

PARKINSON'S DETECTOR SYSTEM USING DEEP NEURAL NETWORK CLASSIFICATION

A PROJECT REPORT

Submitted by

MALLARAPU MOHANA MURALI	211420106138
VAKATI SAI SANTHOSH REDDY	211420106276
VANJIVAKA CHANUKYA	211420106277

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING



PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“PARKINSON’S DETECTOR SYSTEM USING DEEP NEURAL NETWORK CLASSIFICATION”** is the bonafide work of **“MALLARAPU MOHANA MURALI (211420106138), VAKATI SAI SANTHOSH REDDY (211420106276), VANJIVAKA CHANUKYA (211420106277)”** who carried out the project work under my supervision.

SIGNATURE

DR. S.RAJAKUMAR, M.E., Ph.D.,
PROFESSOR,

HEAD OF THE DEPARTMENT

Department of Electronics and
Communication Engineering,
Panimalar Engineering College,
Chennai - 123

SIGNATURE

SATHYABHAMA BADRI NARAYANAN,
M.E., (Ph.D.), ASSISTANT PROFESSOR,

SUPERVISOR,

Department of Electronics and
Communication Engineering,
Panimalar Engineering College,
Chennai - 123

Certified that the above mentioned students were examined in the university project viva-voce held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

A project of this magnitude and nature requires kind cooperation and support from many, for successful completion. We wish to express our sincere thanks to all those who helped us in the completion of this project.

We would like to express our deep gratitude to Our Beloved Secretary and Correspondent, **Dr.P.Chinnadurai M.A., Ph.D.**, for his motivation which inspired us a lot in completing this project work. We also offer our sincere thanks to our dynamic directors **Mrs. C.Vijaya Rajeswari, Dr. C.Sakthi Kumar M.E., Ph.D., and Dr. Saranya Sree Sakthi Kumar B.E., M.B.A, Ph.D.**, for providing us with the necessary facilities and support needed to complete this project.

We also express our appreciation and gratefulness to our principal **Dr. K. Mani M.E., Ph.D.**, who never failed to motivate and encourage us in all our initiative and also helped us in the completion of the project.

We would like to sincerely thank and dedicate this project report to our former HOD **Dr.P.Kannan M.E., Ph.D.**, for being a strong pillar of support and guiding us in the right direction throughout our academic journey and specifically, for his encouraging support in the completion of this project work.

We would like to extend our gratitude and thanks to our Head of the Department **Dr. S.Rajakumar M.E., Ph.D.**, for his valuable suggestions and continuous encouragement throughout the completion of our project.

Our utmost gratitude and thanks to our project supervisor **Mrs. B.Sathyabhama, M.E., (Ph.D)**, Assistant Professor, Department of Electronics and communication engineering for her constant guidance, motivation and valuable suggestions throughout the course of this project work and helping us to complete this project work on time.

We are grateful to our beloved parents for providing us with all the support and the opportunities to complete this project work. We also express our sincere thanks to all our friends and family members for their endless support.

ABSTRACT

This project endeavours to develop an advanced system for the early detection and staging of Parkinson's disease through the analysis of MRI scan images. The primary focus is on creating a robust classification framework capable of accurately distinguishing between MRI images associated with Parkinson's and those without. By harnessing the power of deep learning algorithms, the system not only identifies the presence of Parkinson's disease but also categorizes patients into different stages based on the severity of their symptoms. The proposed system aims to enhance diagnostic accuracy and provide valuable insights into the progression of Parkinson's disease, contributing to more effective and personalized patient care. Importantly, the abstract refrains from explicitly mentioning algorithm names to maintain a succinct and accessible overview of the project's objectives.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 OBJECTIVES	2
2	LITERATURE REVIEW	3
3	PARKINSON’S DISEASE DETECTOR SYSTEM	8
	3.1 SYSTEM ANALYSIS	8
	3.1.1 EXISTING SYSTEM	8
	3.1.2 DRAWBACKS	8
	3.1.3 PROPOSED SYSTEM	9
	3.1.4 ADVANTAGES	9
	3.2 SYSTEM CONFIGURATION	10

	3.3 BLOCK DIAGRAM	10
	3.4 SYMPTOMS OF PARKINSON’S DISEASE	11
	3.5 STAGES OF PARKINSONS DISEASE	11
4	SYSTEM ANALYSIS	14
	4.1 MODULE DESCRIPTION	14
	4.1.1 IMAGE ACQUISITION	14
	4.1.2 IMAGE PRE-PROCESSING	14
	4.1.3 FEATURE EXTRACTION	14
	4.1.4 MRI CLASSIFICATION	15
	4.1.5 CONVOLUTIONAL NEURAL NETWORK	15
	4.1.6 LONG SHORT-TERM MEMORY (LSTM)	16
	4.1.7 STAGE DETECTION MODULE	16
	4.2 PROPOSED SYSTEM ALGORITHM	17
	4.2.1 CNN ALGORITHM	17
	4.3 LSTM ALGORITHM	18
	4.4 UML DIAGRAM	20

	4.5 DATAFLOW DIAGRAM	21
	4.6 SOFTWARE DESCRIPTION	22
	4.6.1 HTML	22
	4.6.2 THE GUI SERVER	22
	4.6.3 PYTHON CODE	22
	4.6.4 TECHNICAL FEASIBILITY	24
	4.6.5 SOCIAL FEASIBILITY	24
5	RESULTS AND DISCUSSIONS	25
	5.1 RESULTS ANALYSIS	25
6	CONCLUSION	28
	FUTURE ENHANCEMENTS	29
	REFERENCES	30
	APPENDIX	35

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	CLASSIFICATION OF STAGES	10
4.1	FRAME DIAGRAM OF LSTM-CNN MODEL	19
4.2	USE CASE DIAGRAM	20
4.3	ACTIVITY DIAGRAM	20
4.4	DATA FLOW DIAGRAM	21
5.1	IDENTIFICATION OF PARKINSON'S DISEASE	25
5.2	MODEL LOSS GRAPH	25
5.3	MODEL ACCURACY GRAPH	26
5.4	IMAGE UPLOADING PAGE	26
5.5	RESULT PAGE	27
5.6	STAGE PREDICTION PAGE	27

LIST OF ABBREVIATIONS

PD	Parkinson's Disease
CNN	Convolutional Neural Network
LSTM	Long short-term memory
RNN	Recurrent Neural Network
SVM	Support Vector Machines
MRI	Magnetic Resonance Imaging
DNN	Deep Neural Network
GUI	Graphical User Interface

CHAPTER 1

INTRODUCTION

The aging of today's society is associated with an increasing number of patients suffering from neurodegenerative disorders. One of these disorders is Parkinson's disease (PD), and current estimates indicate that the number of people with PD will rise more than twofold, from 4 million in 2005 to 9 million by 2030 . The clinical presentations of PD include progressively slowing movements, limb rigidity, restremor, and posture instability. Unfortunately, even those patients who receive dopaminergic treatment or deep brain stimulation still deteriorate with increasing age, and their mortality rate is two- to three-fold higher than that of the general population. Therefore, recognizing PD in its early stage is critical for initiating proper treatments to decrease morbidity and ease the medical burden in the elderly.

The clinical severity of PD can be divided into five stages, called the Hoehn-Yahr Stages I–V . In Stage I, the patients experience unilateral symptoms, such as asymmetrical gait or hand swing; in Stage II, the disease influences are bilateral and the patient's stability degrades; in Stage III, the disease affects the central reflex mechanism, and the patient tends to fall because of trunk instability; in Stage IV, the patient needs a wheelchair and other assistive devices; and in Stage V, the patient is wheelchair bound or even bedridden. Patients with PD can be classified as having early-stage or advanced-stage disease.

In its early stages, denoted in this paper as Early PD and defined as Hoehn-Yahr Stage ≤ 2 , the symptoms include asymmetrical movement reduction of one limb, asymmetrical hand movements, and shuffling when walking, with a preserved posture reflex. In the advanced stages, denoted here as Adv PD and defined as Hoehn-Yahr Stage > 2 , the symptoms are more progressed and include postural

reflex losses, festinating gaits that cause walking instability, and increased risk of falling. However, early detection of PD is challenging because the normal aging population might also exhibit progressive gait slowness, termed senile gait, due to joint osteoarthritis or sarcopenia. Therefore, the aim of the present study was to develop a neural network model that could help physicians recognize the PD gait based on motion characteristics occurring during walking. The model would also facilitate monitoring of the PD disease severity stages for appropriate medication adjustment and intervention. Advances in technology are now improving the early, timely, and accurate diagnosis of PD, especially when machine-learning techniques are applied.

1.1 OBJECTIVES

The project aims to implement a sophisticated framework integrating Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) algorithms. This integrated model goes beyond simple detection, actively categorizing patients into distinct stages based on the severity of their Parkinson's symptoms. The classification system will span stages one through five, providing a nuanced understanding of the disease progression and allowing for tailored interventions and treatments. This multifaceted approach aims to enhance diagnostic precision and contribute to a more comprehensive understanding of Parkinson's disease in a clinical setting.

CHAPTER 2

LITERATURE REVIEW

2.1 Deep Spatial-Temporal Feature Fusion from Adaptive Dynamic Functional Connectivity for MCI Identification

Authors Yang Li, Jingyu Liu, Zhenyu Tang, Baiying Lei

Dynamic functional connectivity (dFC) analysis using resting-state functional Magnetic Resonance Imaging (rs-fMRI) is currently an advanced technique for capturing the dynamic changes of neural activities in brain disease identification. Most existing dFC modeling methods extract dynamic interaction information by using the sliding window-based correlation, whose performance is very sensitive to window parameters. Because few studies can convincingly identify the optimal combination of window parameters, sliding window-based correlation may not be the optimal way to capture the temporal variability of brain activity. In this paper, we propose a novel adaptive dFC model, aided by a deep spatial-temporal feature fusion method, for mild cognitive impairment (MCI) identification. Specifically, we adopt an adaptive Ultra-weighted-lasso recursive least squares algorithm to estimate the adaptive dFC, which effectively alleviates the problem of parameter optimization. Then, we extract temporal and spatial features from the adaptive dFC. In order to generate coarser multi-domain representations for subsequent classification, the temporal and spatial features are further mapped into comprehensive fused features with a deep feature fusion method. Experimental results show that the classification accuracy of our proposed method is reached to 87.7%, which is at least 5.5% improvement than the state-of-the-art methods. These results elucidate the superiority of the proposed method for MCI classification, indicating its effectiveness in the early identification of brain abnormalities.

2.2 Implementation of a Deep Learning Algorithm Based on Vertical Ground Reaction Force Time–Frequency Features for the Detection and Severity Classification of Parkinson’s Disease

Authors Febryan Setiawan and Che-Wei Lin

Conventional approaches to diagnosing Parkinson’s disease (PD) and rating its severity level are based on medical specialists’ clinical assessment of symptoms, which are subjective and can be inaccurate. These techniques are not very reliable, particularly in the early stages of the disease. A novel detection and severity classification algorithm using deep learning approaches was developed in this research to classify the PD severity level based on vertical ground reaction force (vGRF) signals. Different variations in force patterns generated by the irregularity in vGRF signals due to the gait abnormalities of PD patients can indicate their severity. The main purpose of this research is to aid physicians in detecting early stages of PD, planning efficient treatment, and monitoring disease progression. The detection algorithm comprises preprocessing, feature transformation, and classification processes. In preprocessing, the vGRF signal is divided into 10, 15, and 30 s successive time windows. In the feature transformation process, the time domain vGRF signal in windows with varying time lengths is modified into a time–frequency spectrogram using a continuous wavelet transform (CWT). Then, principal component analysis (PCA) is used for feature enhancement. Finally, different types of convolutional neural networks (CNNs) are employed as deep learning classifiers for classification. The algorithm performance was evaluated using k-fold cross-validation (kfoldCV). The best average accuracy of the proposed detection algorithm in classifying the PD severity stage classification was 96.52% using ResNet-50 with vGRF data from the PhysioNet database. The proposed detection algorithm can effectively differentiate gait patterns based on time–

frequency spectrograms of vGRF signals associated with different PD severity levels.

2.3 Parkinson's Disease Detection based on Changes of Emotions during Speech

Authors Justyna Skibinska , Radim Burget

Parkinson's disease (PD) is the neurodegenerative disease which affects 2-3 % of the population beyond 65 years of age in EU. When PD treatment is administered early, it is significantly more effective. Unfortunately, it is quite challenging to detect this disease at its early stage and when the symptoms can be recognized it is usually quite late. For this reason there is big motivation for development more accessible and accurate solutions for the detection of PD. One of the early symptoms is so-called hypomimia. This paper introduces an automatic method, which can objectively detect PD. The method is based on analysis of emotion changes during pronunciation defined speech exercises. We achieved balanced accuracy 69 % using XGBoost algorithm. As the exercise we proposed to use a Czech tongue twister - the difficult to pronounce sentence. The features can be explained and thus it can be used in clinical practice. We identified that the most valuable emotion for PD detection in this case is fear.

2.4 HYPOMIMIA IN PARKINSON'S DISEASE: AN AXIAL SIGN RESPONSIVE TO LEVODOPA

Authors Ricciardi, De Angelis , Marsili , Faiman , Pradhan , Pereira , Edwards, Morgante and Bologna

As digital health technology becomes more pervasive, machine learning (ML) provides a robust way to analyze and interpret the myriad of collected features. The purpose of this preliminary work was to use ML classification to assess the benefits and relevance of neurocognitive features both tablet-based assessments and self-

reported metrics, as they relate to Parkinson's Disease (PD) and its stages [Hoehn and Yahr (H&Y) Stages 1–5]. Further, this work aims to compare perceived versus sensor-based neurocognitive abilities. In this study, 75 participants (n=50 PD; n=25 control) completed 14 tablet-based neurocognitive functional tests (e.g., motor, memory, speech, executive, and multifunction), functional movement assessments (e.g., Berg Balance Scale), and standardized health questionnaires (e.g., PDQ-39). Decision tree classification of sensor-based features allowed for the discrimination of PD from healthy controls with an accuracy of 92.6%, and early and advanced stages of PD with an accuracy of 73.7%; compared to the current gold standard tools [e.g., standardized health questionnaires (78.3% accuracy) and functional movement assessments (70% accuracy)]. Significant features were also identified using decision tree classification. Device magnitude of acceleration was significant in 12 of 14 tests (85.7%), regardless of test type. For classification between diagnosed and control populations, 17 motor (e.g., device magnitude of acceleration), 9 accuracy (e.g., number of correct/incorrect interactions), and 8 timing features (e.g., time to between interactions) were significant. For classification between early (H&Y Stages 1 and 2) and advanced (H&Y Stages 3, 4, and 5) stages of PD, 7 motor, 12 accuracy, and 14 timing features were significant. Finally, this work depicts that perceived functionality of individuals with PD differed from sensor-based functionalities. In early-stage PD was shown to be 21.6% lower than sensor-based scores with notable perceived deficits in memory and executive function. However, individuals in advanced stages had elevated perceptions (1.57x) for executive and behavioral functions compared to early-stage populations. Machine learning in digital health systems allows for a more comprehensive understanding of neurodegenerative diseases and their stages and may also depict new features that influence the ways digital health technology should be configured.

2.5 Deep 1D-Convnet for accurate Parkinson disease detection and severity prediction from gait

Authors Imanne El Maachia , Guillaume-Alexandre Bilodeaua , Wassim Bouachirb

Diagnosing Parkinson’s disease is a complex task that requires the evaluation of several motor and non-motor symptoms. During diagnosis, gait abnormalities are among the important symptoms that physicians should consider. However, gait evaluation is challenging and relies on the expertise and subjectivity of clinicians. In this context, the use of an intelligent gait analysis algorithm may assist physicians in order to facilitate the diagnosis process. This paper proposes a novel intelligent Parkinson detection system based on deep learning techniques to analyze gait information. We used 1D convolutional neural network (1D-Convnet) to build a Deep Neural Network (DNN) classifier. The proposed model processes 18 1D-signals coming from foot sensors measuring the vertical ground reaction force (VGRF). The first part of the network consists of 18 parallel 1D-Convnet corresponding to system inputs. The second part is a fully connected network that connects the concatenated outputs of the 1D-Convnets to obtain a final classification. We tested our algorithm in Parkinson’s detection and in the prediction of the severity of the disease with the Unified Parkinson’s Disease Rating Scale (UPDRS). Our experiments demonstrate the high efficiency of the proposed method in the detection of Parkinson disease based on gait data. The proposed algorithm achieved an accuracy of 98.7%. To our knowledge, this is the state-of-the-art performance in Parkinson’s gait recognition. Furthermore, we achieved an accuracy of 85.3% in Parkinson’s severity prediction. To the best of our knowledge, this is the first algorithm to perform a severity prediction based on the UPDRS. These results show that the model is able to learn intrinsic characteristics from gait data and to generalize to unseen subjects, which could be helpful in a clinical diagnosis.

CHAPTER 3

PARKINSON'S DISEASE DETECTOR SYSTEM

3.1 SYSTEM ANALYSIS

3.1.1 EXISTING SYSTEM

Parkinson's disease is a neurodegenerative disorder that primarily affects movement control. In the context of diagnosing Parkinson's disease from MRI images, the existing system explores a different path by employing traditional machine learning techniques, specifically the Support Vector Machine (SVM). This technique contrasts with the proposed deep learning-based approach, aiming to achieve accurate classification by leveraging handcrafted features and a linear decision boundary. The existing system adopts an alternative approach by utilizing SVM for diagnosing Parkinson's disease from MRI images. While offering an alternative to deep learning, this approach comes with limitations pertaining to feature representation, adaptability, scalability, and the manual nature of feature engineering. These limitations can affect the accuracy and robustness of the classification system, especially when contrasted with the proposed deep learning-based approach. Accurate and early diagnosis of Parkinson's disease is critical, and the choice of methodology plays a significant role in achieving this goal.

3.1.2 DRAWBACKS

- Handcrafted features might miss subtle image details crucial for accurate Parkinson's disease classification.
- SVM struggles with noisy or variable data often found in medical images, affecting classification reliability.
- Existing system's performance suffers when data is scarce or imbalanced, unlike deep learning approaches.

- It requires labeled data only, missing out on the potential of utilizing unlabeled data.

3.1.3 PROPOSED SYSTEM

The proposed system introduces a robust approach for Parkinson's disease detection utilizing MRI scan images. The dataset consists of a varied collection, with images categorized into two classes: those indicative of Parkinson's and those without. The pre-processing stage involves essential steps such as brain cropping, resizing, and label encoding to enhance image quality and facilitate subsequent analysis. To discern the stage of Parkinson's disease, the data is split into training and testing sets, with 80% allocated for training and 20% for testing. The model, a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) algorithms, undergoes deep neural network training for multiple epochs. The classification stage is pivotal, distinguishing between the presence or absence of Parkinson's disease based on learned features. The model identifies symptoms associated with each stage, offering a comprehensive analysis. The stages, ranging from one to five, correspond to the progression of Parkinson's disease. Symptoms encompass tremors, posture abnormalities, walking difficulties, facial expression changes, and overall rigidity, providing a nuanced understanding of the patient's condition.

3.1.4 ADVANTAGES

- It is considered as the best DL technique for image classification due to high accuracy.
- Image pre-processing required is much less compared to other algorithms.
- CNN with combined LSTM has been widely used to solve different problems, but its performance is very good for image processing in health applications.

3.2 SYSTEM CONFIGURATION

3.2.1 H/W SYSTEM CONFIGURATION:

- Processor - Intel
- Speed - 1.1 GHz
- RAM - 8 Gb
- Hard Disk - 500 GB

3.2.2 S/W SYSTEM CONFIGURATION:

- Operating System - Windows 8/10
- Language - Python(3.10) Version.
- Server - GUI.

3.3 BLOCK DIAGRAM

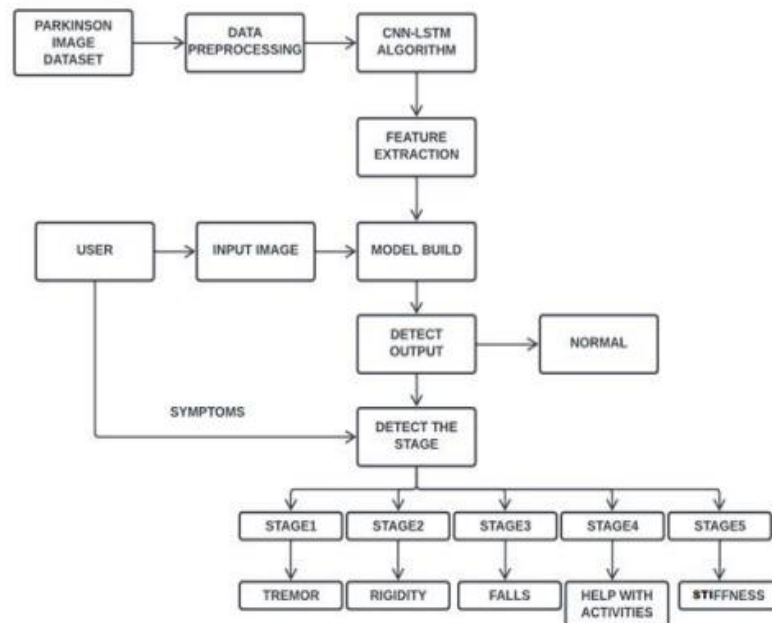


FIG 3.1 CLASSIFICATION OF STAGES

3.4 SYMPTOMS OF PARKINSON'S DISEASE:

The main symptoms of Parkinson's include:

- uncontrollable shaking and tremors
- slowed movement (bradykinesia)
- balance difficulties and eventual problems standing up
- stiffness in limbs

3.5 STAGES OF PARKINSON'S DISEASE:

Many doctors who diagnose this brain disorder rely on the Hoehn and Yahr rating scale to classify the severity of symptoms. The scale is broken into five stages based on disease progression. The five stages help doctors evaluate how far the disease has advanced.

Stage One:

During this initial stage, the person has mild symptoms that generally do not interfere with daily activities. Tremor and other movement symptoms occur on one side of the body only. Changes in posture, walking and facial expressions occur.

Stage 1 is the mildest form of Parkinson's. At this stage, there may be symptoms, but they're not severe enough to interfere with daily tasks and overall lifestyle. In fact, the symptoms are so minimal at this stage that they're often missed. But family and friends may notice changes in your posture, walk, or facial expressions.

A distinct symptom of stage 1 Parkinson's is that tremors and other difficulties in movement are generally exclusive to one side of the body. Prescribed medications can work effectively to minimize and reduce symptoms at this stage.

Stage Two:

Symptoms start getting worse. Tremor, rigidity and other movement symptoms affect both sides of the body or the midline (such as the neck and the trunk). Walking problems and poor posture may be apparent. The person is able to live alone, but daily tasks are more difficult and lengthier. Stage 2 is considered a moderate form of Parkinson's, and the symptoms are much more noticeable than those experienced in stage 1. Stiffness, tremors, and trembling may be more noticeable, and changes in facial expressions can occur.

While muscle stiffness prolongs task completion, stage 2 does not impair balance. Difficulties walking may develop or increase, and the person's posture may start to change. The majority of people with stage 2 Parkinson's can still live alone, though they may find that some tasks take longer to complete. The progression from stage 1 to stage 2 can take months or even years. And there is no way to predict individual progression.

Stage Three:

Stage 3 is the middle stage in Parkinson's, and it marks a major turning point in the progression of the disease. Many of the symptoms are the same as those in stage 2. However, you're now more likely to experience loss of balance and decreased reflexes. Your movements become slower overall. This is why falls become more common in stage 3. Parkinson's significantly affects daily tasks at this stage, but people are still able to complete them. Medication combined with occupational therapy may help decrease symptoms.

Considered mid-stage, loss of balance (such as unsteadiness as the person turns or when he/she is pushed from standing) is the hallmark. Falls are more common. Motor symptoms continue to worsen. Functionally the person is somewhat restricted

in his/her daily activities now, but is still physically capable of leading an independent life. Disability is mild to moderate at this stage.

Stage Four:

Independence separates people with stage 3 Parkinson's from those with stage 4. During stage 4, it's possible to stand without assistance. However, movement may require a walker or other type of assistive device. Many people are unable to live alone at this stage of Parkinson's because of significant decreases in movement and reaction times. Living alone at stage 4 or later may make many daily tasks impossible, and it can be dangerous.

At this point, symptoms are fully developed and severely disabling. The person is still able to walk and stand without assistance, but may need to ambulate with a cane/walker for safety. The person needs significant help with activities of daily living and is unable to live alone.

Stage Five:

This is the most advanced and debilitating stage. Stiffness in the legs may make it impossible to stand or walk. The person is bedridden or confined to a wheelchair unless aided. Around-the-clock care is required for all activities. Stage 5 is the most advanced stage of Parkinson's disease. Advanced stiffness in the legs can also cause freezing upon standing, making it impossible to stand or walk.

People in this stage require wheelchairs, and they're often unable to stand on their own without falling. Around-the-clock assistance is required to prevent falls. Up to 50 percent Trusted Source of people at stages 4 and 5 experience confusion, hallucinations, and delusions. Hallucinations occur when you see things that aren't there. Delusions happen when you believe things that aren't true, even when you have been presented with evidence that your belief is wrong.

CHAPTER 4

SYSTEM ANALYSIS

4.1 MODULE DISCRPTION

4.1.1 IMAGE ACQUISITION:

The Parkinson's disease MRIs dataset acquisition has been used to implement the proposed methods. This method is used to design for extraction of Parkinson's with accuracy and composed number of stages is including image capturing, edge detection, and classification of Parkinson's disease.

4.1.2 IMAGE PRE-PROCESSING:

In this module, we are performing some basic operation on image to get proper image for processing. In this module, we are perform certain operation like gray-scale conversion, filtering, sharpening, smoothing, edging, and image segmentation to get proper and clean image. Preprocessing step enhances the quality of the images by eliminating noise. The Gray scale images, kind of black-and-white or gray monochrome images, are composed exclusively of shades of gray. Gray scale images can be measuring the intensity of light at each pixel. The Filtering operation is performed on the image to increase the smoothness, sharpness as well as edge enhancement. Sharpening filter is used to enhancement the images in sharpening and to enhance detail that has been blurred. Smoothing filter is used to reduce the noise. It has used many different algorithms. Edging is a technique of finding and identifying sharpness presented in an image.

4.1.3 FEATURE EXTRACTION:

In this module, we are performing some more operation on segmented image. In this module we will perform feature extraction operation to get all detailed information

about brain image. Feature Extraction and reduction has been playing a vital role for Parkinson's disease region into their relevant categories in the field of computer vision and machine learning. The major issue behind feature extraction is to compute the most active or robust features for classification, which produced an efficient performance. The Feature extraction is used related to dimensionality reduction.

4.1.4 MRI CLASSIFICATION:

In this module, we are performing classification techniques with help of deep learning algorithm to determine Parkinson disease condition. The Parkinson's disease classification is the final step of the proposed approach that is used to identify the type of Parkinson's disease normal or abnormal. After features are extracted and selected the classification step using CNN is performed on the resulted feature vector. Classification is performed by using training phase and testing phase of CNN structure.

4.1.5 CONVOLUTIONAL NEURAL NETWORK:

The name of "Convolutional Neural Network" performs the mathematical operation called convolution. Convolution is a specialized kind of linear operation. In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, Convolutional networks are simply neural networks that use convolution in matrix multiplication in at least one of their layers. ConvNets have been successful in Identifying faces, objects, and diseases detection. A convolutional neural network consists of an input layer, output layer, as well as multiple hidden layers. CNN which is feed forward neural network and is widely used for image recognition and classification. The Convolutional neural layers convolve the input and pass its result output to the next layer. CNNs are regularized versions of multilayer perceptron's. The Multilayer perceptron's are the fully connected

networks each one neuron is connected to all other neurons in next layer. The "fully-connected" network means over fitting data.

4.1.6 LONG SHORT-TERM MEMORY (LSTM):

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture designed to address the vanishing gradient problem in traditional RNNs, making it well-suited for sequential data tasks such as natural language processing and time series prediction. LSTMs utilize a system of specialized gates (input, forget, and output) to control the flow of information through the network, enabling it to capture and remember long-range dependencies in the data while avoiding issues like gradient instability. This makes LSTMs particularly effective for tasks where context over time is crucial, as they can model and retain important information over extended sequences, making them a popular choice in various machine learning applications.

4.1.7 STAGE DETECTION MODULE:

Symptoms considered for classification encompass a spectrum, including tremors, posture abnormalities, walking difficulties, facial expression changes, and overall rigidity. Through a meticulous analysis of these symptoms, the system aims to assign patients to predefined stages, ranging from the initial to advanced stages. This categorization facilitates a comprehensive understanding of the disease's progression, empowering healthcare professionals with valuable insights for tailored treatment strategies. By adopting a symptom-based approach, the Stage Detection module enhances the diagnostic capabilities of the system, allowing for a nuanced and personalized assessment of Parkinson's disease severity in individual patients.

4.2 PROPOSED SYSTEM ALGORITHM:

4.2.1 CNN ALGORITHM:

4.2.1.1 Why CNN for Image Classification?

Image classification involves the extraction of features from the image to observe some patterns in the dataset. Using an ANN for the purpose of image classification would end up being very costly in terms of computation since the trainable parameters become extremely large.

For example, if we have a 50 X 50 image of a cat, and we want to train our traditional ANN on that image to classify it into a dog or a cat the trainable parameters become $(50 \times 50) \times 100$ image pixels multiplied by hidden layer + 100 bias + 2×100 output neurons + 2 bias = 2,50,30

4.2.1.2 Examples of different filters and their effects

Filters help us exploit the spatial locality of a particular image by enforcing a local connectivity pattern between neurons.

Convolution basically means a pointwise multiplication of two functions to produce a third function. Here one function is our image pixels matrix and another is our filter. We slide the filter over the image and get the dot product of the two matrices. The resulting matrix is called an “Activation Map” or “Feature Map”.

Step 1: Choose a Dataset

Choose a dataset of your interest or you can also create your own image dataset for solving your own image classification problem. An easy place to choose a dataset is on kaggle.com. The dataset I’m going with can be found here. This dataset contains 12,500 augmented images of blood cells (JPEG) with accompanying cell type labels (CSV). There are approximately 3,000 images for each of 4 different cell types grouped into 4 different folders (according to cell type). The cell types are

Eosinophil, Lymphocyte, Monocyte, and Neutrophil. Here are all the libraries that we would require and the code for importing them.

Step 2: Prepare Dataset for Training

Preparing our dataset for training will involve assigning paths and creating categories(labels), resizing our images. Resizing images into 200 X 200

Step 3: Create Training Data

Training is an array that will contain image pixel values and the index at which the image in the CATEGORIES list.

Step 4: Shuffle the Dataset

Step 5: Assigning Labels and Features

This shape of both the lists will be used in Classification using the NEURAL NETWORKS.

Step 6: Normalising X and converting labels to categorical data

Step 7: Split X and Y for use in CNN

Step 8: Define, compile and train the CNN Model

Step 9: Accuracy and Score of model

4.3 LSTM ALGORITHM:

To solve the problem of Vanishing and Exploding Gradients in a Deep Recurrent Neural Network, many variations were developed. One of the most famous of them is the Long Short Term Memory Network(LSTM). In concept, an LSTM recurrent unit tries to “remember” all the past knowledge that the network is seen so far and to “forget” irrelevant data. This is done by introducing different activation function

layers called “gates” for different purposes. Each LSTM recurrent unit also maintains a vector called the Internal Cell State which conceptually,

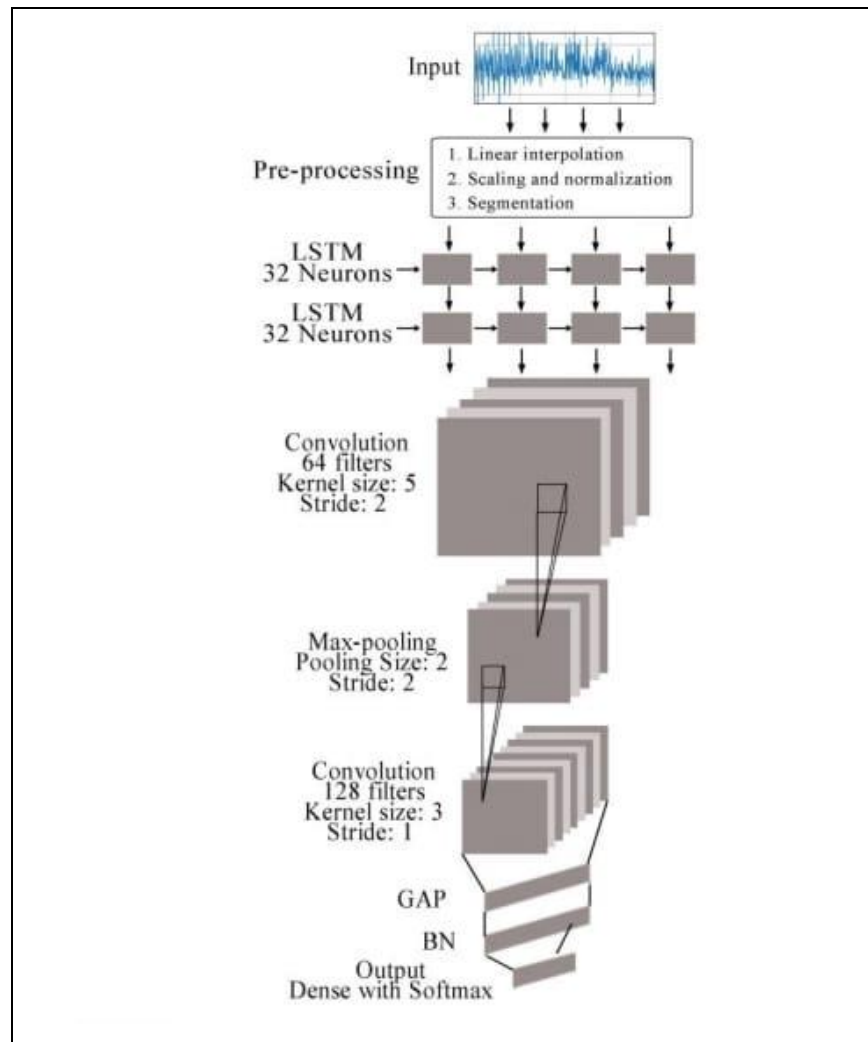


FIG 4.1 FRAME DIAGRAM OF LSTM-CNN MODEL

LSTMs utilize a system of specialized gates (input, forget, and output) for the information flow control through the connection, activating it to capture and remember long-term dependencies in the information while avoiding issues like gradient instability. This makes LSTMs particularly effective for tasks where context over time is crucial, as they can model and retain important information over extended sequences, making them a popular choice in various machine learning applications.

4.4 UML DIAGRAM:

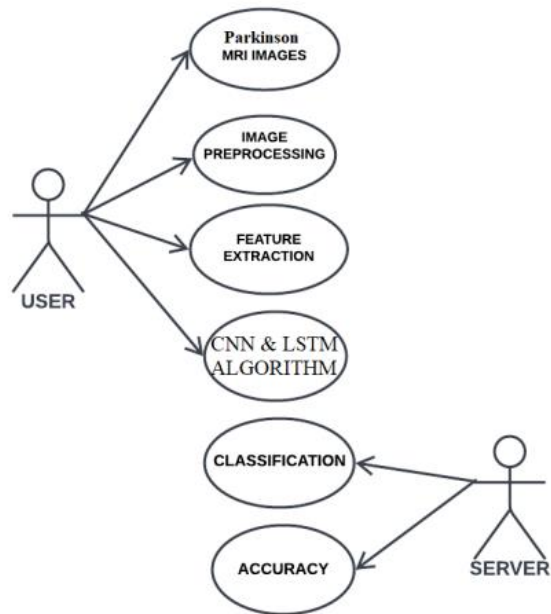


FIG 4.2 USE CASE DIAGRAM

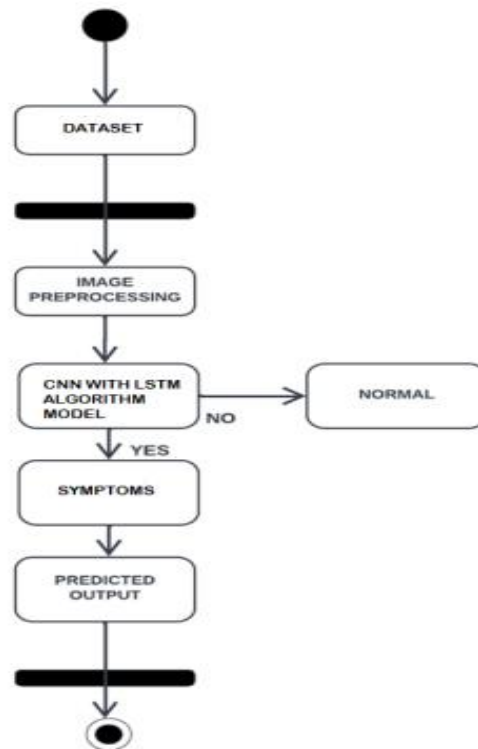


FIG 4.3 ACTIVITY DIAGRAM

4.5 DATA FLOW DIAGRAM

LEVEL 0:



LEVEL 1:



LEVEL 2:



OVERALL DIAGRAM:

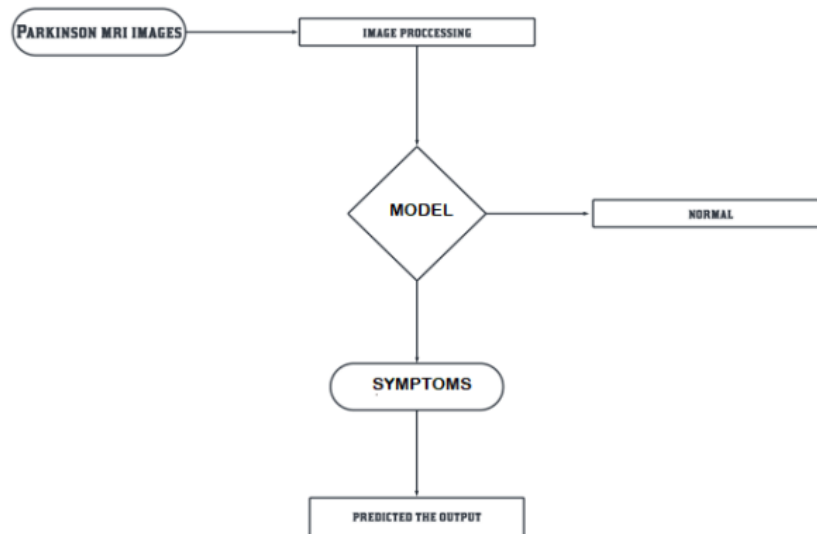


FIG 4.4 DATA FLOW DIAGRAM

4.6 SOFTWARE DESCRIPTION:

4.6.1. HTML:

HTML (HyperText Markup Language) is a standard language used to structure web pages and their content. It's the foundation of web development, providing structure and framework for web content. HTML allows you to define the layout, headings, paragraphs, links, images, and other things, and it works with CSS and JavaScript to create the websites we see while browsing.

HTML allows you to add images, videos, and other multimedia content to your pages, and it provides ways to control how your content is displayed and interacted with by users. For example, you can structure content within a set of paragraphs, a list of bulleted points, or using images and data tables.

4.6.2. The GUI Server

The GUI Server is responsible for generating the dynamic web-based user interface of Netcool/Impact. It brokers requests between users web browsers and Netcool/Impact. Then, it returns the graphical views that you use to work with the data model, services, and policies.

The WebSphere Application Server Liberty Core is installed and configured during the installation if you chose to install the GUI Server as one of the deployment components. The installer sets all the default configuration properties for the server. After the installation, you can change the configuration of the GUI Server by editing its properties files.

4.6.3. Python code:

Python is a high-level object-oriented programming language that was created by Guido van Rossum. It is also called general-purpose programming language as it is used in almost every domain we can think of as mentioned below:

- Web Development
- Software Development
- Game Development
- AI & ML
- Data Analytics

Python packages used for Parkinson disease prediction are:

- 1.flask
- 2.tensorflow
- 3.numpy
- 4.pillow

4.6.3.1. FLASK:

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

4.6.3.2. TENSORFLOW:

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains.

4.6.3.3. Numpy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software.

4.6.4 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.6.5 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 RESULTS ANALYSIS

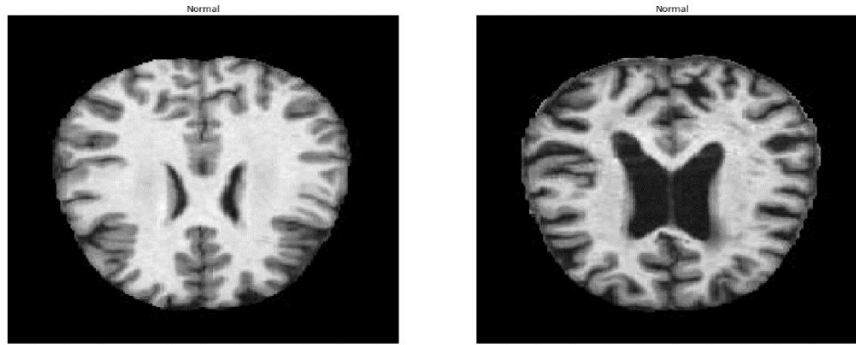


FIG 5.1 IDENTIFICATION OF PARKINSON'S DISEASE

The above MRI images are the classified images of trained data according to the stages of Parkinson's disease.

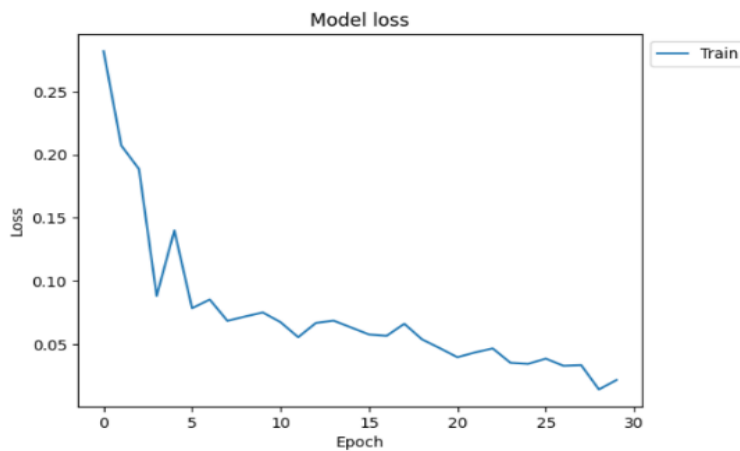


FIG 5.2 MODEL LOSS GRAPH

The Model Loss graph represents the graphical representation that shows how the loss function's value changes over successive epochs during the training process.

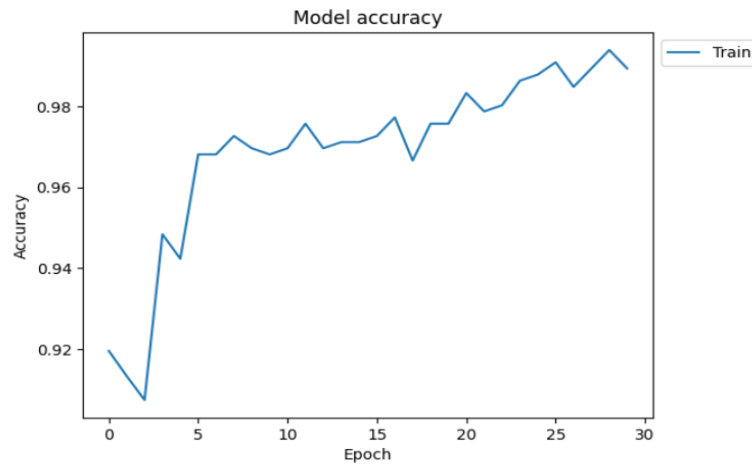


FIG 5.3 MODEL ACCURACY GRAPH

The Model Accuracy graph is the graphical representation that shows how the accuracy of the model changes over successive epochs during the training process.

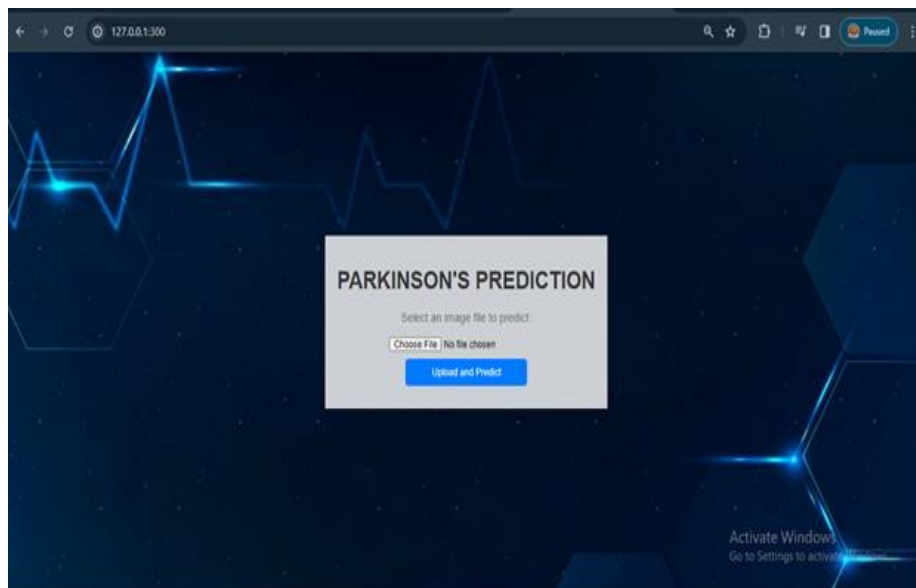


FIG 5.4 INPUT PAGE

After executing, the page opens as shown above. The user has to provide input image that needs to be predicted.

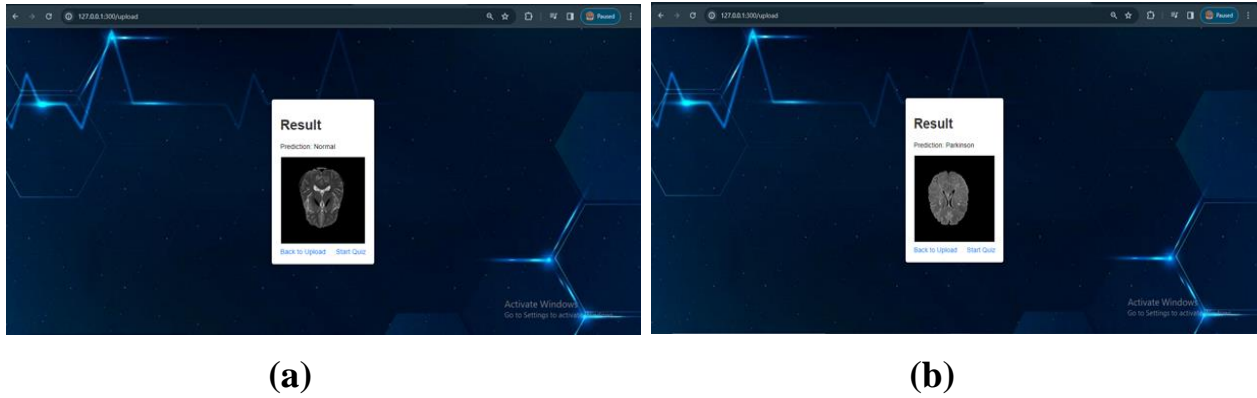


FIG 5.5 RESULT PAGE

The above figures depicts that the image we uploaded will be analyzed using the LSTM-CNN algorithm method. After analysis, it classifies the condition and provides result about whether it is normal or abnormal. In the above figure, (a) shows that the input provided has a Parkinson's condition. While (b) shows that the input image provided is normal and has symptoms of Parkinson's.

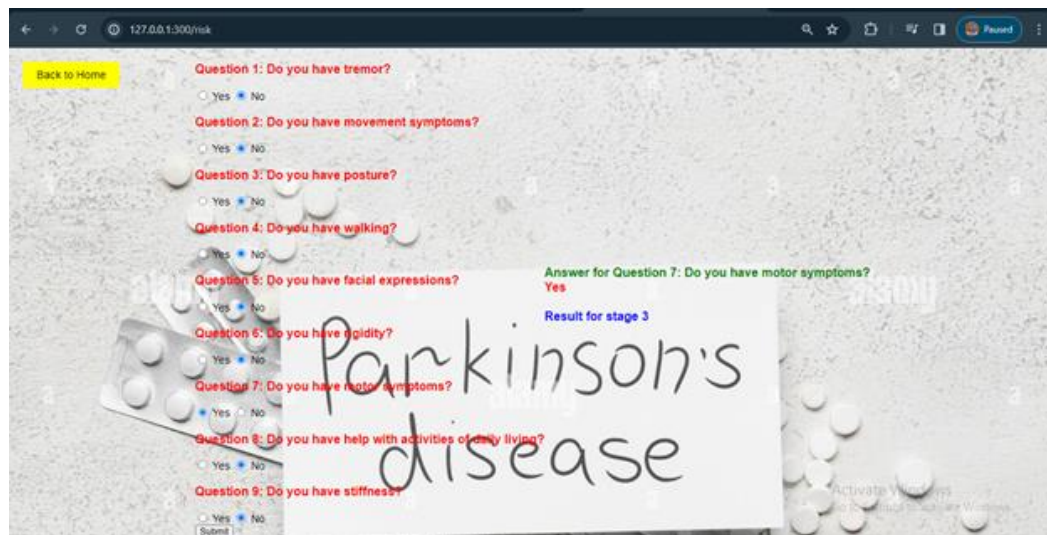


FIG 5.6 STAGE PREDICTION PAGE

After successful completion of the analysis, we can start a personal analysis by selecting the Start Quiz option. Based on the symptoms we have and the questions we answer, the CNN model predicts the stage of Parkinson's condition. The above figure is the final result page of the person having stage 3 of Parkinson's disease.

CHAPTER 6

CONCLUSION

In this study, we used transfer learning to develop a CNN model for automatic Parkinson's disease detection using MR images. Transfer learning uses weights from networks previously trained on millions of data. The proposed study implements four different transfer learning models with different optimizers (ADAM, SGD, RMSprop), and extensive experiments were performed on the two datasets with the largest number of MR images currently available. For these four models, the features are extracted using transfer learning, and three dense layers along with the softmax layer are used for classification purposes. The proposed deep TL models shows fast learning by using the Adam optimizer, and the dropout method avoids the problem of overfitting. In future work, the performance of the system can still be improved by using larger data sets and using other deep learning techniques.

FUTURE ENHANCEMENTS

Although this study focused on five other convolutional models and transfer learning designs for Parkinson's disease in the medical imaging field, further research is needed. We will investigate more significant and influential deep CNN models for Parkinson disease classification and conduct segmentation with reduced time complexity in future approaches. Also, to improve the accuracy of the proposed model, we will increase the number of MRI scans in the dataset used for this study. Furthermore, we will also be applying the proposed approach to other medical images such as x-ray, computed tomography (CT), and ultrasound which may serve as a foundation for future research.

REFERENCES

- [1] A. Agarwal, S. Chandrayan and S. S. Sahu, (2016) "Prediction of Parkinson's disease using speech signal with Extreme Learning Machine," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, pp. 3776-3779
- [2] Alzubaidi, Mahmood Saleh, et al. (2012) "The role of neural network for the detection of Parkinson's disease: a scoping review." *Healthcare*. Vol. 9. No. 6. MDPI.
- [3] Babu, Giduthuri Sateesh, Sundaram Suresh, and Belathur Suresh Mahanand. (2014) "A novel PBL-McRBFN-RFE approach for identification of critical brain regions responsible for Parkinson's disease." *Expert Systems with Applications* 41.2: 478-488.
- [4] Camacho, Milton, et al. (2023) "Explainable classification of Parkinson's disease using deep learning trained on a large multi-center database of T1-weighted MRI datasets." *NeuroImage: Clinical* 38: 103405.
- [5] Cui, Xinchun, et al. (2023) "An adaptive weighted attention-enhanced deep convolutional neural network for classification of MRI images of Parkinson's disease." *Journal of Neuroscience Methods* 394: 109884.
- [6] E. Adeli et al., (2019) "Semi-supervised discriminative classification robust to sample-outliers and feature-noises," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 515–522.
- [7] El Maachi, I.; Bilodeau, G.-A.; Bouachir, W. (2020) Deep 1D-Convnet for accurate Parkinson disease detection and severity prediction from gait. *Expert Syst. Appl.*, 143, 113075, doi:10.1016/j.eswa.2019.113075.

- [8] F. Setiawan and C.-W. Lin, (2021) "Implementation of a deep learning algorithm based on vertical ground reaction force time–frequency features for the detection and severity classification of Parkinson’s disease,” *Sensors*, vol. 21, no. 15, p. 5207.
- [9] Grover, Srishti, et al. (2018) "Predicting severity of Parkinson’s disease using deep learning." *Procedia computer science* 132: 1788-1794.
- [10] H. Lei et al., (2019) “Parkinson’s disease diagnosis via joint learning from multiple modalities and relations,” *IEEE J. Biomed. Health Inform.*, vol. 23, no. 4, pp. 1437–1449.
- [11] J. Shi et al., (2019) “Cascaded multi-column RVFL+ classifier for single-modal neuroimagingbased diagnosis of Parkinson’s disease,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 8, pp. 2362– 2371.
- [12] Joy, Md Ashif Mahmud, et al. (2023) "Automated Parkinson’s Disease Detection from Brain MRI Images Using Deep Convolutional Neural Network." 2023 26th International Conference on Computer and Information Technology (ICCIT). IEEE.
- [13] Justyna Skibińska, Radim Burget, (2020) “Parkinson’s Disease Detection based on Changes of Emotions during Speech”, Brno University of Technology Brno, Czech Republic, 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)
- [14] Khanna, Ketna, Sapna Gambhir, and Mohit Gambhir. (2023) "A novel technique for classifying parkinson’s disease using structural mri scans." *Multimedia Tools and Applications* 82.29: 46011-46036.
- [15] L. Ricciardi, A. De Angelis, L. Marsili, I. Faiman, P. Pradhan, E. Pereira, M. Edwards, F. Morgante, and M. Bologna, (2020) “Hypomimia in Parkinson’s disease: an axial sign responsive to levodopa,” *European Journal of Neurology*.

- [16] Lakshmi, T.S., Ramani, B.L., Jayana, R.K., Kaza, S., Kamatam, S.S.S.T., Raghava, B. (2023). An Ensemble Model to Detect Parkinson's Disease Using MRI Images. In: Bhateja, V., Sunitha, K.V.N., Chen, YW., Zhang, YD. (eds) Intelligent System Design. Lecture Notes in Networks and Systems, vol 494. Springer, Singapore.
- [17] M. Ariz et al., (2019) "Dynamic atlas-based segmentation and quantification of neuromelaninrich brainstem structures in Parkinson disease," IEEE Trans. Med. Imag., vol. 38, no. 3, pp. 813–823.
- [18] Manimegalai, R., et al. (2022) "Deep Learning Based Approach for Identification of Parkinson's Syndrome." 2022 International Conference on Intelligent Innovations in Engineering and Technology (ICIET). IEEE.
- [19] Mishra, Rishik, et al. (2022) "A deep learning approach for the early diagnosis of Parkinson's disease using brain MRI scans." International Journal of Applied Pattern Recognition 7.1: 64-77.
- [20] N. Tuovinen et al., (2018) "The reorganization of functional architecture in the early-stages of Parkinson's disease," Parkinsonism Rel. Disorders, vol. 50, pp. 61–68.
- [21] Nilashi, Mehrbakhsh, et al. (2023) "Early diagnosis of Parkinson's disease: A combined method using deep learning and neuro-fuzzy techniques." Computational biology and chemistry 102: 107788.
- [22] Oh, Shu Lih, et al. (2020) "A deep learning approach for Parkinson's disease diagnosis from EEG signals." Neural Computing and Applications 32: 10927-10933.

- [23] Pahuja, Gunjan, and Bhanu Prasad. (2022) "Deep learning architectures for Parkinson's disease detection by using multi-modal features." *Computers in Biology and Medicine* 146: 105610.
- [24] Peker, Musa, Baha Şen, and Dursun Delen. (2015) "Computer-aided diagnosis of Parkinson's disease using complexvalued neural networks and mRMR feature selection algorithm." *Journal of healthcare engineering* 6.3.
- [25] Singh, Anshul, et al. (2017) "Analysis and Identification of Parkinson disease based on fMRI." *Int. J. Electron. Electr. Comput. Syst. IJEECS* 6.1: 201-205.
- [26] Sivaranjini, S., and C. M. Sujatha. (2020) "Deep learning based diagnosis of Parkinson's disease using convolutional neural network." *Multimedia tools and applications* 79.21: 15467-15479.
- [27] Sreelakshmi, S., and Robert Mathew. (2022) "A Hybrid Approach for Classifying Parkinson's Disease from Brain MRI." *Proceedings of International Conference on Information Technology and Applications: ICITA 2021*. Singapore: Springer Nature Singapore.
- [28] Tuncer, Turker, Sengul Dogan, and Udyavara Rajendra Acharya. (2020) "Automated detection of Parkinson's disease using minimum average maximum tree and singular value decomposition method with vowels." *Biocybernetics and Biomedical Engineering* 40.1: 211-220.
- [29] Y. Li, J. Liu, Z. Tang, and B. Lei, (2020) "Deep spatial-temporal feature fusion from adaptive dynamic functional connectivity for MCI identification," *IEEE Trans. Med. Image.*, vol. 39, no. 9, pp. 2818–2830.

[30] Zhang, He-Hua, et al. (2016) "Classification of Parkinson's disease utilizing multi-edit nearest-neighbor and ensemble learning algorithms with speech samples." Biomedical engineering online 15: 1-22.

APPENDIX

SOURCE CODE

TRAINING DATASET CODE:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import skimage.io

import os

import tqdm

import glob

import tensorflow

from keras.utils import to_categorical

from tqdm import tqdm

from sklearn.utils import shuffle

from sklearn.model_selection import train_test_split

from skimage.io import imread, imshow

from skimage.transform import resize

#from skimage.color import grey2rgb
```

```

from skimage.color import rgb2gray

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import InputLayer, BatchNormalization, Dropout,
Flatten, Dense, Activation, MaxPool2D, Conv2D

from keras.models import model_from_json

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

from tensorflow.keras.applications.densenet import DenseNet169

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


batch_size = 64

# All images will be rescaled by 1./255

train_datagen = ImageDataGenerator(rescale=1/255)

# Flow training images in batches of 128 using train_datagen generator

train_generator = train_datagen.flow_from_directory(

    'dataset', # This is the source directory for training images

    target_size=(200, 200), # All images will be resized to 200 x 200

    batch_size=batch_size,

```

```

# Specify the classes explicitly

classes = ['Normal','Parkinson'],

# Since we use categorical_crossentropy loss, we need categorical labels

class_mode='categorical')

valid_datagen = ImageDataGenerator(rescale = 1./255,

                                   validation_split = 0.2)

test_datagen = ImageDataGenerator(rescale = 1./255)

fig, ax = plt.subplots(nrows = 1, ncols = 2, figsize=(20,20))

for i in tqdm(range(0,2)):

    rand1 = np.random.randint(len(train_generator))

    rand2 = np.random.randint(50)

    ax[i].imshow(train_generator[rand1][0][rand2])

    ax[i].axis('off')

    a = train_generator[rand1][1][rand2]

    if a[0] == 1:

        ax[i].set_title('Normal')

    elif a[1] == 1:

        ax[i].set_title('Parkinson')

```

```

import tensorflow as tf

model = tf.keras.models.Sequential([

    # Note the input shape is the desired size of the image 200x 200 with 3 bytes
    color

    # The first convolution

    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200, 200, 3)),

    tf.keras.layers.MaxPooling2D(2, 2),

    # The second convolution

    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2,2),

    # The third convolution

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2,2),

    # The fourth convolution

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2,2),

    # The fifth convolution

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),

```

```

tf.keras.layers.MaxPooling2D(2,2),

# Flatten the results to feed into a dense layer

tf.keras.layers.Flatten(),

# 128 neuron in the fully-connected layer

tf.keras.layers.Dense(128, activation='relu'),

# 5 output neurons for 5 classes with the softmax activation

tf.keras.layers.Dense(5, activation='softmax')

])

model.summary()


from tensorflow.keras.optimizers import RMSprop

model.compile(loss='categorical_crossentropy',

              optimizer=RMSprop(lr=0.001),

              metrics=['acc'])

total_sample=train_generator.n

n_epochs = 30


history = model.fit_generator(

    train_generator,

    steps_per_epoch=int(total_sample/batch_size),

```



```
epochs=n_epochs,

verbose=1)

model.save('model2.h5')

# Plot the loss over epochs

plt.plot(history.history['loss'])

plt.title('Model loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.legend(['Train'], loc='upper left', bbox_to_anchor=(1,1))

plt.show()

plt.plot(history.history['acc'])

plt.title('Model accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.legend(['Train'], loc='upper left', bbox_to_anchor=(1,1))

plt.show()
```

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Questionnaire</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-image: url('static/i.jpg');
    background-size: cover; /* Make the background image cover the entire
viewport */
    background-repeat: no-repeat; /* Prevent background image from repeating */
    background-attachment: fixed; /* Keep the background image fixed while
scrolling */
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }

  /* Style for questions */
  p.question {
    color: red; /* Change color to your preferred color */
```

```

    font-weight: bold; /* Make the text bold */
    font-size: 18px ;
}

/* Style for result */
#result {
    color:red; /* Change color to your preferred color */
    font-weight: bold; /* Make the text bold */
    font-size: 18px; /* Change font size to your preferred size */
}

.back-button {
    background-color: yellow;
    color: black;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    position: absolute; /* Position button absolutely */
    top: 20px; /* Adjust top position as needed */
    left: 20px; /* Adjust left position as needed */
    text-decoration: none;
}

.back-button:hover {
    background-color: lightgreen;
}

```

</style>

</head>

<body>

<form id="questionForm" onsubmit="return false;">

<p class="question">Question 1: Do you have tremor?</p>

<input type="radio" name="q1" value="yes"> Yes

<input type="radio" name="q1" value="no"> No

<p class="question">Question 2: Do you have movement symptoms?</p>

<input type="radio" name="q2" value="yes"> Yes

<input type="radio" name="q2" value="no"> No

<p class="question">Question 3: Do you have posture?</p>

<input type="radio" name="q3" value="yes"> Yes

<input type="radio" name="q3" value="no"> No

<p class="question">Question 4: Do you have walking?</p>

<input type="radio" name="q4" value="yes"> Yes

<input type="radio" name="q4" value="no"> No

<p class="question">Question 5: Do you have facial expressions?</p>

<input type="radio" name="q5" value="yes"> Yes

<input type="radio" name="q5" value="no"> No

<p class="question">Question 6: Do you have rigidity?</p>

<input type="radio" name="q6" value="yes"> Yes

```
<input type="radio" name="q6" value="no"> No<br>
```

```
<p class="question">Question 7: Do you have motor symptoms?</p>
```

```
<input type="radio" name="q7" value="yes"> Yes
```

```
<input type="radio" name="q7" value="no"> No<br>
```

```
<p class="question">Question 8: Do you have help with activities of daily  
living?</p>
```

```
<input type="radio" name="q8" value="yes"> Yes
```

```
<input type="radio" name="q8" value="no"> No<br>
```

```
<p class="question">Question 9: Do you have stiffness?</p>
```

```
<input type="radio" name="q9" value="yes"> Yes
```

```
<input type="radio" name="q9" value="no"> No<br>
```

```
<button type="submit" onclick="showResult()">Submit</button>
```

```
<a href="{{ url_for('index') }}" class="back-button">Back to Home</a></div>
```

```
</form>
```

```
<div id="result"></div>
```

```
<script>
```

```
function showResult() {
```

```
    var stage = 1;
```

```
    // Check which question's "yes" button is clicked last to determine the stage
```

```
    for (var i = 9; i >= 1; i--) {
```

```

var answer = document.querySelector('input[name="q' + i + '"]:checked');
if (answer && answer.value === "yes") {
    stage = i;
    break;
}
}

// Display result based on stage
var resultDiv = document.getElementById("result");
switch (stage) {
    case 1:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 1 Symptoms:</span><br>Tremor<br>";
        break;
    case 2:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 1 Symptoms:</span><br>Movement Symptoms<br>";
        break;
    case 3:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 1 Symptoms:</span><br>Posture<br>";
        break;
    case 4:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 1 Symptoms:</span><br>Walking<br>";
        break;
    case 5:

```

```

        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 1 Symptoms:</span><br>Facial Expressions<br>";
        break;
    case 6:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 2 Symptoms:</span><br>Rigidity<br>";
        break;
    case 7:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 3 Symptoms:</span><br>Motor Symptoms<br>";
        break;
    case 8:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 4 Symptoms:</span><br>Help with Activities of Daily Living<br>";
        break;
    case 9:
        resultDiv.innerHTML = "<span style='color: green; font-weight:
bold;'>Stage 5 Symptoms:</span><br>Stiffness<br>";
        break;
    default:
        resultDiv.innerHTML = "Invalid stage";
        break;
    }
}
</script>

</body>

```

</html>

CONNECTING TO FLASK:

```
from flask import Flask, render_template, request

import tensorflow as tf

from keras.preprocessing import image as keras_image

import numpy as np

from PIL import Image

import os


app = Flask(__name__)


# Load the TensorFlow model

classifierLoad = tf.keras.models.load_model('model2.h5')

class_labels = ["Normal", "Parkinson"]


@app.route('/')

def index():

    return render_template('frontpage.html')
```



```

@app.route("/risk")

def risk():

    return render_template("risk.html")

@app.route('/upload', methods=['POST'])

def upload():

    try:

        # Check if a file was uploaded

        if 'image_file' not in request.files:

            return render_template('predict.html', error='No file part')

        uploaded_file = request.files['image_file']

        # Check if the file has a valid name and extension

        if uploaded_file.filename == "":

            return render_template('predict.html', error='No selected file')

        allowed_extensions = {'jpg', 'jpeg', 'png', 'gif'}

        if not '.' in uploaded_file.filename or uploaded_file.filename.split('.')[-1].lower() not
in allowed_extensions:

            return render_template('predict.html', error='Invalid file extension')

```

```

# Process the uploaded file

img = Image.open(uploaded_file)

img = img.resize((200, 200))

test_image2 = keras_image.img_to_array(img)

test_image2 = np.expand_dims(test_image2, axis=0)

result = classifierLoad.predict(test_image2)


# Convert the result probabilities to class labels

if result[0][0] > result[0][1]:

    prediction = "Normal"

else:

    prediction = "Parkinson"


# Save the uploaded image

upload_folder = 'static/uploads' # You can change this folder path as needed

os.makedirs(upload_folder, exist_ok=True)

image_filename = os.path.join(upload_folder, uploaded_file.filename)

img.save(image_filename)

```

```
        return render_template('predict.html', prediction=prediction,  
                                image_filename=image_filename)
```

```
    except Exception as e:
```

```
        return render_template('predict.html', error=str(e))
```

```
if __name__ == '__main__':
```

```
    app.run(debug=False, port=300)
```