 expressions obtained from 200 facial palsy patients and 10 healthy participants were manually annotated by localizing 68 facial landmarks in each photograph and by 3 trained clinicians using a custom graphical user interface. A novel ML model for automated facial landmark localization was trained using this disease-specific database. Algorithm accuracy was compared with manual markings and the output of a model trained using a larger database consisting only of healthy subjects.

Root mean square error normalized by the interocular distance (NRMSE) of facial landmark localization between prediction of ML algorithm and manually localized landmarks.Publicly available algorithms for facial landmark localization provide poor localization accuracy when applied to photographs of patients compared with photographs of healthy controls (NRMSE, 8.56 ± 2.16 vs. 7.09 ± 2.34, $p \ll 0.01$). We found significant improvement in facial landmark localization accuracy for the facial palsy patient population when using a model trained with a relatively small number photographs (1440) of patients compared with a model trained using several thousand more images of healthy faces (NRMSE, 6.03 ± 2.43 vs. 8.56 ± 2.16, $p \ll 0.01$).

Retraining a computer vision facial landmark detection model with fewer than 1600 annotated images of patients significantly improved landmark detection performance in frontal view photographs of this population. The new annotated database and facial landmark localization model represent the first steps toward an automatic system for computer-aided assessment in facial palsy.

The spectrum of facial palsy: The MEEI facial palsy photo and video standard set

Facial palsy causes variable facial disfigurement ranging from subtle asymmetry to crippling deformity. There is no existing standard database to serve as a resource for facial palsy education and research. We present a standardized set of facial photographs and videos representing the entire spectrum of flaccid and nonflaccid (aberrantly regenerated) facial palsy. To demonstrate the utility of the dataset, we describe the relationship between level of facial function and perceived emotion expression as determined by an automated emotion detection, machine learning-based algorithm.

Facial imaging and landmark detection technique for objective assessment of unilateral peripheral facial paralysis

In this paper, we propose a hypothesis that the facial landmark detection methods constructed by a private UPFP facial dataset can perform better than the model on a healthy facial dataset in the task of UPFP facial landmark detection. For proving this hypothesis, a customized UPFP facial dataset with 68 facial landmark annotations was built. A state-of-the-art facial landmark detection method was employed on the three evaluation datasets to exploit and prove the hypothesis. The mean error of validation dataset is 3.15, 56% lower than 7.42 that of the healthy dataset, which proves the hypothesis is true.

Automatic assessment of facial paralysis based on facial landmarks

Unilateral peripheral facial paralysis is the most common case of facial paralysis. It affects only one side of the face, which will cause facial asymmetry. Clinically, unilateral peripheral facial paralysis is often classified by clinicians according to evaluation scales, based on patients’ condition of facial symmetry. A prevalent scale is House-Brackmann grading system (HBGS).

However, assessment results from scales are often with great subjectivity, and will bring high interobserver and interobserver variability. Therefore, this manuscript proposed an objective method to provide assessment results by using facial videos and applying machine learning models. This grading method is based on HBGS, but it is automatically implemented with high objectivity. Images with facial expressions will be extracted from the videos to be analyzed by a machine learning model. Facial landmarks will be acquired from the images by using a 68-points model provided by dlib. Then index and coordinate information of the landmarks will be used to calculate the values of features pre-designed to train the model and predict the result of new patients. Due to the difficulty of collecting facial paralysis samples, the data size is limited. Random Forest (RF) and support vector machine (SVM) were compared as classifiers. This method was applied on a data set of 33 subjects. The highest overall accuracy rate reached 88.9%, confirming the effectiveness of this method.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system for diagnosing facial paralysis predominantly relies on clinical assessments conducted by healthcare professionals, which may include physical examinations, medical history evaluations, and specialized tests such as electromyography (EMG) to measure muscle response to nerve stimulation. While effective, these methods require access to medical facilities and specialists, which can be challenging in remote or underserved areas. Moreover, the diagnostic process can be time-consuming and costly, with a potential delay in treatment initiation. Telemedicine platforms have begun to address these accessibility issues, offering remote consultations; however, they still depend on the availability of specialists and may not provide the immediate, automated feedback that advanced technology solutions can offer. Consequently, there exists a significant need for a more accessible, efficient, and cost-effective solution to detect facial paralysis early and accurately, leveraging the advancements in artificial intelligence and machine learning.

LIMITATIONS OF EXISTING SYSTEM

1. **Specialist Dependency**: Diagnosis often requires consultation with specialists who are trained to recognize and assess the nuances of facial paralysis. This dependency can create bottlenecks in healthcare systems with limited specialist availability, leading to delayed diagnoses and treatment initiation.

2. **Geographical Barriers**: Access to healthcare facilities and specialists is uneven across regions, with rural and underserved areas facing significant disadvantages. Patients in these locations may experience delays in receiving timely care or may incur substantial costs and inconvenience traveling to distant medical centers.

3. **Diagnostic Delays**: Traditional diagnostic methods, including physical examinations and electromyography (EMG), can be time-consuming. The process from initial

consultation to definitive diagnosis may involve multiple appointments and tests, prolonging the period before treatment can begin.

4. **Cost Factors**: The cost of specialist consultations, diagnostic tests, and travel to medical facilities can be prohibitive for some patients, potentially deterring them from seeking timely medical advice.
5. **Lack of Early Detection**: Early stages of facial paralysis might be subtle and overlooked by patients until more evident symptoms emerge. The existing system's reliance on subjective patient reporting and specialist availability can lead to missed opportunities for early intervention.
6. **Scalability Issues**: The ability to scale specialist consultations to meet high demand or during health crises (e.g., pandemics) is limited, which can exacerbate delays and accessibility issues.

3.2 PROPOSED SYSTEM

The proposed system leverages a deep learning approach, utilizing the InceptionV3 algorithm, to accurately detect facial paralysis from both static images and live video feeds. By harnessing a dataset of 3,000 images from Kaggle, enhanced through data augmentation techniques, the system achieves remarkable detection accuracy, demonstrated by a 99.3% success rate in tests. Integrated into a user-friendly web application, it offers functionalities for user registration, image upload, and live detection, managed through an admin module for user authentication and model oversight. This system not only bridges the gap in accessible diagnostic tools for facial paralysis but also sets a precedent for applying AI in telehealth solutions, promising widespread impact on early detection and patient care.

ADVANTAGES OF PROPOSED SYSTEM

1. **Enhanced Accessibility**: By deploying the diagnostic tool in a web application, the system becomes accessible to users globally, removing geographical and logistical barriers to specialized healthcare services. This means that individuals in remote or underserved areas can also benefit from early detection capabilities.

2. **Real-Time Detection**: The use of the INCEPTIONV3 algorithm enables the system to process images and live video feeds in real-time, providing immediate feedback to users. This immediacy is crucial for early detection and intervention, which can significantly improve treatment outcomes for facial paralysis.

3. **High Accuracy**: With a detection accuracy of 99.3%, the system demonstrates exceptional performance in identifying signs of facial paralysis. This high level of accuracy is pivotal in ensuring reliable diagnostics, reducing the likelihood of false positives or negatives, which are common challenges in medical diagnoses.

4. **Reduced Dependency on Specialists**: While not a replacement for professional medical advice, the system serves as a valuable preliminary screening tool. It can potentially reduce the workload on specialists and expedite the diagnostic process for patients, enabling healthcare systems to allocate resources more efficiently.

5. **Cost-Effectiveness**: By automating the initial screening process, the proposed system offers a more cost-effective solution compared to traditional diagnostic methods. It minimizes the need for in-person consultations and specialized tests, which can be expensive and time-consuming, making early detection more financially accessible to a wider population.

3.3 MODULES DESCRIPTION

1. Data Management Module

- **Functionality**: Manages the collection, augmentation, and preprocessing of the dataset. This includes sourcing images from Kaggle, applying data augmentation techniques to increase the diversity of the training set, and preprocessing images to fit the input requirements of the INCEPTIONV3 model.
- **Components**: Dataset collection, Data augmentation, Preprocessing.

2. Model Training and Validation Module

- **Functionality**: Handles the core development of the deep learning model using the INCEPTIONV3 algorithm. It oversees the training process, including the selection of hyperparameters, the division of data into training and validation sets, and the evaluation of model performance to prevent overfitting.

- **Components**: Model architecture setup, Training loop, Validation and performance evaluation.

3. Real-time Detection Module

- **Functionality**: Facilitates the real-time detection of facial paralysis from both static images and live video feeds. This module leverages the trained INCEPTIONV3 model to analyze uploaded images or camera streams, identifying signs of facial paralysis with high accuracy.
- **Components**: Image upload, Live video feed processing, Detection algorithm.

4. User Interface (UI) Module

- **Functionality**: Provides the front-end interface for user interaction with the web application. It enables users to register, login, upload images, access the live camera feed, and receive diagnostic feedback. The UI is designed to be intuitive and user-friendly, ensuring ease of use for individuals with varying levels of technical proficiency.
- **Components**: Registration and login system, Image upload interface, Live camera access, Feedback display.

5. Admin and Security Module

- **Functionality**: Offers tools for administrative tasks, including user management, model management, and security measures. It ensures that only authorized users can access the system, provides the admin with the ability to authorize new users, and oversees the training and testing of the model for continual improvement.
- **Components**: User authentication and authorization, Model training and testing oversight, Security protocols.

3.4 PROCESS MODEL USED WITH JUSTIFICATION

CONVOLUTIONAL NEURAL NETWORKS

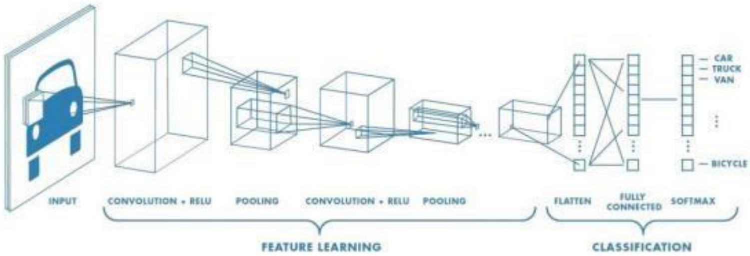


Fig 3.4.1: Convolutional neural network

The field of Artificial Intelligence (AI) is rapidly advancing, significantly narrowing the gap between human and machine capabilities. Enthusiasts and researchers are delving into various facets of AI to create remarkable innovations. A prime example of such a domain is Computer Vision. The objective in this area is to empower machines with the ability to interpret the world in a manner akin to human vision, and to utilize this understanding in a range of applications. These include tasks like Image and Video Recognition, Image Analysis and Classification, Media Recreation, and

Recommendation Systems, as well as extending into areas like Natural Language Processing. The progress in Computer Vision, particularly through the application of Deep Learning, has been substantial. This progress is largely attributed to the refinement of a key algorithm: the Convolutional Neural Network (CNN), which has evolved and been honed over time.

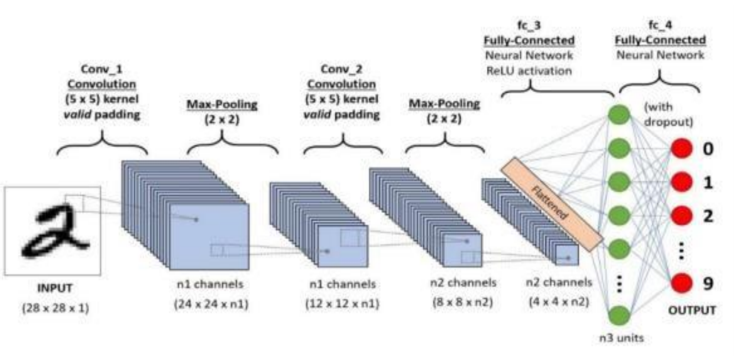


Fig 3.4.2: layers of ConNet

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that processes input images by assigning significance to various aspects within them, enabling it to distinguish between different objects. Compared to other classification algorithms, CNNs require significantly less preprocessing. Traditional methods involve manually creating filters, but with sufficient training, CNNs are capable of learning these filters autonomously. The design of a CNN mirrors the neural connectivity patterns found in the human brain, particularly inspired by the structure of the visual cortex. In this system, individual neurons are activated by specific stimuli within a limited area of the visual field, known as the receptive field. These receptive fields overlap to comprehensively cover the entire visual spectrum.

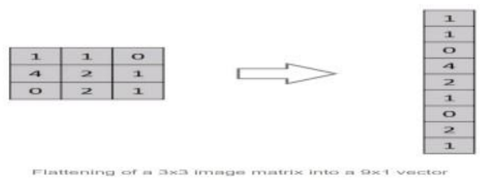


Fig 3.4.3: flattening of 3x3 image matrix

An image is essentially a matrix composed of pixel values. One might wonder why not simply flatten this matrix and input it into a multi-layer perceptron for classification. The reason lies in the unique capabilities of a Convolutional Neural Network (ConvNet). A ConvNet excels in capturing the spatial and temporal dependencies in an image by applying appropriate filters. Its architecture is more adept at conforming to the image dataset, primarily due to a decrease in the number of parameters and the recyclability of weights. In essence, this means that the network can be more effectively trained to comprehend the complexities inherent in images.

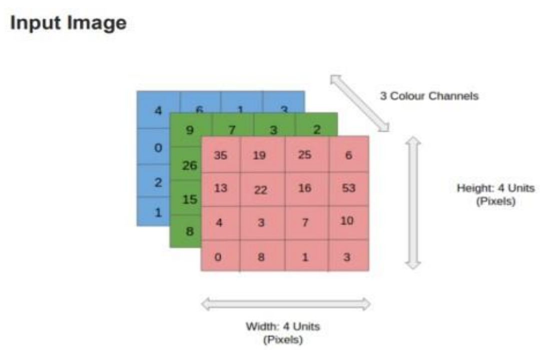


Fig 3.4.4: RGB image separated by its colors

In the given illustration, an RGB image is depicted, segmented into its three primary color components: red, green, and blue. Images can exist in various color spaces, such as grayscale and RGB. Imagine the computational load when dealing with high resolution images, such as those in 8K. The function of a Convolutional Neural Network (ConvNet) in this context is to simplify the images into a more manageable form for processing, while preserving essential features crucial for accurate predictions.

This aspect is particularly vital in designing an architecture that is not only efficient in feature learning but also capable of scaling to handle extensive datasets.

POOLING LAYER

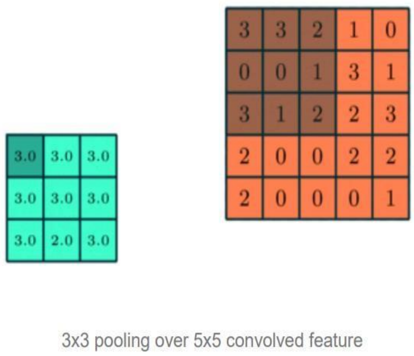


Fig 3.4.5: pooling layers

Much like the Convolutional Layer, the role of the Pooling layer in a Convolutional Neural Network is to reduce the spatial dimensions of the Convolved Feature. This reduction is key in lessening the computational burden by decreasing the data's dimensionality. Additionally, Pooling is instrumental in extracting pivotal features that are invariant to rotation and position, aiding in the efficient training of the model.

Pooling comes in two main forms: Max Pooling and Average Pooling. Max Pooling operates by selecting the maximum value from the image area covered by the Kernel, effectively acting as a noise suppressant by eliminating noisy activations and aiding in denoising as well as reducing dimensionality. Average Pooling, in contrast, computes the average value of all elements in the Kernel's coverage area and primarily focuses on dimensionality reduction as its method of noise suppression. Consequently, Max Pooling is often considered more effective than Average Pooling due to its dual role in noise suppression and dimensionality reduction.

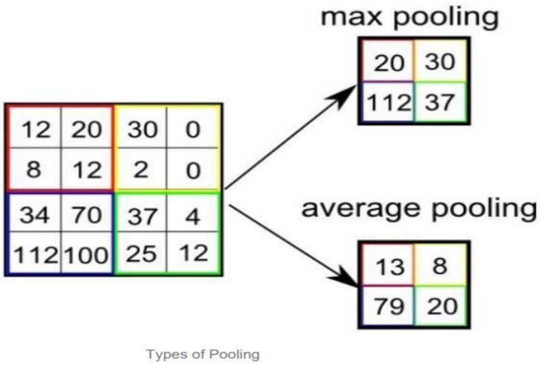


Fig 3.4.6: types of pooling layers

The Convolutional Layer and the Pooling Layer collectively constitute the i-th layer in a Convolutional Neural Network's architecture. To capture more nuanced, low-level details in complex images, the network may incorporate additional layers of this kind. However, this enhancement comes with the trade-off of increased computational demands. Once the image data has passed through these layers, the model gains a comprehensive understanding of the image features. Subsequently, the final output is flattened and then input into a standard neural network, which performs the task of classifying the data.

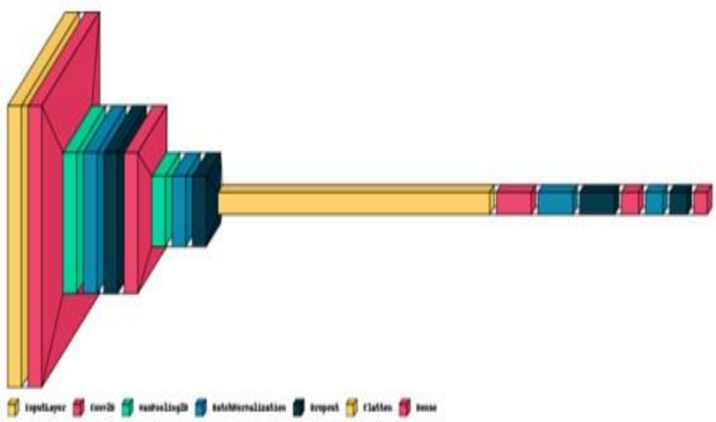


Fig 3.4.7: ConvNet layers

3.5 SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM STUDY

1. TECHNICAL FEASIBILITY

2. OPERATIONAL FEASIBILITY

3. ECONOMIC FEASIBILITY

3.5.1 OVERALL DESCRIPTION

A feasibility study assesses the operational, technical and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worthy of proceeding to the analysis phase. From the systems analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

The feasibility study is a management-oriented activity. The objective of a feasibility study is to find out if an information system project can be done and to suggest possible alternative solutions.

Projects are initiated for two broad reasons:

1. Problems that lend themselves to systems solutions

2. Opportunities for improving through:

    (a) upgrading systems

    (b) altering systems

    (c) installing new systems

A feasibility study should provide management with enough information to decide:

- Whether the project can be done
- Whether the final product will benefit its intended users and organization
- What are the alternatives among which a solution will be chosen

TECHNICAL FEASIBILITY

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does the technology exist at all?
- Is it available within given resource constraints?
- Is it a practical proposition?
- Manpower- programmers, testers & debuggers
- Software and hardware
- Are the current technical resources sufficient for the new system?
- Can they be upgraded to provide to provide the level of technology necessary for the new system?
- Do we possess the necessary technical expertise, and is the schedule reasonable?
- Can the technology be easily applied to current problems?
- Does the technology have the capacity to handle the solution?
- Do we currently possess the necessary technology?

OPERATIONAL FEASIBILITY

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

The essential questions that help in testing the operational feasibility of a system include the following:

- Does current mode of operation provide adequate throughput and response time?
- Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?
- Does current mode of operation provide cost-effective information services to the business?
- Could there be a reduction in cost and or an increase in benefits?
- Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information?
- Does current mode of operation make maximum use of available resources, including people, time, and flow of forms?
- Does current mode of operation provide reliable services
- Are the services flexible and expandable?
- Are the current work practices and procedures adequate to support the new system?
- If the system is developed, will it be used?

- Manpower problems
- Labour objections
- Manager resistance
- Organizational conflicts and policies
- Are the users not happy with current business practices?
- Will it reduce the time (operation) considerably?
- Have the users been involved in the planning and development of the project?
- Will the proposed system really benefit the organization?
- Does the overall response increase?
- Will accessibility of information be lost?
- Will the system affect the customers in considerable way?
- Legal aspects
- How do the end-users feel about their role in the new system?
- What end-users or managers may resist or not use the system?
- How will the working environment of the end-user change?
- Can or will end-users and management adapt to the change?

ECONOMIC FEASIBILITY

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs?
- The cost of doing full system study

- The cost of business employee time
- Estimated cost of hardware
- Estimated cost of software/software development

The concerned business must be able to see the value of the investment it is pondering before committing to an entire system study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, then the system is not economically feasible, and the project should not proceed any further. If the expected benefits equal or exceed costs, the system can be judged to be economically feasible. Economic analysis is used for evaluating the effectiveness of the proposed system.

The economic feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe. The required timeframe would need to be set by the organization.

3.5.2 EXTERNAL INTERFACE REQUIREMENTS:

User Interface:

The user interface of this system is a user-friendly python Graphical User Interface.

Hardware Interfaces:

The interaction between the user and the console is achieved through python capabilities.

Software Interfaces:

The required software is python.

Operating Environment:

Windows XP.

3.5.3 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

MINIMUM (Required for Execution)		MY SYSTEM (Development)
System	Pentium IV 2.2 GHz	i3 Processor 5^{th} Gen
Hard Disk	20 Gb	500 Gb
Ram	1 Gb	4 Gb

SOFTWARE REQUIREMENTS

Operating System	Windows 10/11
Development Software	Python 3.10
Programming Language	Python
Integrated Development Environment (IDE)	Visual Studio Code
Front End Technologies	HTML5, CSS3, Java Script
Back End Technologies or Framework	Django
Database Language	SQL
Database (RDBMS)	MySQL
Database Software	WAMP or XAMPP Server
Web Server or Deployment Server	Django Application Development Server

CHAPTER 4

SYSTEM DESIGN

UML DIAGRAMS:

CLASS DIAGRAM:

The fundamental component of object-oriented modeling is the class diagram. It is used for both technical modeling—which converts the models into computer code—and general conceptual modeling of the application's systematics. Data modeling is another application for class diagrams. In a class diagram, the classes stand in for the primary objects, application interactions, and classes that need to be programmed. Classes are shown in the diagram as boxes with three sections each:

- The left side box holds the admin of the class.
- The right side box holds the user class.
- In the boxes contains the attributes of the class
- Admin box has the methods or operations the class take or undertake.

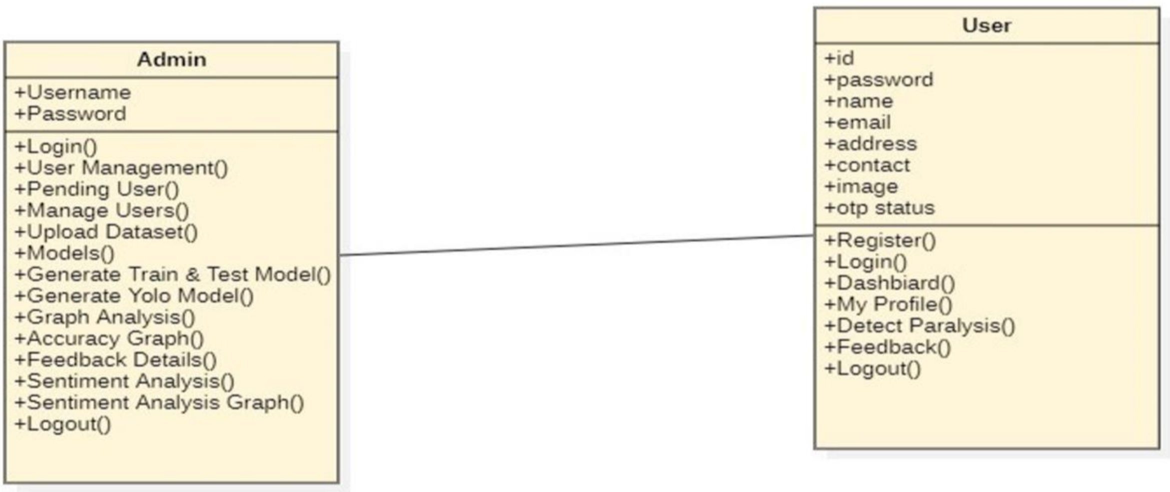


Fig 4.1 Class Diagram

Use case Diagram:

A use case diagram is one that shows the details of a use case and shows how a user interacts with the system. It is possible to depict the various user types and ways in which they engage with a system through a use case diagram. Different kinds of diagrams are frequently employed in addition to this kind, which is usually utilized in connection with the textual use case.

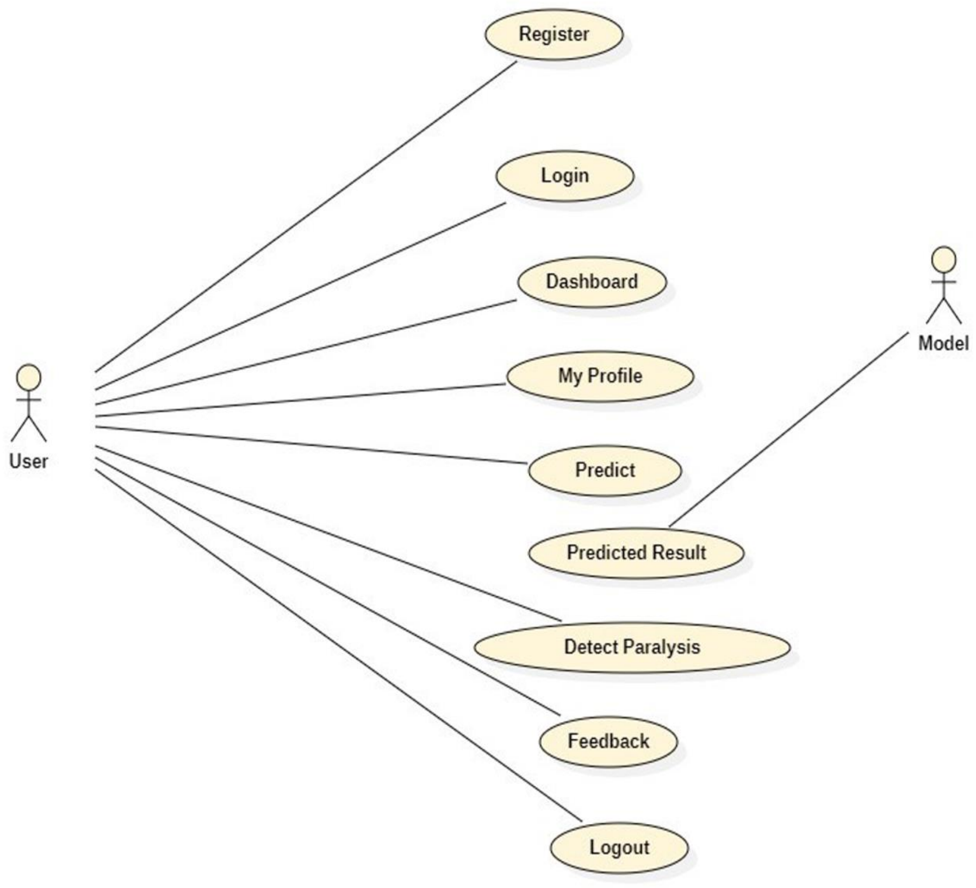


Fig 4.2 Use case Diagram

Sequence Diagram:

A sequence diagram is a type of interaction diagram that illustrates the order in which processes interact with one another. It is an example of a message sequence chart construct. The interactions between objects are arranged chronologically in a sequence diagram. The objects and classes involved in the scenario are shown, along with the message sequence that needs to be exchanged between the objects in order for the scenario to function. In the Logical View of the system that is being developed, sequence diagrams are usually linked to use case realizations. Event diagrams, timing diagrams, and event scenarios are other names for sequence diagrams.

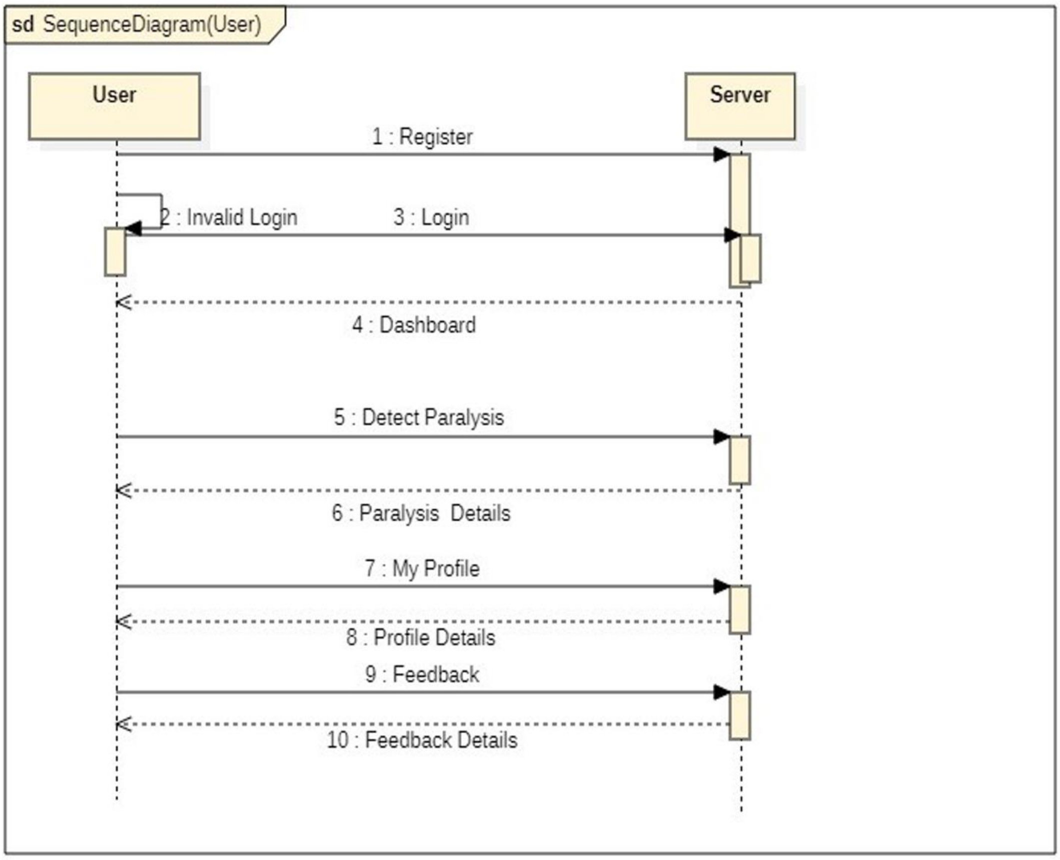


Fig 4.3 Sequence Diagram

Activity Diagram:

Another crucial UML diagram for describing the system's dynamic elements is the activity diagram. To depict the flow from one action to another, it is essentially a flow chart. One could characterize the activity as a system operation. Consequently, the control flow is transferred from one operation to another. This flow has the potential to be concurrent, branching, or sequential.

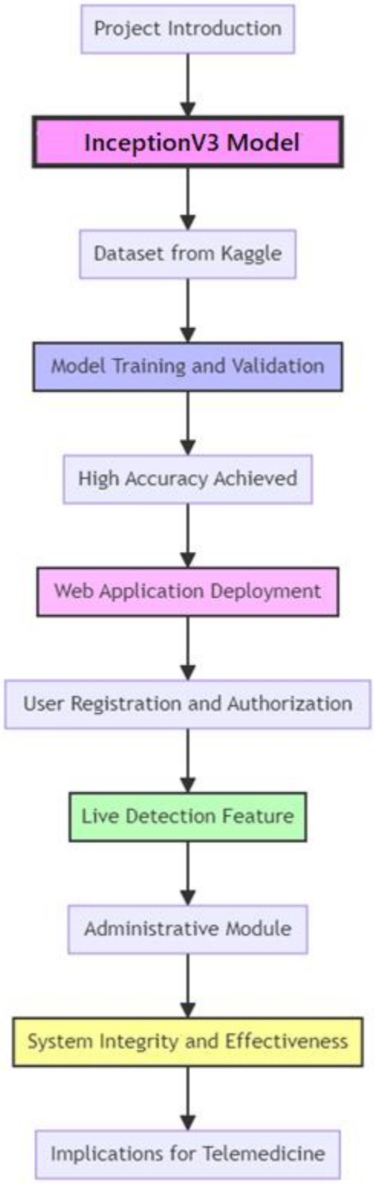


Fig 4.4 Activity Diagram

Data Flow Diagram

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required.The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the processto another and the end results.

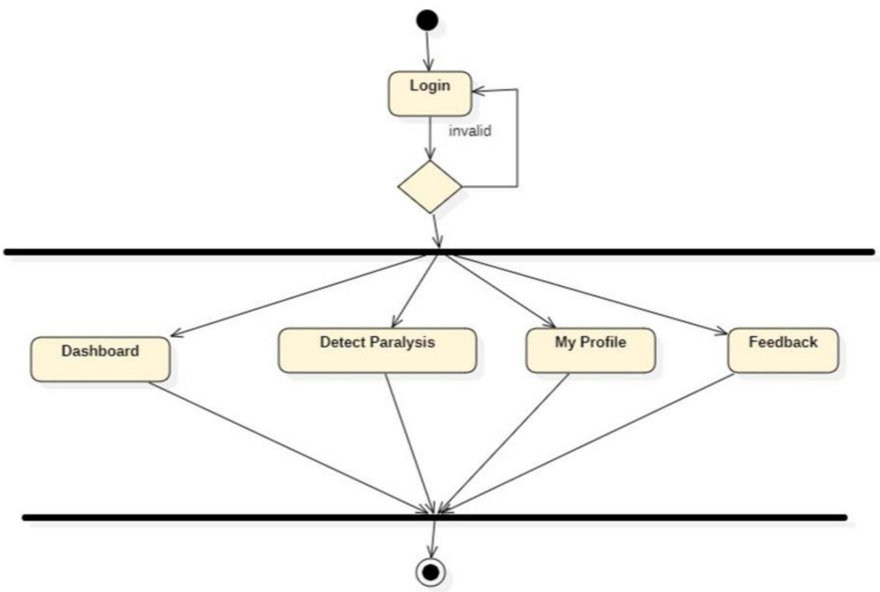


Fig 4.5 Data Flow Diagram

CHAPTER 5
IMPLEMENTATION

PYTHON:

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think aboutthe problem rather than focusing on the syntax.

History of Python:

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980sand was first released in February 1991.

Why Python was created?

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible.This led to design of a new language which was later named Python.

Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

Features of Python:

A simple language which is easier to learn:

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows youto focus on the solution rather than syntax.If you are a newbie, it's a great choice to start.

Free and open-source:

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

Portability:

You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

Extensible and Embeddable:

Suppose an application requires high performance. You can easily combine pieces of C/C++ orother languages with Python code.

This will give your application high performance as well as scripting capabilities which otherlanguages may not provide out of the box.

A high-level, interpreted language:

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

Large standard libraries to solve common tasks:

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

Object-oriented:

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complexproblem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

Applications of Python:

1.Simple Elegant Syntax:

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntaxfeels natural. Take this source code for an example:

```
a = 2
b = 3
sum = a + b print(sum)
```

2.Not overly strict:

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolonat the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

3.Expressiveness of the language:

Python allows you to write programs having greater functionality with fewer lines of code. Here'sa link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

4.Great Community and Support:

Python has a large supporting community. There are numerous active forums online which can behandy if you are stuck.

SAMPLE CODE:

```
# import the libraries as shown below

from tensorflow.keras.layers import Input, Lambda, Dense, Flatten

from tensorflow.keras.models import Model #functional API

from tensorflow.keras.applications.inception_v3 import InceptionV3

#from keras.applications.vgg16 import VGG16

from tensorflow.keras.applications.inception_v3 import preprocess_input

from tensorflow.keras.preprocessing import image

from tensorflow.keras.preprocessing.image import load_img

from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img

from tensorflow.keras.models import Sequential #Sequential API


import numpy as np

from glob import glob

import matplotlib.pyplot as plt


dataset_path = "dataset"

path  = r"dataset\test\bad\apple_bad\1.jpg"

import cv2

import numpy as np

import warnings
```

```
img = cv2.imread(path)

print(img)

# # re-size all the images to this

IMAGE_SIZE = [224, 224]

train_path = 'train/'

valid_path = 'val'

img = cv2.imread(train_path)

print(img)

print()


# re-size all the images to this

IMAGE_SIZE = [224, 224]

train_path =dataset_path +'/train'

valid_path = dataset_path +'/val'

test_path = dataset_path + '/test'


# Import the Vgg 16 library as shown below and add preprocessing layer to the front of VGG

# Here we will be using imagenet weights


inception = InceptionV3(input_shape=IMAGE_SIZE + [3], weights='imagenet',
include_top=False)


# don't train existing weights
```

```
for layer in inception.layers:
    layer.trainable = False
  # useful for getting number of output classes
folders = glob(train_path +'/*')
# our layers - you can add more if you want
x = Flatten()(inception.output)
prediction = Dense(len(folders), activation='softmax')(x)
# create a model object
model = Model(inputs=inception.input, outputs=prediction)
# view the structure of the model
model.summary()
# tell the model what cost and optimization method to use
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
# Use the Image Data Generator to import the images from the dataset
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
```

horizontal_flip = True)

val_datagen = ImageDataGenerator(rescale = 1./255)

test_datagen = ImageDataGenerator(rescale = 1./255,

shear_range = 0.2,

zoom_range = 0.2,

horizontal_flip = True)

Make sure you provide the same target size as initialied for the image size

train_set = train_datagen.flow_from_directory(train_path,target_size = (224, 224),batch_size = 32,class_mode = 'categorical')

val_set = val_datagen.flow_from_directory(valid_path,

target_size = (224, 224),

batch_size = 32,

class_mode = 'categorical')

test_set = test_datagen.flow_from_directory(test_path,

target_size = (224, 224),

batch_size = 32,

class_mode = 'categorical')

fit the model

Run the cell. It will take some time to execute

from PIL import Image from tensorflow.keras.preprocessing.image import load_img import scipy

```
r = model.fit_generator(
train_set, validation_data=val_set, epochs=10,
steps_per_epoch=len(train_set), validation_steps=len(val_set) )
from PIL import Image
from tensorflow.keras.preprocessing.image import load_img
import scipy


# Assuming you have already defined your model and data generators
# train_set and val_set are instances of ImageDataGenerator
# model is your Keras model built using the Functional API
# fit the model
# Run the cell. It will take some time to execute
r = model.fit(train_set, validation_data=val_set, epochs=10, steps_per_epoch=len(train_set),
validation_steps=len(val_set))


# plot the loss
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')


# plot the accuracy
```

```
plt.plot(r.history['accuracy'], label='train acc')

plt.plot(r.history['val_accuracy'], label='val acc')

plt.legend()

plt.show()

plt.savefig('AccVal_acc')

# Print the final training and validation accuracy

final_train_accuracy = r.history['accuracy'][-1]

final_val_accuracy = r.history['val_accuracy'][-1]

print("Final Training Accuracy:", final_train_accuracy)

print("Final Validation Accuracy:", final_val_accuracy)

# save it as a h5 file in inversion v3 model

from tensorflow.keras.models import load_model

model.save('face_stroke.h5')

#load the model

import numpy as np

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

from tensorflow.keras.applications.inception_v3 import preprocess_input

model=load_model('face_stroke.h5')


#img=image.load_img("/content/drive/MyDrive/notebook/dataset/test/Potato___Early_blight
/fa8656fd-03b1-467c-a877-99ce5632d580___RS_Early.B 7630.JPG")

ref = dict(zip(list(train_set.class_indices.values()),list(train_set.class_indices.keys())))
```

```
# def prediction(path):
#   img = image.load_img(path, target_size=(224, 224))
#   i = image.img_to_array(img)
#   i = np.expand_dims(i, axis=0)
#   img = preprocess_input(i)
#   pred = np.argmax(model.predict(img), axis=1)
#   print(f"the image belongs to {ref[pred]}")
# path = r"dataset\test\bad\banana_bad\3.jpg"
# prediction(path)
train_set.class_indices
def prediction(path):
  img = image.load_img(path, target_size=(224, 224))
  i = image.img_to_array(img)
  i = np.expand_dims(i, axis=0)
  img = preprocess_input(i)
  pred = np.argmax(model.predict(img), axis=1)
  print(f"the image belongs to {ref[pred[0]]}")
path = r"dataset\train\Not Paralyse\aug_0_87.jpg"
prediction(path)
#load the model
import numpy as np
from tensorflow.keras.models import load_model
```

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
# load the model
# model=load_model(r"D:\Codebook\fruit_quality_prediction\model_inception.h5")
model = load_model('face_stroke.h5')
def prediction(path):
  img = image.load_img(path, target_size=(224, 224))
  i = image.img_to_array(img)
  i = np.expand_dims(i, axis=0)
  img = preprocess_input(i)
  pred = np.argmax(model.predict(img), axis=1)
  print(f"the image belongs to {[pred[0]]}")
  path = r"dataset\test\Paralyse\27.jpg"
# path = r"dataset\train\Paralyse\13.jpg"
prediction(path)
# {'Not Paralyse': 0, 'Paralyse': 1}
# prediction with display images
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.models import load_model
```

```
model=load_model(r"face_stroke.h5")
def predict_and_display(image_path, model, class_labels):
    img = image.load_img(image_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    prediction = model.predict(img_array)
    predicted_class_index = np.argmax(prediction)
    predicted_class_label = class_labels[predicted_class_index]
    plt.imshow(img)
    plt.axis('off')
    plt.title(f"The Prediction is: {predicted_class_label}")
    plt.show()
# Load your trained model
model.load_weights('face_stroke.h5')

# Define your class labels (e.g., ['car', 'truck', ...])
class_labels = [ 'Not Paralyse', 'Paralyse']

# Replace 'path_to_test_image' with the path to the image you want to test
image_path_to_test = r"dataset\train\Paralyse\13.jpg"
predict_and_display(image_path_to_test, model, class_labels)
```

CHAPTER 6

TESTING

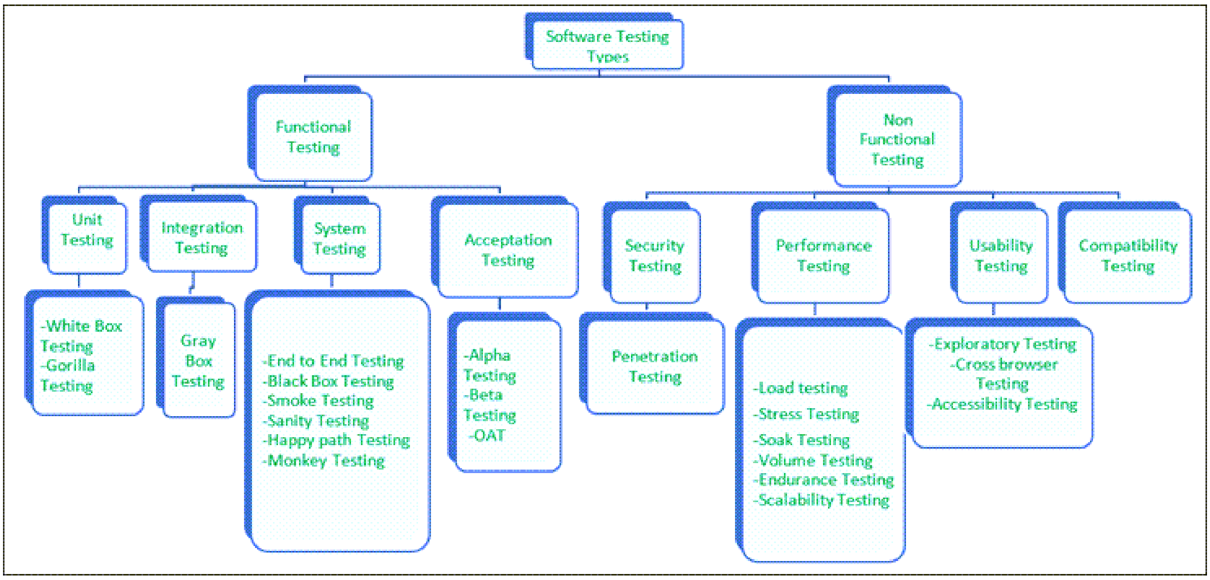
SYSTEM TESTING

Types of Software Testing: Different Testing Types with Details

We, as testers, are aware of the various types of Software Testing like Functional Testing, Non-Functional Testing, Automation Testing, Agile Testing, and their sub-types, etc.

Each type of testing has its own features, advantages, and disadvantages as well. However, in this tutorial, we have covered mostly each and every type of software testing which we usually use in our day-to-day testing life.

Different Types of Software Testing



6.1 Functional Testing

There are four main types of functional testing.

6.1.1 Unit Testing

Unit testing is a type of software testing which is done on an individual unit or component to test its corrections. Typically, Unit testing is done by the developer at the application development phase. Each unit in unit testing can be viewed as a method, function, procedure. Developers often use test automation tools such as NUnit, Xunit, JUnit for the test execution. Unit testing is important because we can find more defects at the unit test level.

For example, there is a simple calculator application. The developer can write the unit test to check if the user can enter two numbers and get the correct sum for addition functionality.

a) White Box Testing

White box testing is a test technique in which the internal structure or code of an application is visible and accessible to the tester. In this technique, it is easy to find loopholes in the design of an application or fault in business logic. Statement coverage and decision coverage/branch coverage are examples of white box test techniques.

b) Gorilla Testing

Gorilla testing is a test technique in which the tester and/or developer test the module of the application thoroughly in all aspects. Gorilla testing is done to check how robust your application is.

For example, the tester is testing the pet insurance company's website, which provides the service of buying an insurance policy, tag for the pet, Lifetime membership. The tester can focus on any one module, let's say, the insurance policy module, and test it thoroughly with positive and negative test scenarios.

6.1.2 Integration Testing

Integration testing is a type of software testing where two or more modules of an application are logically grouped together and tested as a whole. The focus of this type of testing is to find the defect on interface, communication, and data flow among modules. Top-down or Bottom-up approach is used while integrating modules into the whole system.

This type of testing is done on integrating modules of a system or between systems.

For example, a user is buying a flight ticket from any airline website. Users can see flight details and payment information while buying a ticket, but flight details and payment processing are two different systems. Integration testing should be done while integrating of airline website and payment processing system.

a) Gray box testing

As the name suggests, gray box testing is a combination of white-box testing and black-box testing. Testers have partial knowledge of the internal structure or code of an application.

6.1.3 System Testing

System testing is types of testing where tester evaluates the whole system against the specified requirements.

a) End to End Testing

It involves testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

For example, a tester is testing a pet insurance website. End to End testing involves testing of buying an insurance policy, LPM, tag, adding another pet, updating credit card information on users' accounts, updating user address information, receiving order confirmation emails and policy documents.

b) Black Box Testing

Blackbox testing is a software testing technique in which testing is performed without knowing the internal structure, design, or code of a system under test. Testers should focus only on the input and output of test objects. Detailed information about the advantages, disadvantages, and types of Black Box testing can be found here.

c) Smoke Testing

Smoke testing is performed to verify that basic and critical functionality of the system under test is working fine at a very high level.

Whenever a new build is provided by the development team, then the Software Testing team validates the build and ensures that no major issue exists. The testing team will ensure that the build is stable, and a detailed level of testing will be carried out further.

For example, tester is testing pet insurance website. Buying an insurance policy, adding another pet, providing quotes are all basic and critical functionality of the application. Smoke testing for this website verifies that all these functionalities are working fine before doing any in-depth testing.

d) Sanity Testing

Sanity testing is performed on a system to verify that newly added functionality or bug fixes are working fine. Sanity testing is done on stable build. It is a subset of the regression test.

For example, a tester is testing a pet insurance website. There is a change in the discount for buying a policy for second pet. Then sanity testing is only performed on buying insurance policy module.

e) Happy path Testing

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on valid and positive inputs through which the application generates the expected output.

f) Monkey Testing

Monkey Testing is carried out by a tester, assuming that if the monkey uses the application, then how random input and values will be entered by the Monkey without any knowledge or understanding of the application.

The objective of Monkey Testing is to check if an application or system gets crashed by providing random input values/data. Monkey Testing is performed randomly, no test cases are scripted, and it is not necessary to be aware of the full functionality of the system.

6.1.4 Acceptance Testing

Acceptance testing is a type of testing where client/business/customer test the software with real time business scenarios.

a) Alpha Testing

Alpha testing is a type of acceptance testing performed by the team in an organization to find as many defects as possible before releasing software to customers.

For example, the pet insurance website is under UAT. UAT team will run real-time scenarios like buying an insurance policy, buying annual membership, changing the address, ownership transfer of the pet in a same way the user uses the real website. The team can use test credit card information to process payment-related scenarios.

b) Beta Testing

Beta Testing is a type of software testing which is carried out by the clients/customers. It is performed in the **Real Environment** before releasing the product to the market for the actual end-users. Beta Testing is carried out to ensure that there are no major failures in the software or product, and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software.Usually, this testing is typically done by the end-users. This is the final testing done before releasing the application for commercial purposes. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.So, the end-user uses the software and shares the feedback with the company. The company then takes necessary action before releasing the software worldwide.

c) Operational acceptance testing (OAT)

Operational acceptance testing of the system is performed by operations or system administration staff in the production environment. The purpose of operational acceptance testing is to make sure that the system administrators can keep the system working properly for the users in a real-time environment.

The focus of the OAT is on the following points:

- Testing of backup and restore.
- Installing, uninstalling, upgrading software.
- The recovery process in case of natural disaster.
- User management.

6.2 Non-Functional Testing

There are four main types of functional testing.

6.2.1 Security Testing

It is a type of testing performed by a special team. Any hacking method can penetrate the system.

Security Testing is done to check how the software, application, or website is secure from internal and/or external threats. This testing includes how much software is secure from malicious programs, viruses and how secure & strong the authorization and authentication processes are.
It also checks how software behaves for any hacker's attack & malicious programs and how software is maintained for data security after such a hacker attack.

a) Penetration Testing

Penetration Testing or Pen testing is the type of security testing performed as an authorized cyberattack on the system to find out the weak points of the system in terms of security.Pen testing is performed by outside contractors, generally known as ethical hackers. That is why it is also known as ethical hacking. Contractors perform different operations like SQL injection, URL manipulation, Privilege Elevation, session expiry, and provide reports to the organization.

Notes: Do not perform the Pen testing on your laptop/computer. Always take written permission to do pen tests.

6.2.2 Performance Testing

Performance testing is testing of an application's stability and response time by applying load. The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, LoadRunner, etc. are good tools available in the market.Your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

a) Load testing

Load testing is testing of an application's stability and response time by applying load, which is equal to or less than the designed number of users for an application.

For example, your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

b) Stress Testing

Stress testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.

For example, your application handles 1000 users at a time with a response time of 4 seconds, then stress testing can be done by applying a load of more than 1000 users. Test the application with 1100,1200,1300 users and notice the response time. The goal is to verify the stability of an application under stress.

c) Scalability Testing

Scalability testing is testing an application's stability and response time by applying load, which is more than the designed number of users for an application.

For example, your application handles 1000 users at a time with a response time of 2 seconds, then scalability testing can be done by applying a load of more than 1000 users and gradually increasing the number of users to find out where exactly my application is crashing

d) Volume testing (flood testing)

Volume testing is testing an application's stability and response time by transferring a large volume of data to the database. Basically, it tests the capacity of the database to handle the data.

e) Endurance Testing (Soak Testing)

Endurance testing is testing an application's stability and response time by applying load continuously for a longer period to verify that the application is working fine.

For example, car companies soak testing to verify that users can drive cars continuously for hours without any problem.

6.2.3 Usability Testing

Usability testing is testing an application from the user's perspective to check the look and feel and user-friendliness.

For example, there is a mobile app for stock trading, and a tester is performing usability testing. Testers can check the scenario like if the mobile app is easy to operate with one hand or not, scroll bar should be vertical, background colour of the app should be black and price of and stock is displayed in red or green colour.

The main idea of usability testing of this kind of app is that as soon as the user opens the app, the user should get a glance at the market.

Exploratory Testing is informal testing performed by the testing team. The objective of this testing is to explore the application and look for defects that exist in the application. Testers use the knowledge of the business domain to test the application. Test charters are used to guide the exploratory testing.

a) Cross browser testing

Cross browser testing is testing an application on different browsers, operating systems, mobile devices to see look and feel and performance. Why do we need cross-browser testing? The answer is different users use different operating systems, different browsers, and different mobile devices. The goal of the company is to get a good user experience regardless of those devices. Browser stack provides all the versions of all the browsers and all mobile devices to test the application. For learning purposes, it is good to take the free trial given by browser stack for a few days.

b) Accessibility Testing

The aim of Accessibility Testing is to determine whether the software or application is accessible for disabled people or not.

Here, disability means deafness, colour blindness, mentally disabled, blind, old age, and other disabled groups. Various checks are performed, such as font size for visually disabled, colour and contrast for colour blindness, etc.

6.2.4 Compatibility testing

This is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment. Various checks are performed, such as font size for visually disabled, colour and contrast for colour blindness, etc.

CHAPTER 7

RESULTS

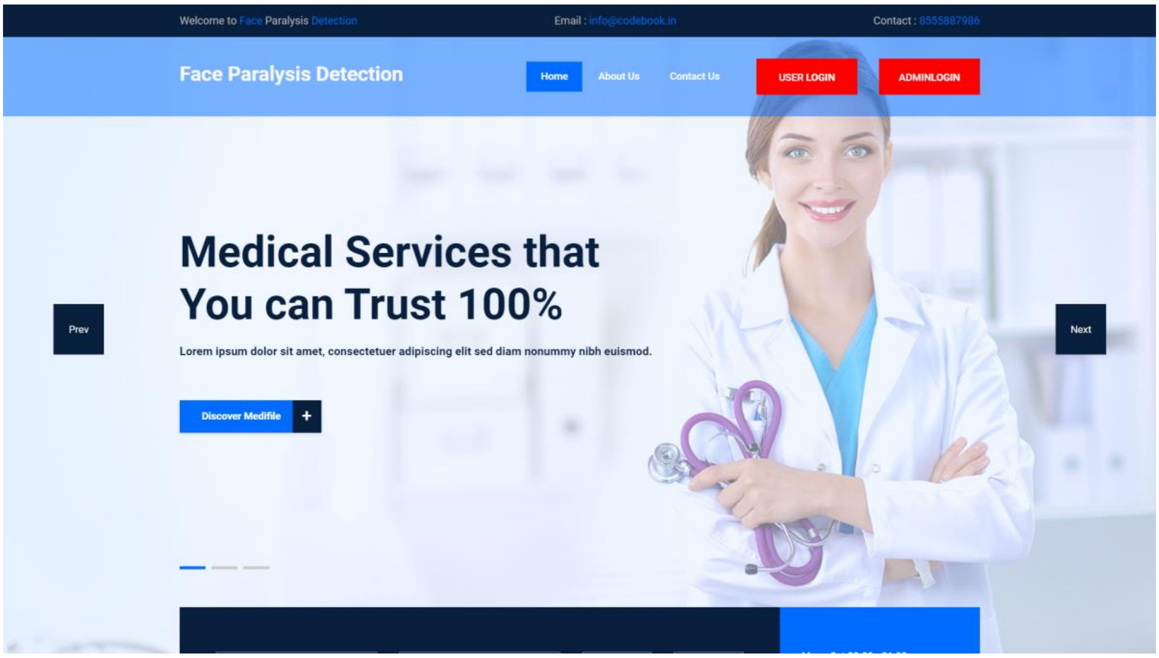


Fig:7.1 Home page

In the above screen click on 'User Login' button to login to your account

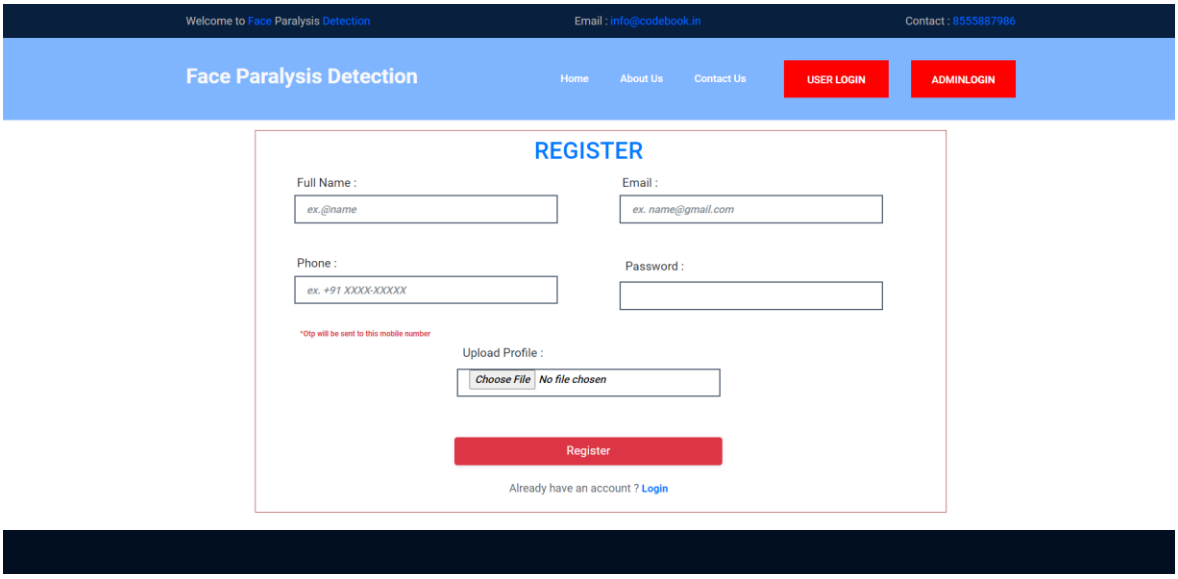


Fig 7.2:User registration

In the above screen click on 'register' first before login to account

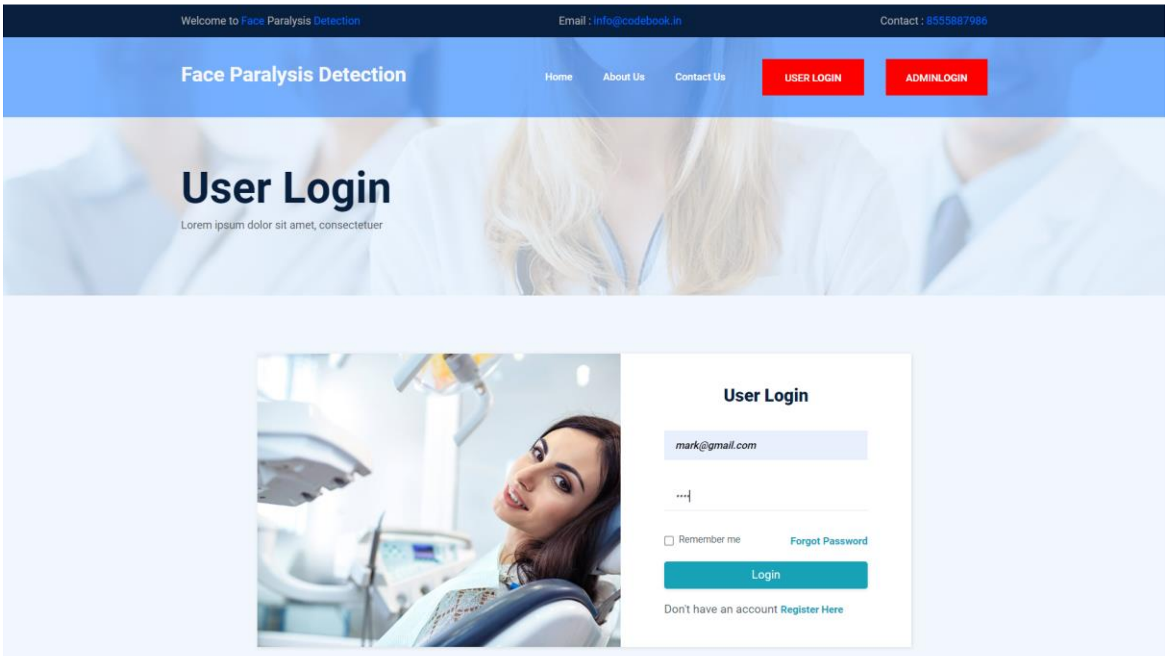


Fig 7.3 User login page

In the above screen click on 'login' to upload images

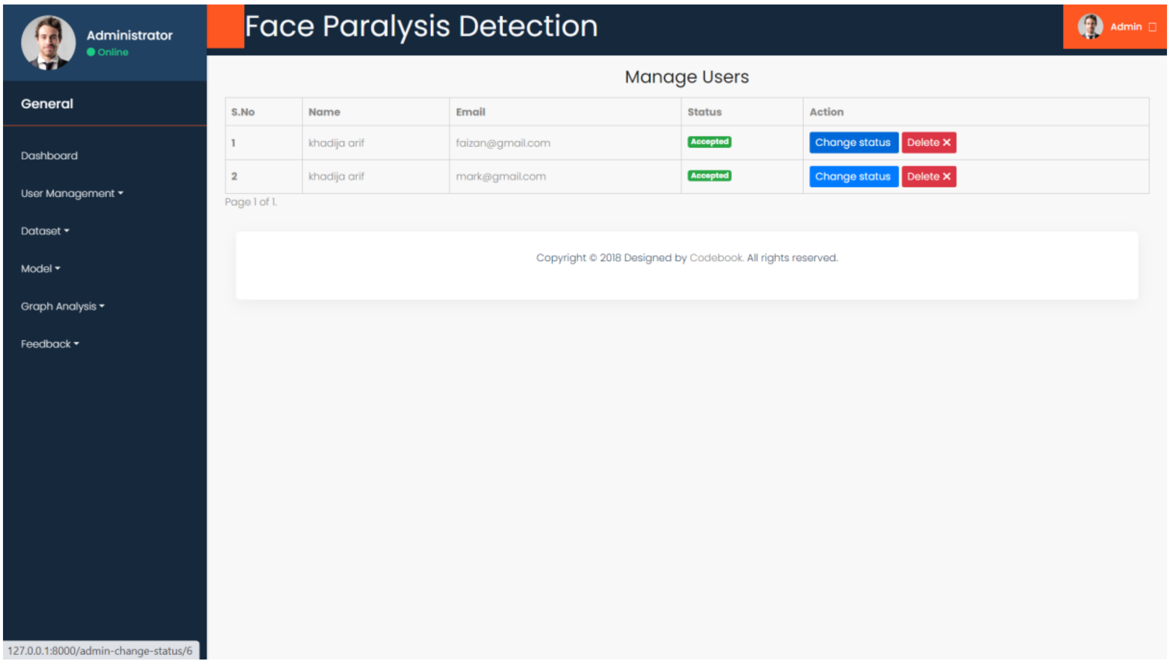


Fig 7.4 Admin Manage Users

In the above screen admin 'accept the user request'

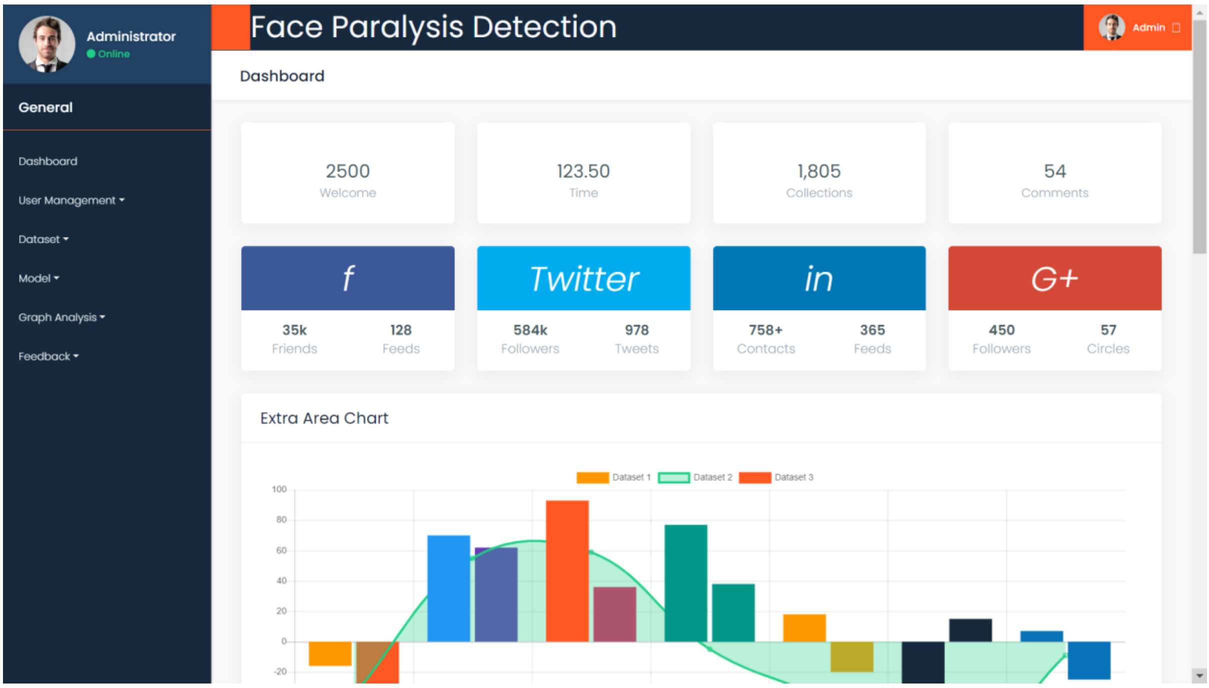


Fig 7.5 Admin Dashboard

In the above screen admin see the ‘Dashboard’ shows all the details

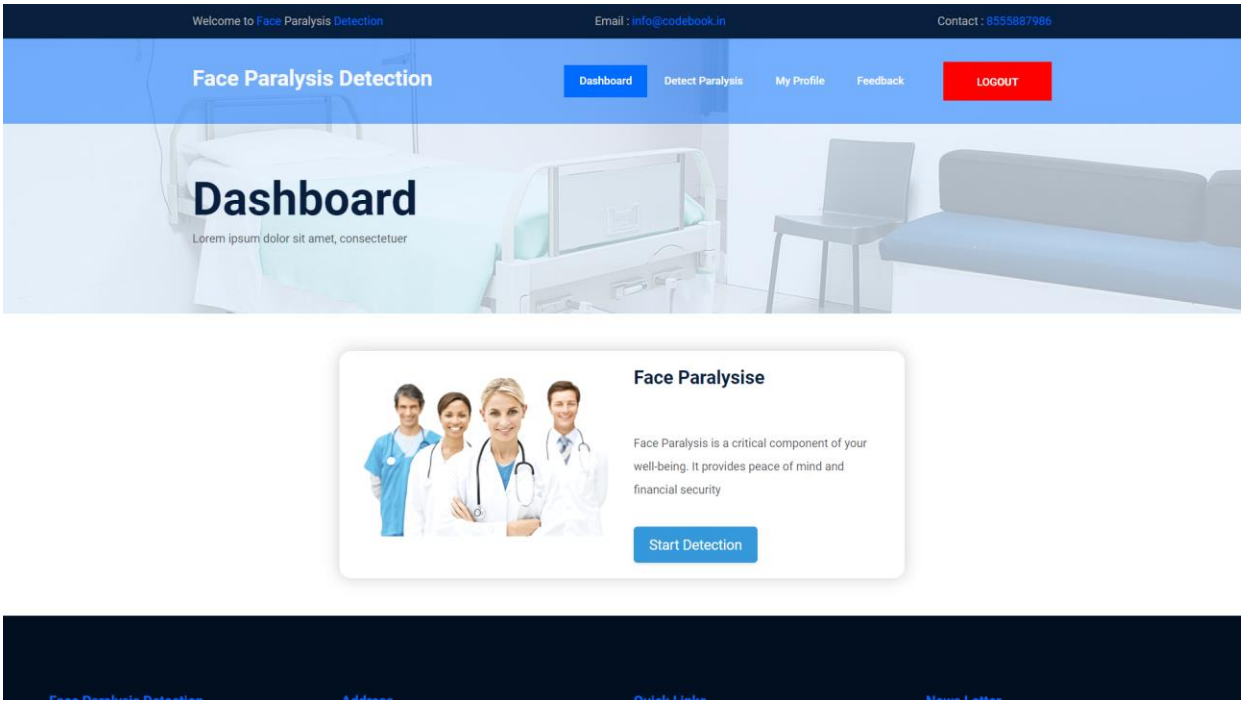


Fig 7. 6 User Dashboard

In the above screen click on ‘Start Detection’ to detect

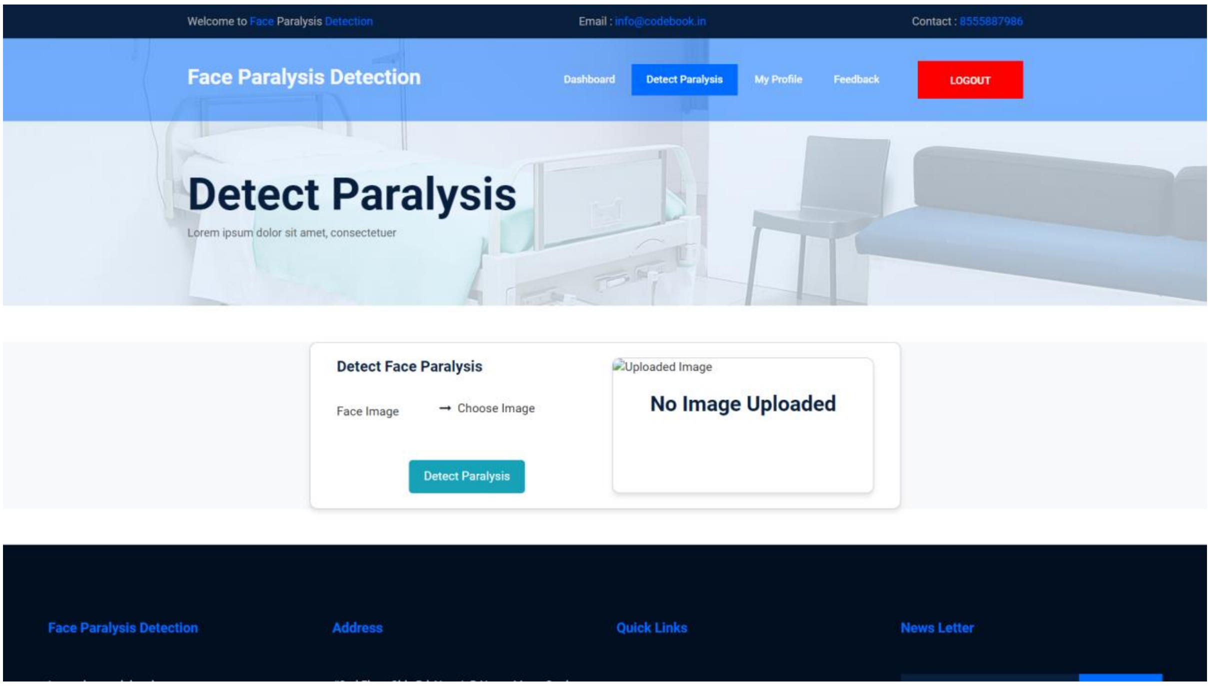


Fig 7.7 Upload Image

In the above screen select and upload the image then click on 'Detect Paralysis' to load dataset

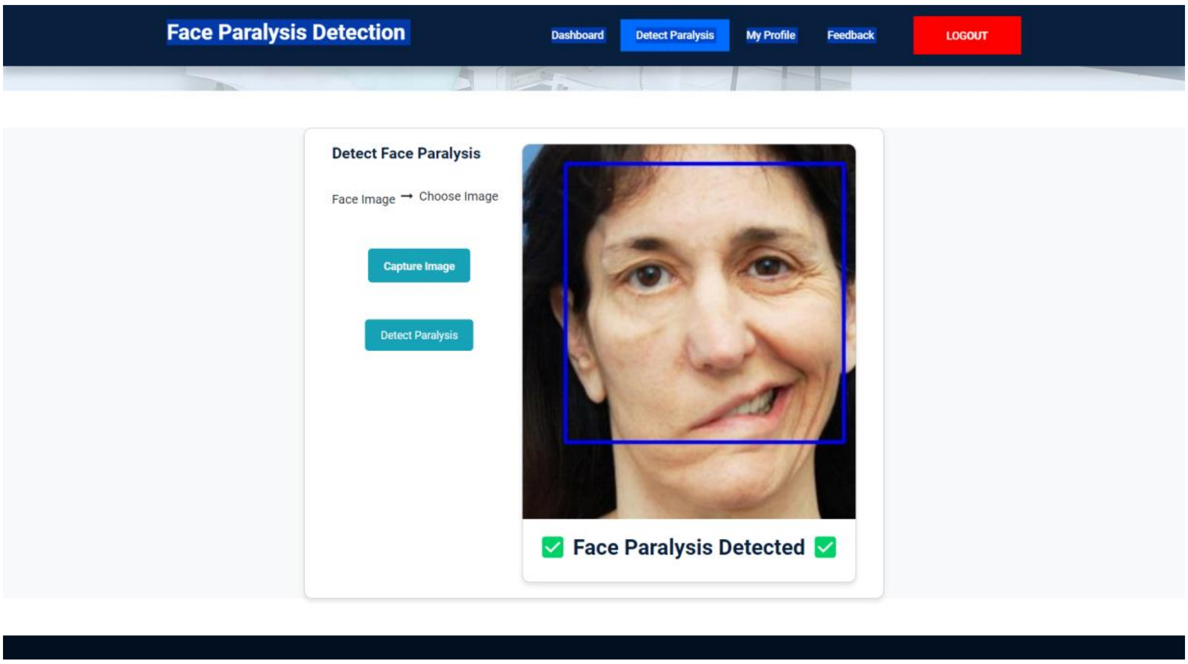


Fig 7.8 Face paralysis Detection

In the above screen image results are uploaded

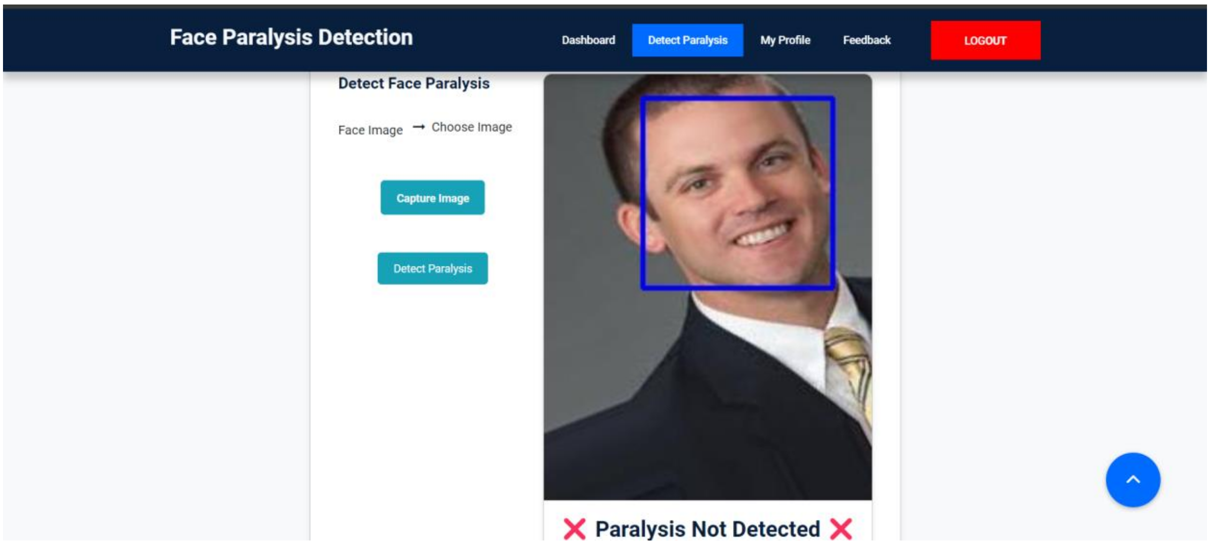


Fig 7.9 Face paralysis Detection

In the above screen results are shown

CHAPTER 8

CONCLUSION

This paper presents an approach for the severity classification of face paralysis, which occurs due to facial muscle weakness and nerve damage, resulting in impaired facial function. Our transfer learning approach addresses the limitations of existing models and proposes a facial palsy classification model for grading the severity level of patients. The results reveal the superior performance of our model by achieving a remarkable accuracy of 99.3%. Overall, our deep learning approach, incorporating transfer learning and fine-tuning, enables us to leverage pre-existing knowledge and features extracted from facial images for accurate and efficient face paralysis detection and classification. An automatic facial paralysis detection and classification system has significant benefits for both physicians and patients involved in the rehabilitation process.

It can assist physicians in selecting the most suitable treatment plan based on accurate and objective assessments of the patient’s condition. Additionally, it enables patients to track and evaluate their recovery progress throughout the treatment process. Future research in the field of facial paralysis holds significant potential for advancements and improvements in healthcare facilities. The domain of facial paralysis lacks generalized models for effective detection and classification. Key areas where advancements can be made include the preparation of datasets related to facial paralysis. This involves the collection of diverse and representative datasets featuring cases of facial paralysis. In the future, an incremental learning approach could be employed. The model can be designed to learn incrementally as new data becomes available over time.

CHAPTER 9

FUTURE ENHANCEMENTS

1. **Fine-grained Transfer Learning**:

Current transfer learning methods often involve transferring knowledge from a pre-trained model to a related task. Future advancements may focus on fine-grained transfer, where models can transfer knowledge at a more granular level, adapting to specific nuances and requirements of new tasks.

2. **Domain Adaptation**:

AI systems often struggle when faced with data from new domains or environments. Future enhancements could include more robust domain adaptation techniques, allowing models to generalize better across diverse datasets without extensive retraining.

3. **Lifelong Learning**:

Continual learning or lifelong learning is an area of active research. Future AI systems may be designed to learn continuously from new data, retaining past knowledge while efficiently incorporating new information over time.

4. **Zero-shot and Few-shot Learning**:

These approaches aim to train models with minimal labeled data. Future enhancements may lead to AI systems that can generalize effectively from just a few examples or even perform tasks they haven't been explicitly trained on, using zero-shot learning methods.

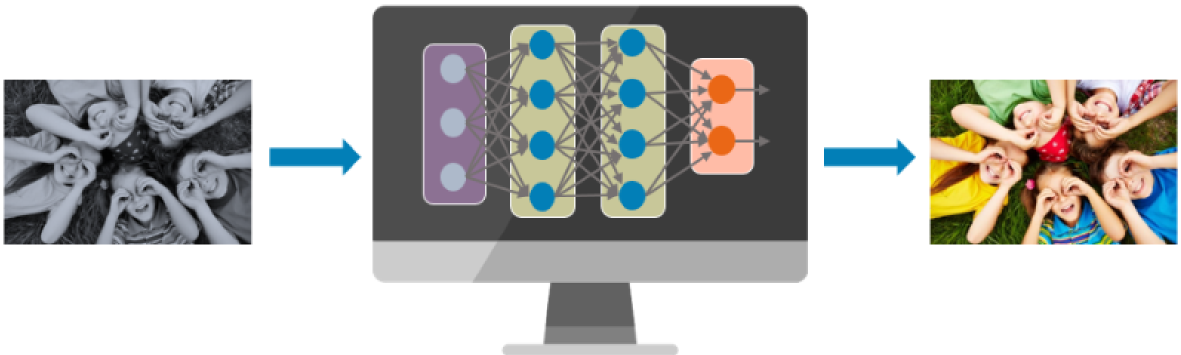
These enhancements collectively aim to advance the capabilities, reliability, and ethical standards of AI systems, making them more adaptable, intelligent, and trustworthy in various applications.

CHAPTER 10

APPLICATIONS

1. Image Coloring

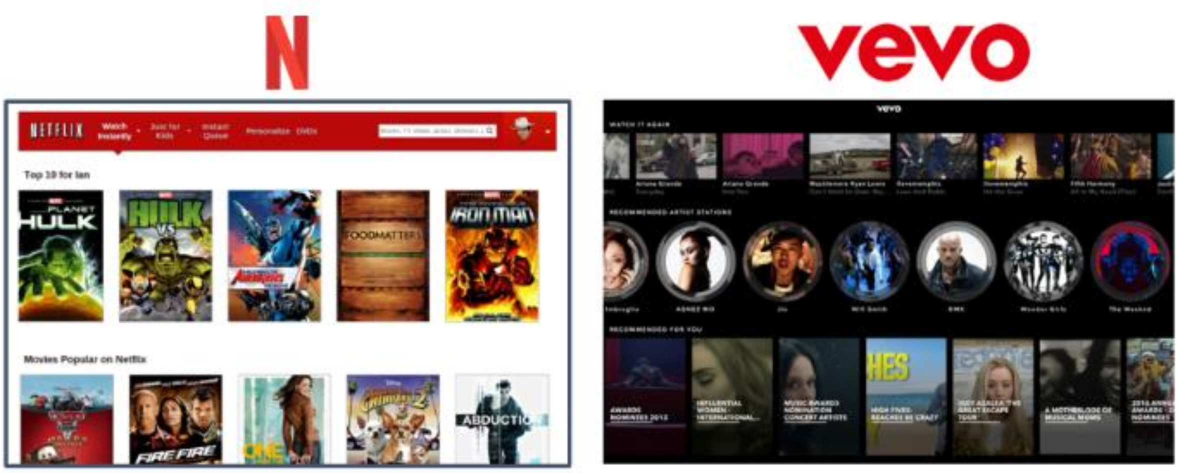
Deep Learning has significantly improved image colorization, with models like ChromaGAN incorporating perceptual and semantic understanding of class distributions and color to produce



colorized images from grayscale inputs.

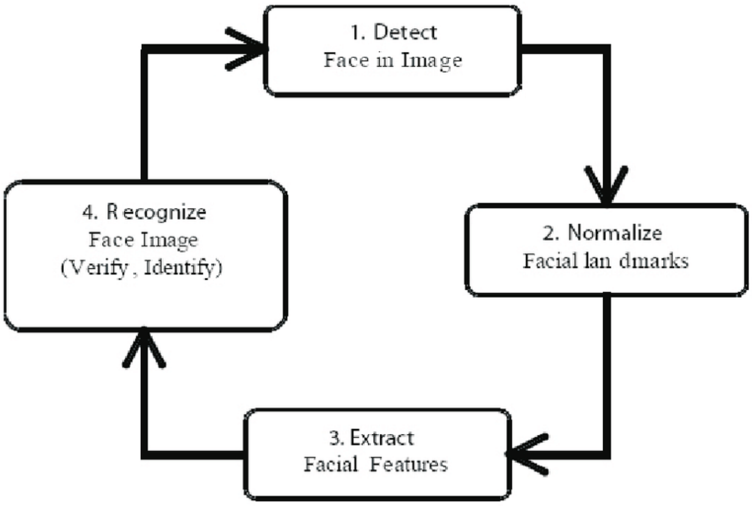
2. Entertainment

Deep Learning is used by companies like Netflix, Amazon, YouTube, and Spotify to provide personalized recommendations based on browsing history, interest, and behaviour, and automatically add subtitles.

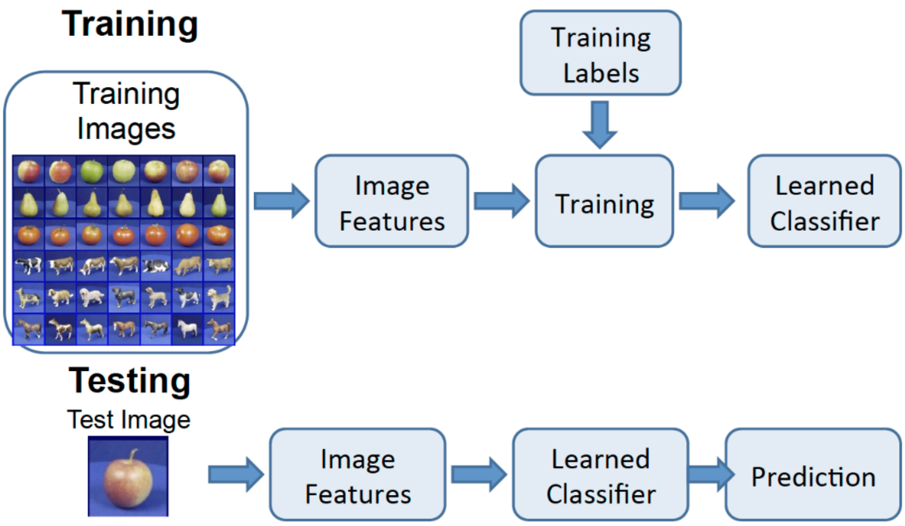


3. Image recognition

Image recognition is a technology used to identify objects, people, and entities in digital images or videos. With the rise of smartphones and high-resolution cameras, industries utilize this vast volume of data to deliver innovative services. Computer vision, a broader team, includes object recognition, image reconstruction, event detection, and video tracking.

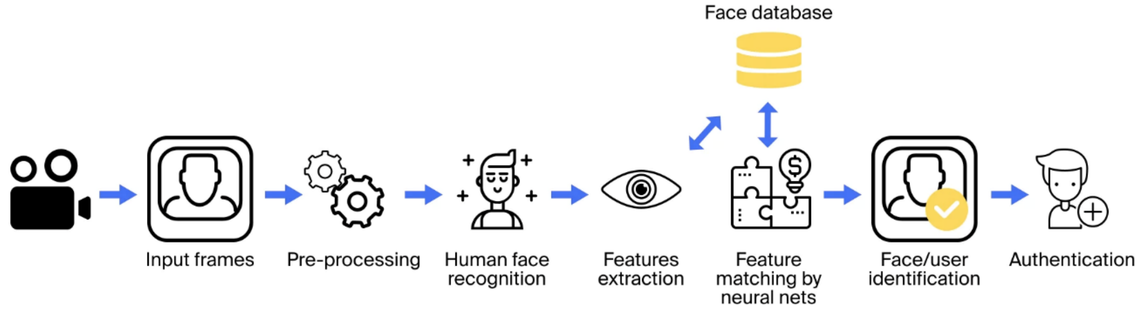


4. Object Detection



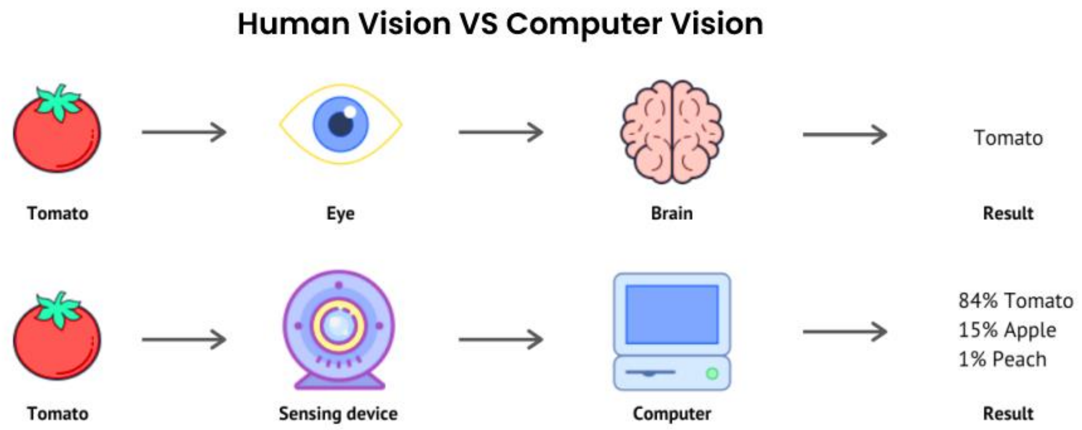
Object detection is a method that classifies and locates objects in an image, combining image classification and object localization. It involves inputs and outputs, with the input being an image with objects, and outputs being bounding boxes labelled for classification.

5. Facial Recognition



A face analyser is software that uses facial features to identify or confirm a person's identity, often used in biometric security systems for unique identification during onboarding or logins, and for device security on mobile and personal devices.

6.Computer vision



Deep learning has revolutionized computer vision, enabling object detection, image classification, and segmentation in applications like autonomous driving, robotics, healthcare, and immersive virtual and augmented reality experiences, transforming machine interpretation and understanding of visual information.

CHAPTER 11

REFERENCES

[1] P Subba Rao. "Applications of machine learning in environmental engineering" JOURNAL OF Science, Technology and Development (2021): 643-648.

[2] P Subba Rao. "Using an Energy Efficient Extended Leach Work with Multilevel Clustering Approach" from JOURNAL of Research Publication and Reviews (2020): 60-64.

[3] C. Jiang, J. Wu, W. Zhong, M. Wei, J. Tong, H. Yu, and L. Wang, ‘‘Automatic facial paralysis assessment via computational image analysis,’’ J. Healthcare Eng., vol. 2020, pp. 1–10, Feb. 2020.

[4] D. L. Guarin, Y. Yunusova, B. Taati, J. R. Dusseldorp, S. Mohan, J. Tavares, M. M. van Veen, E. Fortier, T. A. Hadlock, and N. Jowett, ‘‘Toward an automatic system for computer aided assessment in facial palsy,’’ Facial Plastic Surg. Aesthetic Med., vol. 22, no. 1, pp. 42–49, Feb. 2020.

[5] J. J. Greene, D. L. Guarin, J. Tavares, E. Fortier, M. Robinson, J. Dusseldorp, O. Quatela, N. Jowett, and T. Hadlock, ‘‘The spectrum of facial palsy: The MEEI facial palsy photo and video standard set,’’ Laryngoscope, vol. 130, no. 1, pp. 32–37, Jan. 2020.

[6] Z. Guo, W. Li, J. Dai, J. Xiang, and G. Dan, ‘‘Facial imaging and landmark detection technique for objective assessment of unilateral peripheral facial paralysis,’ Enterprise Inf. Syst., vol. 16, nos. 10–11, pp. 1556–1572, Oct. 2022.

ANNEXURE

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12 , Issue 2, April 2024

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

Paradesi Subba Rao[a)], N. Rama Devi[b)] , Palagulla Lakshmi Priya[c)],

Kristipati Akhila, Myla Venkata Ruchitha, Shaik Ayesha, Shaik karimun

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, Andhra Pradesh, 518501, India.

[a)]Corresponding author: subbarao.cse@srecnandyal.edu.in,

[b)]ramadevi.cse@srecnandyal.edu.in,

[c)]20x51a3238@srecnandyal.edu.in.

ABSTRACT

This project introduces a novel deep learning based approach to detecting facial paralysis, a condition that affects millions worldwide but often goes undiagnosed in its early stages. Leveraging the InceptionV3 algorithm for its robustness and efficiency in real-time object detection, we have developed a model that accurately identifies signs of facial paralysis from both static images and live video feeds. The dataset, sourced from Kaggle's official website, comprises 3,000 images categorized into two classes, enabling the model to learn diverse representations of the condition. Our model achieved an exceptional accuracy of 95.3% during testing, underscoring its potential as a reliable diagnostic tool.

The model was trained and validated using Google Collab, benefiting from its powerful computational resources and collaborative environment. Subsequent deployment in a web application provides an accessible platform for users to register and obtain authorization from an administrator. Upon login, users can upload images or access a live camera feed to detect facial paralysis, facilitating early diagnosis and intervention. The web application also features an administrative module responsible for user authentication, model training, and performance analysis, ensuring the system's integrity and effectiveness.

This project not only showcases the application of deep learning in addressing critical health issues but also sets a precedent for the development of accessible diagnostic tools that can be deployed at scale, offering significant implications for telemedicine and remote healthcare services.

Keywords: facial paralysis detection, InceptionV3 algorithm, real-time object detection, telemedicine, remote healthcare services.

I. INTRODUCTION

Facial paralysis is a medical condition characterized by the loss of voluntary muscle movement in one or both sides of the face, which can significantly affect a person's quality of life through impairments in facial expressions, eating, speaking, and even leading to psychological distress. The causes of facial paralysis are varied, including Bell's palsy, stroke, Lyme disease, and trauma, making its diagnosis complex and necessitating specialized medical expertise. Traditional diagnostic processes are often time-consuming, costly, and not easily accessible to everyone, particularly in less urbanized or resource-limited regions.

In response to these challenges, this project introduces an innovative solution leveraging the advancements in deep learning and computer vision. By adopting the InceptionV3 algorithm, renowned for its speed and accuracy in object detection tasks, we have developed a sophisticated model capable of detecting facial paralysis from both static images and live video streams. This approach not only democratizes access to diagnostic tools but also significantly enhances the efficiency and accuracy of early-stage detection.

The development and deployment of this model within a web-based application further extend its accessibility and usability. Through this platform, users can easily register, upload images, or use a live camera feed to receive immediate diagnostic feedback on the presence of facial paralysis signs. An administrative module ensures the system's integrity by managing user access, overseeing model performance, and facilitating continuous improvement through retraining processes.

This project stands at the intersection of technology and healthcare, aiming to bridge the gap between advanced medical diagnostics and patient accessibility. By providing a scalable, accurate, and user-friendly tool for the early detection of facial paralysis, it paves the way for prompt intervention and treatment, potentially mitigating long-term impacts on patients' lives.

II. LITERATURE SURVEY

TITLE: Automatic facial paralysis assessment via computational image analysis,'

ABSTRACT: Facial paralysis (FP) is a loss of facial movement due to nerve damage. Most existing diagnosis systems of FP are subjective, e.g., the House–Brackmann (HB) grading system, which highly depends on the skilled clinicians and lacks an automatic quantitative assessment. In this paper, we propose an efficient yet objective facial paralysis assessment approach via automatic computational image analysis. First, the facial blood flow of FP patients is measured by the technique of laser speckle contrast imaging to generate both RGB color images and blood flow images. Second, with an improved segmentation approach, the patient's face is divided into concerned regions to extract facial blood flow distribution characteristics. Finally, three HB score classifiers are employed to quantify the severity of FP patients. The proposed method has been validated on 80 FP patients, and quantitative results demonstrate that our method, achieving an accuracy of 90.14%, outperforms the state-of-the-art systems. Experimental evaluations also show that the proposed approach could yield objective and quantitative FP diagnosis results, which agree with those obtained by an experienced clinician.

TITLE: 'Objective grading facial paralysis severity using a dynamic 3D stereo photogrammetry imaging system,''

ABSTRACT: Facial paralysis is a loss of facial movement due to nerve damage. It is essential for clinicians to diagnose the severity of facial paralysis to treat patients, assess progress and evaluate outcomes. Subjective assessments are common in clinical practices but have their limitations regarding the intra-observer and inter-observer reproducibility. We utilized dynamic 3D stereo photogrammetry technology for the objective grading of facial paralysis by measuring regional facial asymmetries. The correlations between the measured asymmetries and scores of a modified Sunnybrook facial paralysis grading were evaluated to identify the region of interests of objective measurements closely related to the subjective grades. Categorical classifiers were trained to quantify the severity of facial paralysis. Preliminary results showed that the objective asymmetry measurements were highly correlated to the subjective assessments of facial paralysis except the eye region. Machine learning approaches showed the potential of improving the accuracy of severity assessments.

TITLE: Toward an automatic system for computer aided assessment in facial palsy,'

ABSTRACT: Quantitative assessment of facial function is challenging, and subjective grading scales such as House-Brackmann, Sunnybrook, and eFACE have well-recognized limitations. Machine learning (ML) approaches to facial landmark localization carry great clinical potential as they enable high-throughput automated quantification of relevant facial metrics from photographs and videos. However, the translation from research settings to clinical application still requires important improvements.

Objective: To develop a novel ML algorithm for fast and accurate localization of facial landmarks in photographs of facial palsy patients and utilize this technology as part of an automated computer-aided diagnosis system.

Design, Setting, and Participants: Portrait photographs of 8 expressions obtained from 200 facial palsy patients and 10 healthy participants were manually annotated by localizing 68 facial landmarks in each photograph and by 3 trained clinicians using a custom graphical user interface. A novel ML model for automated facial landmark localization was trained using this disease-specific database. Algorithm accuracy was compared with manual markings and the output of a model trained using a larger database consisting only of healthy subjects.

Main Outcomes and Measurements: Root mean square error normalized by the interocular distance (NRMSE) of facial landmark localization between prediction of ML algorithm and manually localized landmarks. **Results:** Publicly available algorithms for facial landmark localization provide poor localization accuracy when applied to photographs of patients compared with photographs of healthy controls (NRMSE, 8.56 ± 2.16 vs. 7.09 ± 2.34, $p \ll 0.01$). We found significant improvement in facial landmark localization accuracy for the facial palsy patient population when using a model trained with a relatively small number photographs (1440) of patients compared with a model trained using several thousand more images of healthy faces (NRMSE, 6.03 ± 2.43 vs. 8.56 ± 2.16, $p \ll 0.01$).

Conclusions and Relevance: Retraining a computer vision facial landmark detection model with fewer than 1600 annotated images of patients significantly improved landmark detection performance in frontal view photographs of this population. The new annotated database and facial landmark localization model represent the first steps toward an automatic system for computer-aided assessment in facial palsy.

TITLE: "The spectrum of facial palsy: The MEEI facial palsy photo and video standard set,'

ABSTRACT: Facial palsy causes variable facial disfigurement ranging from subtle asymmetry to crippling deformity. There is no existing standard database to serve as a resource for facial palsy education and research. We present a standardized set of facial photographs and videos representing the entire spectrum of flaccid and nonflaccid (aberrantly regenerated) facial palsy. To demonstrate the utility of the dataset, we describe the relationship between level of facial function and perceived emotion expression as determined by an automated emotion detection, machine learning-based algorithm.

TITLE: "Facial imaging and landmark detection technique for objective assessment of unilateral peripheral facial paralysis,"

ABSTRACT: In this paper, we propose a hypothesis that the facial landmark detection methods constructed by a private UPFP facial dataset can perform better than the model on a healthy facial dataset in the task of UPFP facial landmark detection. For proving this hypothesis, a customized UPFP facial dataset with 68 facial landmark annotations was built. A state-of-the-art facial landmark detection method was employed on the three evaluation datasets to exploit and prove the hypothesis. The mean error of validation dataset is 3.15, 56% lower than 7.42 that of the healthy dataset, which proves the hypothesis is true.

TITLE: "Automatic assessment of facial paralysis based on facial landmarks,"

ABSTRACT: Unilateral peripheral facial paralysis is the most common case of facial paralysis. It affects only one side of the face, which will cause facial asymmetry. Clinically, unilateral peripheral facial paralysis is often classified by clinicians according to evaluation scales, based on patients' condition of facial symmetry. A prevalent scale is House-Brackmann grading system (HBGS).
However, assessment results from scales are often with great subjectivity, and will bring high interobserver and interobserver variability. Therefore, this manuscript proposed an objective method to provide assessment results by using facial videos and applying machine learning models. This grading method is based on HBGS, but it is automatically implemented with high objectivity. Images with facial expressions will be extracted from the videos to be analyzed by a machine learning model. Facial landmarks will be acquired from the images by using a 68-points model provided by dlib. Then index and coordinate information of the landmarks will be used to calculate the values of features pre-designed to train the model and predict the result of new patients. Due to the difficulty of collecting facial paralysis samples, the data size is limited. Random Forest (RF) and support vector machine (SVM) were compared as classifiers. This method was applied on a data set of 33 subjects. The highest overall accuracy rate reached 88.9%, confirming the effectiveness of this method.

III. METHODOLOGY

. 1. Dataset Collection and Preparation

- **Data Sourcing**: The project utilizes a dataset from Kaggle, comprising 3,000 images categorized into two classes to represent facial paralysis and normal facial expressions.

- **Data Augmentation**: To enhance the model's ability to generalize across various facial features and conditions, data augmentation techniques such as rotation, zoom, and horizontal flipping are applied.

2. Model Development

- **INCEPTIONV3 Algorithm**:
  The INCEPTIONV3 algorithm is chosen for its efficiency and accuracy. Its well-suited for real-time facial paralysis detection.

- **Training and Validation**: The model is trained and validated using Google Collab for its GPU support, facilitating faster computation. A validation set is used to tune hyperparameters and avoid overfitting.

3. Performance Evaluation

- **Accuracy Assessment**: The model's performance is evaluated based on its accuracy in detecting facial paralysis, achieving a remarkable 95.3% accuracy on the test dataset.

- **Real-time Testing**: Additional tests are conducted using live video feeds to ensure the model's effectiveness in real-world scenarios.

4. Web Application Development and Deployment

- **User Interface Design**: A user-friendly web application is developed, allowing users to register, login, and upload images or access the camera for live detection.

- **Admin Module**: An administrative module is incorporated for managing user authentication, overseeing model training, and testing, and conducting model analysis.

- **Deployment**: The web application is deployed on a suitable platform to ensure accessibility and scalability.

MODULES DESCRIPTION

1. Data Management Module

- **Functionality**: Manages the collection, augmentation, and preprocessing of the dataset. This includes sourcing images from Kaggle, applying data augmentation techniques to increase the diversity of the training set, and preprocessing images to fit the input requirements of the INCEPTIONV3 model.

- **Components**: Dataset collection, Data augmentation, Preprocessing.

2. Model Training and Validation Module

- **Functionality**: Handles the core development of the deep learning model using the INCEPTIONV3 algorithm. It oversees the training process, including the selection of hyperparameters, the division of data into training and validation sets, and the evaluation of model performance to prevent overfitting.

- **Components**: Model architecture setup, Training loop, Validation, and performance evaluation.

3. Real-time Detection Module

- **Functionality**: Facilitates the real-time detection of facial paralysis from both static images and live video feeds. This module leverages the trained INCEPTIONV3 model to analyse uploaded images or camera streams, identifying signs of facial paralysis with high accuracy.

- **Components**: Image upload, Live video feed processing, Detection algorithm.

4. User Interface (UI) Module

- **Functionality**: Provides the front-end interface for user interaction with the web application. It enables users to register, login, upload images, access the live camera feed, and receive diagnostic feedback. The UI is designed to be intuitive and user-friendly, ensuring ease of use for individuals with varying levels of technical proficiency.

- **Components**: Registration and login system, Image upload interface, Live camera access, Feedback display.

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12 , Issue 2, April 2024

5. Admin and Security Module

- **Functionality**: Offers tools for administrative tasks, including user management, model management, and security measures. It ensures that only authorized users can access the system, provides the admin with the ability to authorize new users, and oversees the training and testing of the model for continual improvement.
- **Components**: User authentication and authorization, Model training and testing oversight, Security protocols.

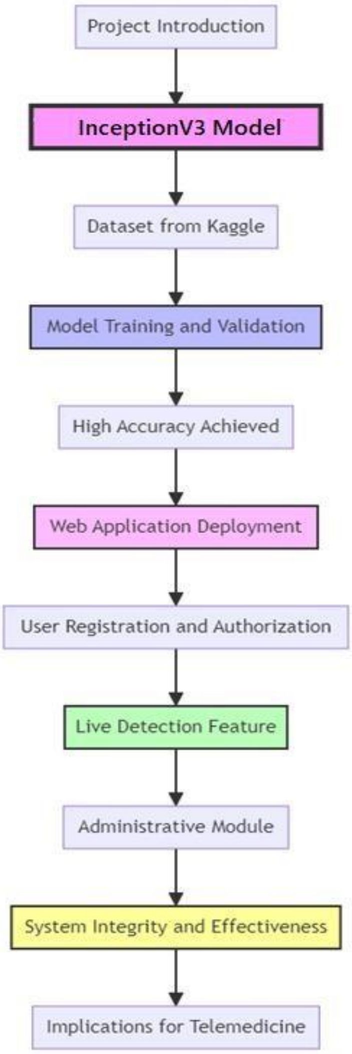


Figure 1: Flow chart

IV. IMPLEMENTATION

CONVOLUTIONAL NEURAL NETWORKS

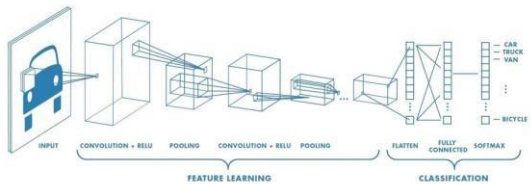
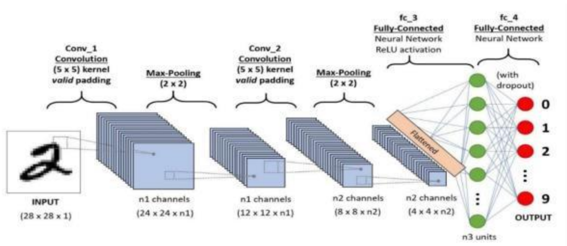


FIG 2: Convolutional neural network

The field of Artificial Intelligence (AI) is rapidly advancing, significantly narrowing the gap between human and machine capabilities. Enthusiasts and researchers are delving into various facets of AI to create remarkable innovations. A prime example of such a domain is Computer Vision. The objective in this area is to empower machines with the ability to interpret the world in a manner akin to human vision, and to utilize this understanding in a range of applications. These include tasks like Image and Video Recognition, Image Analysis and Classification, Media Recreation, and Recommendation Systems, as well as extending into areas like Natural Language Processing. The progress in Computer Vision, particularly through the application of Deep Learning, has been substantial. This progress is largely attributed to the refinement of a key algorithm: the Convolutional Neural Network (CNN), which has evolved and been honed over time.



International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 12 , Issue 2, April 2024

Fig 3: layers of ConNet

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that processes input images by assigning significance to various aspects within them, enabling it to distinguish between different objects. Compared to other classification algorithms, CNNs require significantly less preprocessing. Traditional methods involve manually creating filters, but with sufficient training, CNNs are capable of learning these filters autonomously. The design of a CNN mirrors the neural connectivity patterns found in the human brain, particularly inspired by the structure of the visual cortex. In this system, individual neurons are activated by specific stimuli within a limited area of the visual field, known as the receptive field. These receptive fields overlap to comprehensively cover the entire visual spectrum.

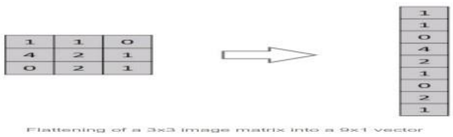


fig 4: flattening of 3x3 image matrix

An image is essentially a matrix composed of pixel values. One might wonder why not simply flatten this matrix and input it into a multi-layer perceptron for classification. The reason lies in the unique capabilities of a Convolutional Neural Network (ConvNet). A ConvNet excels in capturing the spatial and temporal dependencies in an image by applying appropriate filters. Its architecture is more adept at conforming to the image dataset, primarily due to a decrease in the number of parameters and the recyclability of weights. In essence, this means that the network can be more effectively trained to comprehend the complexities inherent in images.

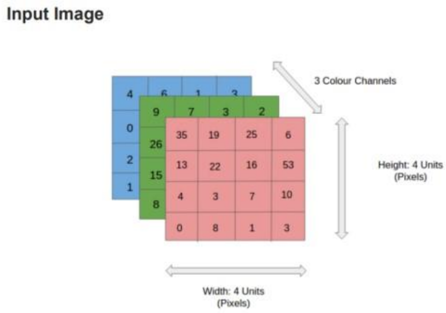


Fig 5: RGB image separated by its colors

In the given illustration, an RGB image is depicted, segmented into its three primary color components: red, green, and blue. Images can exist in various color spaces, such as grayscale and RGB. Imagine the computational load when dealing with high resolution images, such as those in 8K. The function of a Convolutional Neural Network (ConvNet) in this context is to simplify the images into a more manageable form for processing, while preserving essential features crucial for accurate predictions. This aspect is particularly vital in designing an architecture that is not only efficient in feature learning but also capable of scaling to handle extensive datasets.

POOLING LAYER

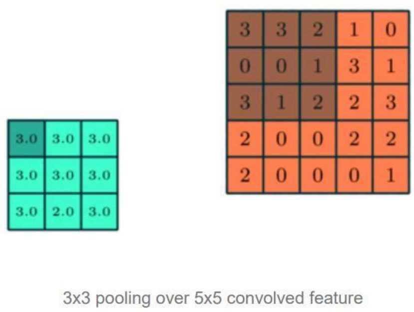


Fig 6: pooling layers

Much like the Convolutional Layer, the role of the Pooling layer in a Convolutional Neural Network

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12 , Issue 2, April 2024

is to reduce the spatial dimensions of the Convolved Feature. This reduction is key in lessening the computational burden by decreasing the data's dimensionality. Additionally, Pooling is instrumental in extracting pivotal features that are invariant to rotation and position, aiding in the efficient training of the model.

Pooling comes in two main forms: Max Pooling and Average Pooling. Max Pooling operates by selecting the maximum value from the image area covered by the Kernel, effectively acting as a noise suppressant by eliminating noisy activations and aiding in denoising as well as reducing dimensionality. Average Pooling, in contrast, computes the average value of all elements in the Kernel's coverage area and primarily focuses on dimensionality reduction as its method of noise suppression. Consequently, Max Pooling is often considered more effective than Average Pooling due to its dual role in noise suppression and dimensionality reduction.

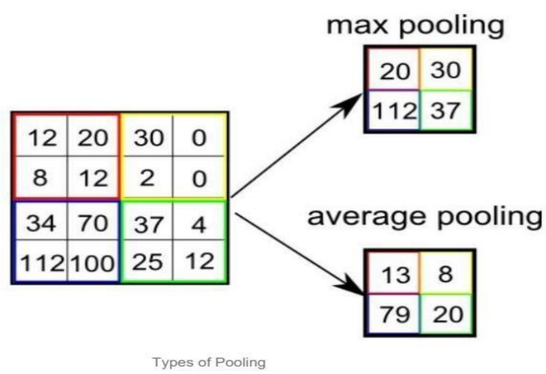


Fig 7: types of pooling layers

The Convolutional Layer and the Pooling Layer collectively constitute the i-th layer in a Convolutional Neural Network's architecture. To capture more nuanced, low-level details in complex images, the network may incorporate additional layers of this kind. However, this enhancement comes with the trade-off of increased computational demands. Once the image data has passed through these layers, the model gains a comprehensive understanding of the image features. Subsequently, the final output is flattened and then input into a standard neural network, which performs the task of classifying the data.

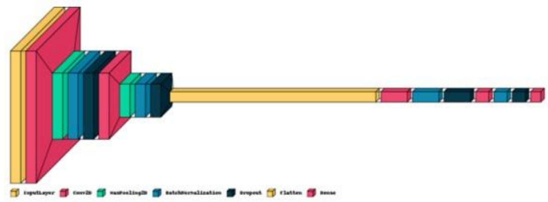


Fig 8: ConvNet layers

V. MATHEMATICAL APPROACH

Inception v3 is a deep convolutional neural network (CNN) architecture designed for image recognition and classification tasks. The architecture includes various types of layers such as convolutional layers, pooling layers, and fully connected layers. Here, I'll provide a high-level overview of the architecture and its mathematical equations:

1.Input:

- The input to Inception V3 is typically a 299x299 RGB image(3 channels).
- Let’s denote the input image as X, where X is a 3D tensor with dimensions (299,299,3).

2. Stem:

The stem consists of basic layers to process the input image.

First, we perform a 2D convolution with stride 2 and ‘SAME’ padding:

- Conv2D(X,32,3 x 3, stride = 2, padding =’SAME’)

Apply Batch Normalization and ReLU activation:

- BatchNorm(Conv2D(…))

Followed by another 2D convolution:

- Conv2D(…,32,3 x3)

And again, Batch Normalization and ReLU:

ISSN 2347–3657

Volume 12 , Issue 2, April 2024

- BachNorm(Conv2D(…))

Finally, a max pooling layer:

- MaxPool(…,3 x 3,stride = 2)

3. Inception Modules:

The core part of Inception networks are the Inception modules. These modules are designed to capture features at various spatial scales. The basic Inception module has parallel branches with different filter sizes.

Let's denote the output of the previous layer as A.

InceptionModule1:

Branch 1:

- A1 = Connv2D(A,64,1 x 1)

Branch 2:

- A2 = Conv2D(A, 96,1 x 1)
- A2 = Conv2D(A2,128,3 x 3)

Branch 3:

- A3 = Conv2D(A, 16,1 x 1)
- A3 = Conv2D(A3,32,5 x 5)

Branch 4:

- A4 = Conv2D(A, 3 x 3)
- A4 = Conv2D(A4,32.1 x 1)

Concatenate along the depth dimension:

- A=Concatenate([A1,A2,A3,A4])

Inception Module 2:

- Similar structure to Module 1 but with different filter size and numbers.

4. Reduction Modules:

These modules are used to reduce the spatial dimensions of the input, thus reducing computational cost.

Reduction Module A:

Branch 1:

- A1 = Conv2D(A,256,1 x 1)
- A1 = Conv2D(A1,384,3 x 3, stride = 2)

Branch 2:

- A2 = Conv2D(A,256,1 x 1)
- A2 = Conv2D(A2,256,3 x 3, stride = 2)

Branch 3:

- A3 = Conv2D(A,256,1 x 1)
- A3 = Conv2D(A3,256,3 x 3)
- A3 = Conv2D(A1,256,3 x 3, stride = 2)

Concatenate along the depth dimension:

- A = Concatenate([A1,A2,A3])

Reduction Module B:

- Similar structure to Reduction Module A but with different filter sizes and numbers.

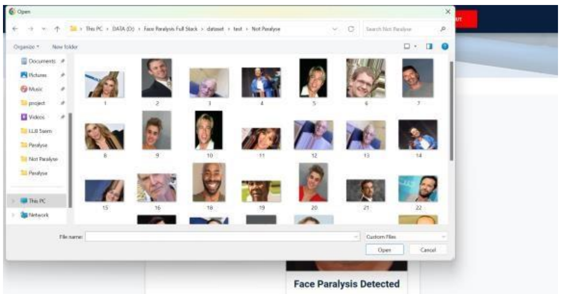
5. Fully Connected Layers:

After several Inception and Reduction modules, the output is passed through fully connected layers for classification.

VI. RESULTS & DISCUSSION

utilizing the InceptionV3 algorithm, to accurately detect facial paralysis from both static images and live video feeds. By harnessing a dataset of 3,000 images from Kaggle, enhanced through data augmentation techniques, the system achieves remarkable detection accuracy, demonstrated by a 95.3% success rate in tests. Integrated into a user-friendly web application, it offers functionalities for user registration, image upload, and live detection, managed through an admin module for user authentication and model oversight. This system not only bridges the gap in accessible diagnostic tools for facial paralysis but also sets a precedent for applying AI in telehealth solutions, promising widespread impact on early detection and patient care.

Sample Input Images for Testing

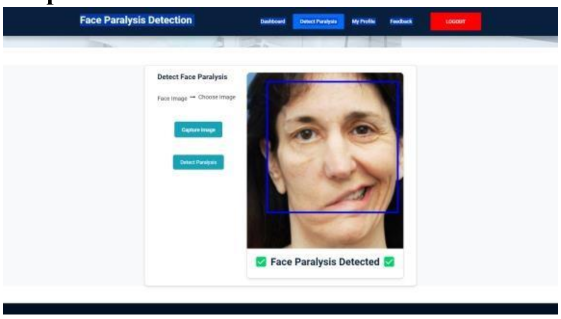


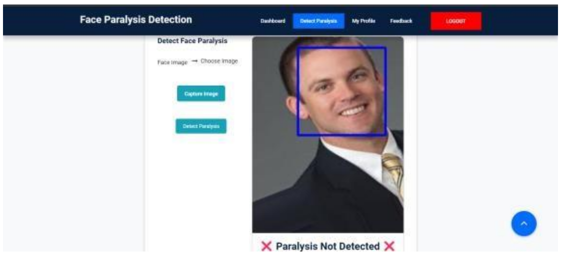
International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12 , Issue 2, April 2024

Output:





MODEL	Accuracy rate
INCEPTIONV3	95.3 %
ALEXNET	93.6%
RESTNET30	91.7%

VII. CONCLUSION

In conclusion, the development of a deep learning model utilizing the INCEPTIONV3 algorithm for the detection of facial paralysis represents a significant leap forward in the application of artificial intelligence in healthcare diagnostics. This project has successfully demonstrated the feasibility and efficiency of using advanced machine learning techniques to address a critical medical diagnosis challenge. By achieving an impressive 95.3% accuracy in detecting facial paralysis from static images and live video feeds, the system sets a new benchmark in the field of automated medical diagnostics.

The deployment of this model within a user-friendly web application significantly enhances the accessibility and convenience of early detection tools for facial paralysis, potentially leading to earlier interventions and better treatment outcomes for affected individuals. The inclusion of an administrative module for user authentication and model management further ensures the integrity and reliability of the system, making it a valuable resource for both patients and healthcare providers.

VIII. FUTURE SCOPE

In moving forward, continuous improvement through user feedback and further model training will be essential to adapt to the evolving landscape of medical diagnostics and technology. This project not only contributes to the advancement of medical technology but also exemplifies the power of AI in making healthcare more accessible, efficient, and patient-centered.

IX. REFERENCE

[1] P Subba Rao. "Applications of machine learning in environmental engineering" JOURNAL OF Science, Technology and Development (2021): 643-648.

[2] P Subba Rao. "Using an Energy Efficient Extended Leach Work with Multilevel Clustering Approach" from JOURNAL of Research Publication and Reviews (2020): 60-64.

[3] C. Jiang, J. Wu, W. Zhong, M. Wei, J. Tong, H. Yu, and L. Wang, "Automatic facial paralysis assessment via computational image analysis," J. Healthcare Eng., vol. 2020, pp. 1–10, Feb. 2020.

[4] D. L. Guarin, Y. Yunusova, B. Taati, J. R. Dusseldorp, S. Mohan, J. Tavares, M. M. van Veen, E. Fortier, T. A. Hadlock, and N. Jowett, "Toward an automatic system for computer aided assessment in facial palsy," Facial Plastic Surg. Aesthetic Med., vol. 22, no. 1, pp. 42–49, Feb. 2020.

[5] J. J. Greene, D. L. Guarin, J. Tavares, E. Fortier, M. Robinson, J. Dusseldorp, O. Quatela, N. Jowett, and T. Hadlock, "The spectrum of facial palsy: The MEEI facial palsy photo and video standard set," Laryngoscope, vol. 130, no. 1, pp. 32–37, Jan. 2020.

[6] Z. Guo, W. Li, J. Dai, J. Xiang, and G. Dan, "Facial imaging and landmark detection technique for objective assessment of unilateral peripheral facial paralysis,' Enterprise Inf. Syst., vol. 16, nos. 10–11, pp. 1556–1572, Oct. 2022.

ISSN
2347- 3657

Certification of Publication

6.2 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

UCC APPROVED

Google Scholar

This Certificate is awarded to

Paradesi Subba Rao

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

In

International Journal of
Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

ISSN
2347- 3657

Certification of Publication

6.2 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

UGC APPROVED

This Certificate is awarded to

N. Rama Devi

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

in

International Journal of
Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

ISSN
2347- 3657

Certification of Publication

Google Scholar

UGC APPROVED

6.2 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

This Certificate is awarded to

Palagulla Lakshmi Priya

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

in

International Journal of Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

ISSN
2347- 3657

Certification of Publication

This Certificate is awarded to

Kristipati Akhila

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

in

International Journal of
Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

ISSN
2347- 3657

Certification of Publication

This Certificate is awarded to

Myla Venkata Ruchitha

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

In

International Journal of
Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

ISSN
2347- 3657

Certification of Publication

This Certificate is awarded to

Shaik Ayesha

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

International Journal of
Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

ISSN
2347- 3657

Certification of Publication

This Certificate is awarded to

Shaik karimun

Department Of Computer Science and Engineering-Data Science, Santhiram Engineering College, Nandyal, AndhraPradesh, 518501, India.

FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICAL INTELLIGENCE

in

International Journal of
Information Technology & Computer Engineering

Volume 12 , Issue 2, April 2024

Editor-In-Chief

DrillBit

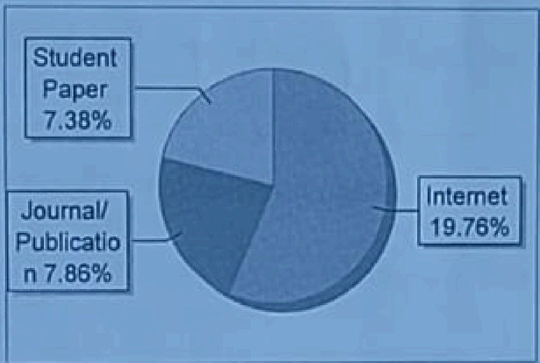
The Report is Generated by DrillBit Plagiarism Detection Software

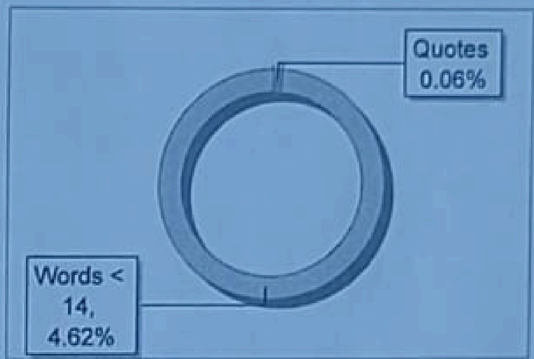
Submission Information

Author Name	Lakshmi Priya
Title	FACIAL PARALYSIS SEVERITY DETECTION USING TRANSFER LEARNING AND ARTIFICIAL INTELLIGENCE
Paper/Submission ID	1671020
Submitted by	principal@srecnandyal.edu.in
Submission Date	2024-04-20 16:12:10
Total Pages	56
Document type	Thesis

Result Information

Similarity **35 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words %	Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

35 SIMILARITY %	**27** MATCHED SOURCES	**B** GRADE	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.softwaretestinghelp.com	6	Internet Data
2	AUDIO-VISUAL SPEECH SEPARATION FROM OVERLAPPED SPEECH FOR THE LRS2 DATASET BY TALAKOTI HINDHUJA YR-2021 , TO JNTUHCE	5	Student Paper
3	osarome.blogspot.com	4	Internet Data
4	Toward an Automatic System for Computer-Aided Assessment in Facial Palsy by Guarin-2020	3	Publication
5	www.slideshare.net	2	Internet Data
6	Automatic Facial Paralysis Assessment via Computational Image Analysis by Jiang-2020	2	Publication
7	moam.info	2	Internet Data
8	www.softwaretestinghelp.com	2	Internet Data
9	www.dx.doi.org	1	Publication
10	dspace.daffodilvarsity.edu.bd 8080	1	Publication
11	HYBRID DRIVER SAFETY MEASUREMENT SYSTEM USING RASPBERRY PI By 16C21D5505 - 2018, JNTUH	1	Student Paper
12	baixardoc.com	1	Internet Data

Submitted to guide
Sowmya

DEAN-R & D and IPR
SANTHIRAM ENGINEERING COLLEGE
NANDYAL-518501, Nandyal (Dt), A.P