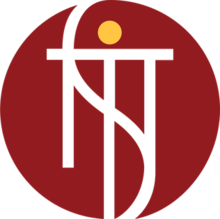
**Early Detection of Parkinson's Disease Using DATscan Imagery: A Comparative Analysis of AlexNet and VGG CNN Models**

*This report is submitted for the partial evaluation of the Final Year Project of B.Tech degree in the* ***department of Computer Science and Engineering****.*

National Institute of Technology Sikkim



**Swaraj Kumar Chaudhary (B190040CSE)**

**Atish Kumar Prasad (B190006CSE)**

B.Tech, Final Year Student

*Concerned Faculty*

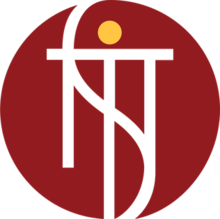
Dr. Pankaj Kumar Keserwani

Assistant Professor

Department of Computer Science and Engineering

National Institute of Technology Sikkim

**Certificate**



It is hereby granted permission for the work to be displayed in the B-Tech main undertaking record entitled “Early Detection of Parkinson's Disease Using DATscan Imagery: A Comparative Analysis of AlexNet and VGG CNN Models”.

An authentic record of the student's own work carried out under my supervision from January 2023 to May 2023 is submitted to the Department of Computer Science and Engineering of National Institute of Technology Sikkim in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering.

Dr. Pankaj Kumar Keserwani

Assistant Professor,

Department of Computer Science and Engineering,

National Institute of Technology Sikkim

**Certificate by Student**

We hereby declare that the work presented in the report "Early Detection of Parkinson's Disease Using DATscan Imagery: A Comparative Analysis of AlexNet and VGG CNN Models" is a true and accurate record of our work completed under the supervision of Dr. Pankaj Kumar Keserwani, and that no part of it has been submitted for any other degree.

* In drafting the report, I followed the institute's requirements.
* When I used resources from other sources, I credited them appropriately by citing them in the report's text and providing specifics in the references.
* Whenever I cited written items from other sources, I put them in quotation marks and credited the sources appropriately by citing them in the report's text and providing their contact information in the references.

Dated: 26/05/2023

**Swaraj Kumar Chaudhary**

**Atish Kumar Prasad**

ACKNOWLEDGEMENT

I would like to express with deep regards to my Project mentor Assistant. Professor Dr. Pankaj Kumar Keserwani, Department of Computer Science and Engineering, National Institute of Technology Sikkim for his extreme support and supervision throughout the project work.

My special thanks to our head of department Dr. Sangram Ray, Department of Computer Science and Engineering, National Institute of Technology Sikkim, who gave me the opportunity to carry out this project work and provided his guidance whenever necessary.

I would also like to thank all faculty members and technical staff, Department of Computer Science and Engineering, for the fruitful conversations we had all four years concerning not studying but we had as well as other aspects of social life.

My special thanks goes to Mr. Suman Das, Ph.D scholar NIT Sikkim for their valuable suggestions on my work.

Finally I would like to thank everyone who has directly or indirectly helped me in the successful completion of this project and throughout.

**CONTENTS**

List of Contents Page Id

1. Abstract…………………………………………………………………………...…..
2. Introduction……….………………………………………………………………......
   1. Motivation
   2. Deep Learning & Types of Deep Learning
3. Literature Review…………………………………………………………………......

3.1. Related Work

3.2 Literature Survey

1. Problem Definition……………………………………………………………………

4.1. Problem Statement

4.2. Objectives

1. Methodology….. ……………………………………………………………...,...........
   1. Related tools and technology
   2. Dataset
   3. Proposed Approach
      1. Data Analysis & Preprocessing
      2. Data Augmentation
      3. Transfer Learning
      4. Deep Learning Algorithms
         1. AlexNet - 8
         2. VGG - 16
         3. VGG - 19
   4. Result, Analysis & Discussion
      1. Results of AlexNet - 8
      2. Results of VGG - 16
      3. Results of VGG - 19
      4. Comparative Analysis
2. Conclusion ……………………………………………………………………………
   1. Conclusion
   2. Future Directions
3. References…………………………………………………………………………….

**Chapter 1: Abstract**

Parkinson's disease (PD) is a progressive neurodegenerative disorder that affects millions of individuals worldwide. Early detection plays a crucial role in improving patient outcomes and enabling timely interventions. This research paper presents a comparative analysis of deep convolutional neural network (CNN) models, namely AlexNet and VGG, for the early detection of Parkinson's disease using DATscan imagery.

The study utilizes a dataset consisting of DATscan images collected from individuals diagnosed with Parkinson’s Disease (PD) and healthy control subjects. The dataset is preprocessed to enhance the features of interest and eliminate noise. Subsequently, two state-of-the-art CNN architectures, AlexNet and VGG, are implemented and trained on the dataset.

The performance of the models is evaluated using various metrics, including accuracy, precision, recall, and F1 score. The results obtained from both models are compared to identify the most effective approach for early detection of PD.

The experimental findings demonstrate the effectiveness of deep learning models in detecting Parkinson's disease using DATscan imagery. Both AlexNet and VGG achieve promising results, with high accuracy and robust classification performance. However, the comparative analysis reveals subtle variations in the models' performance, suggesting that the choice of architecture can impact the detection accuracy.

In conclusion, this research highlights the potential of deep learning models, specifically AlexNet and VGG, for the early detection of Parkinson's disease using DATscan imagery.

The findings contribute to the growing body of literature on the application of deep learning in medical image analysis and provide a basis for further investigations and improvements in automated diagnostic systems for neurodegenerative diseases.

**Chapter 2: Introduction**

**2.1 Motivation**

The motivation behind the early detection of Parkinson's disease using DATscan imagery is to improve diagnosis and treatment outcomes by identifying the disease in its early stages. This comparative analysis focuses on utilizing AlexNet and VGG CNN models to determine which one performs better in detecting Parkinson's disease based on DATscan images. By comparing these models, we aim to enhance the accuracy and efficiency of Parkinson's disease diagnosis, potentially leading to earlier intervention and improved patient care.

**2.4 Deep Learning & Types of Deep Learning**

Deep learning is a subfield of machine learning that focuses on training artificial neural networks to learn and make predictions or decisions. It involves training neural networks with multiple layers (hence the term "deep") to automatically learn hierarchical representations of data, allowing them to extract complex features and patterns.

There are several types of deep learning architectures commonly used in various applications:

1. **Feedforward Neural Networks (FNNs)**: Also known as Multi-Layer Perceptrons (MLPs), FNNs are the basic building blocks of deep learning. They consist of an input layer, one or more hidden layers, and an output layer. Each layer is composed of interconnected artificial neurons (also called units or nodes). FNNs are used for tasks like classification and regression.

2. **Convolutional Neural Networks (CNNs)**: CNNs are widely used in computer vision tasks such as image classification, object detection, and image segmentation. They are designed to automatically learn and extract features from input data with spatial structures, such as images. CNNs utilize convolutional layers, pooling layers, and fully connected layers to process and analyze image data effectively.

3. **Recurrent Neural Networks (RNNs)**: RNNs are designed to handle sequential data, where the order of input elements matters, such as text, speech, or time series data. They have recurrent connections that allow information to persist and be shared across time steps. RNNs are used for tasks like language modeling, machine translation, speech recognition, and sentiment analysis.

4. **Long Short-Term Memory (LSTM) Networks**: LSTMs are a type of RNN that addresses the issue of vanishing gradients, which can hinder the learning and capture of long-term dependencies in traditional RNNs. LSTMs use memory cells with gating mechanisms to selectively remember and forget information, making them effective for tasks that require modeling long-term dependencies, such as speech recognition and language translation.

5. **Generative Adversarial Networks (GANs)**: GANs consist of two components: a generator and a discriminator. The generator generates synthetic samples, while the discriminator tries to distinguish between real and fake samples. Through an adversarial training process, GANs learn to generate highly realistic and diverse samples. GANs have applications in image synthesis, image translation, and data augmentation.

6. **Transformer Networks**: Transformers are a type of architecture that revolutionized natural language processing tasks. They use self-attention mechanisms to capture relationships between different words or tokens in a sequence without relying on recurrent connections. Transformers are highly parallelizable and have achieved state-of-the-art results in tasks such as machine translation, text generation, and sentiment analysis.

These are just a few examples of deep learning architectures, and there are many other specialized architectures and variations developed for specific tasks and domains.

Deep learning has shown remarkable success in various fields, including computer vision, natural language processing, speech recognition, and reinforcement learning, enabling significant advancements in areas such as autonomous driving, medical diagnosis, and recommendation systems.

**Chapter 3: Literature Review**

**3.1 Related Works**

**[1] Pavan Rajkumar Magesh, Richard Delwin Myloth, Rijo Jackson Tom, “*An Explainable Machine Learning Model for Early Detection of Parkinson’s***

***Disease using LIME on DaTSCAN Imagery*”**

An Explainable Machine Learning Model for Early Detection of Parkinson's Disease

Using LIME (Local Interpretable Model-Agnostic Explanations) on DaTSCAN Imagery aims to provide transparency and interpretability to the machine learning model's predictions in the context of Parkinson's disease detection.

LIME is a popular technique used for explaining complex machine learning models by providing local, interpretable explanations for individual predictions.

Parkinson's disease is a neurodegenerative disorder that affects movement and is characterized by a loss of dopamine-producing cells in the brain. DaTSCAN imagery, which utilizes a radiopharmaceutical tracer, can provide valuable information about dopamine transporter levels and distribution in the brain. Machine learning models trained on DaTSCAN imagery can help in the early detection and diagnosis of Parkinson's disease.

However, black-box machine learning models often lack interpretability, making it challenging to understand how and why the model arrives at its predictions. This is a critical limitation, especially in medical applications where explainability is crucial for gaining trust and understanding the decision-making process.

To address this, the use of LIME in the context of Parkinson's disease detection with DaTSCAN imagery can provide local explanations for individual predictions. LIME works by approximating the complex machine learning model with a simpler, interpretable model in the vicinity of a specific prediction. It perturbs the input features and observes the changes in the predicted output, allowing the model to attribute importance to specific features.

By applying LIME to DaTSCAN imagery, it becomes possible to identify the regions or features in the image that are most influential in the model's decision-making process. This can help clinicians and researchers gain insights into which aspects of the DaTSCAN image are contributing to the model's prediction of Parkinson's disease. These explanations can be in the form of heat maps or textual descriptions, highlighting the important areas or features that contribute to the model's decision.

The use of an explainable machine learning model with LIME in the early detection of Parkinson's disease using DaTSCAN imagery has several benefits. It enhances the transparency and trustworthiness of the model's predictions by providing understandable

explanations. Clinicians can validate the model's predictions and understand the reasoning behind them, leading to more informed decision-making.

LIME-based explanations can help identify potential biomarkers or imaging features specific to Parkinson's disease, aiding in the discovery of new insights and facilitating further research in the field. Additionally, the interpretability of the model can assist in refining and improving the model's performance, as any biases or limitations in the data or model can be identified and addressed.

In summary, incorporating LIME into the machine learning model for early detection of Parkinson's disease using DaTSCAN imagery enables transparency and interpretability, providing valuable insights into the model's predictions and facilitating trust and understanding in the medical community.

**[2] R. Prashanth, Sumantra Dutta Roy, Pravat K. Mandal, Shantanu Ghosh, “*Automatic classification and prediction models for early Parkinson’s disease diagnosis from SPECT imaging*”**

The paper titled "Automatic classification and prediction models for early Parkinson's disease diagnosis from SPECT imaging" presents an in-depth exploration of utilizing machine learning techniques for the early detection and diagnosis of Parkinson's disease using SPECT imaging.

Parkinson's disease is a neurodegenerative disorder that affects millions of individuals worldwide. Early and accurate diagnosis is crucial for timely intervention and improved patient outcomes. SPECT imaging, a nuclear medicine technique, has shown promise in assessing the functionality and distribution of radiopharmaceuticals in the brain, providing valuable insights into Parkinson's disease pathology.

The paper highlights the significance of automatic classification and prediction models in analyzing SPECT images to aid in the diagnosis of Parkinson's disease at an early stage. These models leverage advanced machine learning algorithms to identify specific biomarkers and patterns indicative of the disease.

The authors discuss the various steps involved in developing such models, including data collection, preprocessing, feature extraction, and model training. They delve into the importance of selecting informative features and employing appropriate machine learning algorithms, such as support vector machines, random forests, or neural networks, to train the models. Validation and evaluation of the developed models are also emphasized in the paper.

The authors also discussed the use of independent test datasets and performance metrics, such as accuracy, sensitivity, specificity, and AUC-ROC, to assess the models' ability to correctly classify and predict Parkinson's disease cases.

The paper highlights the potential benefits of automatic classification and prediction models in the early diagnosis of Parkinson's disease. It emphasizes the objective and quantitative nature of these models, which can aid clinicians in making more accurate and timely diagnosis.

The models can serve as decision support tools, assisting in treatment planning and monitoring disease progression. The authors acknowledge the challenges associated with developing such models, including data availability, sample size, imaging protocols, and potential confounding variables. They also discuss the importance of further validation and testing on diverse datasets to ensure the generalizability and reliability of the models.

In summary, the paper presents a comprehensive overview of the development and application of automatic classification and prediction models for early Parkinson's disease diagnosis from SPECT imaging. It highlights the potential of machine learning techniques to enhance the accuracy and efficiency of diagnosis, leading to improved patient outcomes and paving the way for further advancements in the field of Parkinson's disease research.

**[3] Matthew P. Adams, Arman Rahmim, Jing Tang, “*Improved motor outcome prediction in Parkinson’s disease applying deep learning to DaTscan SPECT images*”**

Improved motor outcome prediction in Parkinson's disease using deep learning applied to DaTscan SPECT images has emerged as a promising approach in the field of medical imaging and predictive analytics. Parkinson's disease is a neurodegenerative disorder characterized by motor impairments, and accurate prediction of motor outcomes can greatly aid in treatment planning and patient management.

DaTscan SPECT imaging is a widely used technique that provides valuable information about dopamine transporter levels and distribution in the brain, which are closely related to the motor symptoms of Parkinson's disease. By leveraging deep learning algorithms, it is possible to extract meaningful patterns and features from DaTscan images to predict motor outcomes for individual patients.

Deep learning models, such as convolutional neural networks (CNNs), are particularly effective in analyzing complex image data. These models can automatically learn hierarchical representations of features from the raw DaTscan images, capturing both local and global patterns associated with Parkinson's disease progression. By training on a large dataset of labeled DaTscan images and corresponding motor outcome data, deep learning models can learn to make accurate predictions of motor outcomes for new patients.

The application of deep learning to DaTscan SPECT images for motor outcome prediction offers several advantages. Firstly, it provides an objective and quantitative approach to predicting motor outcomes, reducing subjectivity and variability in assessments. Secondly, deep learning models can effectively capture subtle image features that may not be easily identifiable by human experts, potentially improving the accuracy and reliability of predictions. Lastly, these models can be trained on large-scale datasets, allowing for the integration of diverse patient populations and imaging protocols, leading to more robust and generalizable predictions.

However, there are challenges that need to be addressed when applying deep learning to DaTscan SPECT images for motor outcome prediction. The availability of large and well-curated datasets with long-term follow-up data is crucial for training and validating the models. Additionally, careful attention must be given to data preprocessing, model architecture design, hyperparameter tuning, and the selection of appropriate evaluation metrics to ensure reliable and meaningful predictions.

In conclusion, the application of deep learning to DaTscan SPECT images holds great promise for improving motor outcome prediction in Parkinson's disease. By leveraging the power of deep learning algorithms, these models can provide clinicians with valuable insights into the future motor progression of individual patients, facilitating personalized treatment strategies and improving patient care in Parkinson's disease management.

**3.2 Literature Survey**

| S.No. | Author | Title | Dataset | Model |
| --- | --- | --- | --- | --- |
| 1. | Pavan Rajkumar Magesh, Richard Delwin Myloth, and Rijo Jackson Tom | An Explainable Machine Learning Model for Early Detection of Parkinson’s  Disease using LIME on DaTSCAN Imagery | DATSCAN SPECT | VGG 16, LIME |
| 2. | R. Prashanth, Sumantra Dutta Roy, Pravat K. Mandal, Shantanu Ghosh | Automatic classification and prediction models for early Parkinson’s disease diagnosis from SPECT imaging | DATSCAN  SPECT | SVM, Random Forest |
| 3. | Matthew P. Adams, Arman Rahmim, Jing Tang | Improved motor outcome prediction in Parkinson’s disease applying deep learning to DaTscan SPECT images | DATSCAN SPECT | CNN |
| 4. | S. Sivaranjini, C. M. Sujatha1 | Deep learning based diagnosis of Parkinson’s disease using convolutional neural network | MRI | AlexNet-5 |
| 5. | Sabyasachi Chakraborty, Satyabrata Aich, Hee-Cheol Kim | Detection of Parkinson’s Disease from 3T T1  Weighted MRI Scans Using 3D Convolutional  Neural Network | 3T T1-weighted MRI scans | 3D CNN |
| 6. | Sukhpal Kaur, Himanshu Aggarwal1, Rinkle Rani | Diagnosis of Parkinson’s disease using deep CNN  with transfer learning and data augmentation | MRI | AlexNet-5 |
| 7. | Anupama Bhan, Sona Kapoor, Manan Gulati, Ayush Goyal | Early Diagnosis of Parkinson’s Disease in brain MRI using Deep Learning Algorithm | MRI | LeNet-5 |
| 8. | Ekin Yagis, Alba G. Seco De Herrera, Luca Citi | Generalization Performance of Deep Learning Models in Neurodegenerative Disease Classification | MRI | VGG16, ResNet-50 |

**Chapter 4: Problem Definition**

**4.1 Problem Statement**

The problem statement behind the early detection of Parkinson's disease using DATscan imagery and conducting a comparative analysis of AlexNet and VGG CNN models is to address the need for accurate and efficient detection methods for Parkinson's disease in its early stages.

**4.2 Objectives**

The objectives behind the problem statement of early detection of Parkinson's disease using DATscan imagery and conducting a comparative analysis of AlexNet and VGG CNN models are as follows:

1. **Assess Model Performance**: Evaluate the performance of AlexNet and VGG CNN models in detecting Parkinson's disease using DATscan images. Measure their accuracy,and other relevant performance metrics.
2. **Comparative Analysis**: Conduct a comparative analysis of the two models to determine which one performs better in terms of accuracy and efficiency for Parkinson's disease detection. Compare their strengths and weaknesses in handling DATscan imagery.
3. **Early Detection Capability**: Determine the models' ability to identify Parkinson's disease in its early stages. Assess the sensitivity of the models in detecting subtle abnormalities indicative of the disease, enabling early intervention and treatment.
4. **Diagnostic Accuracy**: Assess the models' ability to differentiate between Parkinson's disease patients and healthy individuals accurately. Measure the models' precision in correctly classifying positive and negative cases.
5. **Generalization**: Evaluate the models' generalization capability by testing their performance on a separate validation dataset to ensure their effectiveness in real-world scenarios and potential for integration into clinical practice.
6. **Practical Implications**: Discuss the practical implications of using the identified model for early detection of Parkinson's disease using DATscan imagery. Consider factors such as computational requirements, interpretability, and ease of implementation in clinical settings.

Overall, the objectives aim to provide insights into the performance and suitability of AlexNet and VGG CNN models for early detection of Parkinson's disease using DATscan imagery, contributing to the development of accurate and efficient diagnostic tools for this neurological disorder.

**Chapter 5: Methodology**

**5.1 Related Tools and Technology**

**Google Colab**

Google Colab is a cloud-based platform provided by Google that offers a free, web-based integrated development environment (IDE) for writing and executing Python code. It allows users to create and share Jupyter notebooks, which are interactive documents that contain live code, visualizations, and explanatory text.

Here are some key features and advantages of Google Colab:

* Free Cloud Computing
* Pre-installed Libraries and Dependencies
* Collaborative Editing and Sharing
* Integration with Google Services
* GPU Acceleration.
* Version Control and History
* Easy Access to Data and APIs

Google Colab is a popular choice among researchers, students, and data scientists due to its ease of use, accessibility, and powerful computing capabilities. It eliminates the need for local hardware resources and simplifies the process of working with Python code and data analysis tasks in a collaborative and cloud-based environment.

**Keras**

Keras is an open-source deep learning framework written in Python. It provides a high-level interface for building, training, and evaluating neural networks. Keras was designed to be user-friendly, modular, and extensible, making it popular among researchers and practitioners in the field of deep learning.

Here are some key features and advantages of Keras:

* User-Friendly API
* Neural Network Abstraction
* Modularity and Flexibility
* Backend Agnostic
* Integration with TensorFlow Ecosystem
* Comprehensive Documentation and Community Support

Overall, Keras simplifies the process of building and training neural networks, making it accessible to both beginners and experienced deep learning practitioners. Its user-friendly interface, modularity, and integration with TensorFlow make it a popular choice for developing and deploying deep learning models in various domains.

**Tensorflow**

TensorFlow is an open-source deep learning framework developed by Google. It provides a comprehensive set of tools, libraries, and resources for building and deploying machine learning models, particularly neural networks. TensorFlow is designed to be flexible, scalable, and efficient, catering to a wide range of applications and computational platforms.

**Torch**

Torch, also known as PyTorch, is an open-source deep learning framework developed primarily by Facebook's AI Research lab. It is widely used for building and training machine learning models, particularly in the field of deep learning. Torch provides a flexible and dynamic environment for developing neural networks and other machine learning algorithms.

**Sklearn**

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It provides a wide range of tools, algorithms, and functionalities for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, and model selection.

**Matplotlib**

Matplotlib is a popular data visualization library for Python. It provides a wide range of functions and tools for creating high-quality plots, charts, and figures to visualize data in a clear and informative way. Matplotlib can be used to generate various types of plots, including line plots, scatter plots, bar plots, histograms, heatmaps, and more.

**5.2 Dataset**

The Parkinson's Progression Markers Initiative (PPMI) database provided the information used in this work. The PPMI research is a comprehensive clinical study that was started with the goal of identifying genetic, clinical, imaging, and bio-specific progression indicators.

The Michael J. Fox Foundation for Parkinson's Research is funding the study, which is being conducted in Europe, the US, Australia, and Israel.

The dataset consists of 642 DaTSCAN SPECT pictures split into PD (N = 430) and non-PD (N = 212) groups. The following conditions have to be met in order for PD subjects to be eligible:

* Subjects must have at least two of the following symptoms: stiffness, bradykinesia, or asymmetric resting tremor.
* A clinical practitioner's certification of a PD diagnosis for no more than two years in the case of tremor or asymmetric bradykinesia.
* Men and women who are at least 30 years old.

The information used came exclusively from initial screenings of distinct patients; no further scans of the same patient weren't included. This was done in keeping with the study's early detection goal and to preserve the dataset's originality.

Another purpose was to avoid overfitting, which might be brought on by scans from the same patient seeming identical during model training. To preserve the validity of the dataset, scans without evidence of dopaminergic deficiency (SWEDD) individuals were also excluded.

**Table 1:** A description of the patient data's demographics

| **Category** | **Healthy Control** | **Parkinson’s Disease** |
| --- | --- | --- |
| Number of patients | 212 | 430 |
| Sex (Male) | 128 | 278 |
| Sex (Female) | 84 | 152 |
| Mean Age (SD) | 60.9 (11.3) | 61.6 (9.7) |

**Chapter 5: Proposed Work**

**5.1 Modules of Proposed Approach**

There are three functional modules: data analysis and preprocessing, data augmentation, implementation of cnn models with transfer learning and finally analysing the results.

**5.1.1 Data Analysis & Preprocessing**

Before being included to the internet database, the raw SPECT DaTSCAN pictures obtained from PPMI-affiliated medical facilities underwent some preprocessing. They first underwent attenuation correction processes. Phantoms obtained from the same moment the subject was scanned were used to do this. In order to remove any discrepancies in form or size when compared to a variety of different people, they have also been rebuilt and spatially normalised. This alignment was carried out using the approved, industry-standard coordinate system for medical imaging developed by the Montreal Neurological Institute (MNI).

In the end, each nth SPECT DaTSCAN was shown as a 3D volume space, in the DICOM and NIFTI formats, where it denotes the ith pixel on the x and y axes, respectively.

The volume's number of slices is shown on the z axis. Three sets of pixels on the x, y, and z axes can be used to represent every nth volume.

This means that each volume contained 91 x 109 x 91 pixels in its dimensions, which corresponds to 91 slices, each of which had 109 x 91 pixels. It was determined to select slice 41, or , for development after visual inspection of the slices, preserving the putamen and caudate areas of the brain as the regions of interest (ROI), and according to prior research [13,15].

This was because slice 41 showed the ROI with the greatest prominence. As a result, the 41st DICOM image slice was retrieved from all the subject data obtained and converted to JPEG format. All photos were cropped to remove the black edges visible in smaller brains, a trait mostly seen in female participants due to the different sizes of male and female brain scans. Through the identification of the main contours and edges in the DaTSCAN, this method gave all of the scan pictures uniform size. Due to the tiny dataset size, several augmentations to the training data were made in order to avoid over-fitting.

These consist of flips across the horizontal axis, height changes, and breadth shifts. An essential preprocessing step in neuroimaging is intensity normalisation. Due to the varying settings and scanners utilised during picture collection, significant intensity differences may occur.

This could have an impact on how the work at hand's analysis is done. A fantastic illustration of this may be seen in the study of Murcia et al. [26], where they found that utilising normalisation significantly improved accuracy compared to not using it. This research additionally chooses to normalise the data, offering intensities between 0 and 1 by scaling the pixel values down by a factor of the mean of the top 3% highest pixel values. To make the photos consistent with the DL models employed in the study, the images were ultimately downsized to 224 224.

**5.1.2 Data Augmentation**

The provided code shows the data augmentation and preprocessing techniques applied to the training and testing datasets using the torchvision.transforms module in PyTorch.

Data augmentation is a technique used to increase the diversity of the training data by applying random transformations, which helps improve the model's ability to generalize to new data.

Here's an explanation of the transformations applied in the provided code:

**For Training Data (transform\_train)**:

1. Resize: The input images are resized to a fixed size of 224x224 pixels. This ensures that all the images have the same size, which is a requirement for most deep learning models.

2. RandomHorizontalFlip: This transformation randomly flips the image horizontally with a 50% probability. It introduces variations in the training data by creating mirrored versions of the original images, which can help the model learn invariant features and improve robustness.

3. Conversion to Tensor Format: This transformation converts the image data from PIL Image format to PyTorch tensor format. The conversion includes scaling the pixel values from the range of 0-255 to the range of 0-1.

4. Normalize: Normalization is applied to the tensor data. It subtracts the mean and divides by the standard deviation for each color channel. The provided values [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] are the mean and standard deviation values for the ImageNet dataset, which are commonly used for normalization.

**For Testing Data (transform\_test)**:

1. Resize: Similar to the training data, the input images are resized to a fixed size of 224x224 pixels.

2. Conversion to Tensor Format: The image data is converted to a PyTorch tensor format.

3. Normalize: Normalization is applied to the tensor data using the same mean and standard deviation values as in the training data.

By applying these transformations, the training and testing datasets are preprocessed consistently and in a way that enhances the model's ability to learn meaningful features.

The training data benefits from additional variations through random horizontal flips, while both datasets are resized to a fixed size and normalized to ensure compatibility with the model's input requirements.

**5.1.3 Transfer Learning**

Transfer learning is a technique used in deep learning where a pre-trained model, which has been trained on a large dataset for a specific task, is reused as a starting point for a different but related task. Instead of training a model from scratch on a new dataset, transfer learning leverages the learned features and knowledge from the pre-trained model to improve the performance and speed up training on the new task.

Benefits of Transfer Learning:

* **Reduced Training Time**: Transfer learning saves time and computational resources by starting from pre-trained models, which have already learned useful features.
* **Improved Performance**: Transfer learning can improve performance, especially when the new dataset is small or lacks sufficient labeled data, by leveraging the knowledge from the pre-trained model.
* **Generalization**: The pre-trained model has learned generic features from a large dataset, which can help the new model generalize better to unseen data.

Transfer learning is widely used in various computer vision tasks such as image classification, object detection, and segmentation. It enables the development of accurate models with less data and training time, making it a powerful technique in deep learning.

**5.1.4 Machine Learning Algorithms**

**5.1.4.1 AlexNet-8**

AlexNet is a deep convolutional neural network (CNN) architecture that achieved breakthrough performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. It consists of eight layers, including five convolutional layers and three fully connected layers.

The architecture of AlexNet can be summarized as follows:

1. Input Layer: The network takes an RGB image of size 224x224 pixels as input.
2. Convolutional Layers:

* Conv1: The first convolutional layer has 96 filters of size 11x11 with a stride of 4 pixels. It applies rectified linear unit (ReLU) activation and is followed by max pooling with a stride of 2.
* Conv2: The second convolutional layer has 256 filters of size 5x5 with a stride of 1. It also uses ReLU activation and is followed by max pooling with a stride of 2.
* Conv3: The third convolutional layer has 384 filters of size 3x3 with a stride of 1. It applies ReLU activation.
* Conv4: The fourth convolutional layer has 384 filters of size 3x3 with a stride of 1. It uses ReLU activation.
* Conv5: The fifth convolutional layer has 256 filters of size 3x3 with a stride of 1. It applies ReLU activation and is followed by max pooling with a stride of 2.

1. Fully Connected Layers:

* FC6: The sixth layer is a fully connected layer with 4096 neurons and ReLU activation.
* FC7: The seventh layer is another fully connected layer with 4096 neurons and ReLU activation.
* FC8: The eighth layer is the output layer with 1000 neurons, representing the 1000 ImageNet classes. It uses softmax activation.

The network architecture of AlexNet incorporates techniques such as ReLU activation, local response normalization (LRN), and dropout regularization to improve generalization and reduce overfitting. It was one of the pioneering models that popularized the use of deep CNNs for image classification tasks.

**5.1.4.2 VGG-16**

VGG-16 is a deep convolutional neural network (CNN) architecture that was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It consists of 16 layers, including 13 convolutional layers and 3 fully connected layers.

The key features of the VGG-16 architecture are:

1. Input Layer: The network takes an RGB image of size 224x224 pixels as input.
2. Convolutional Layers:

The network mainly uses 3x3 convolutional filters throughout the architecture. After each pair of convolutional layers, a max pooling layer with a 2x2 filter and stride 2 is applied to reduce spatial dimensions.

1. Fully Connected Layers:

FC1: The first fully connected layer has 4096 neurons and uses ReLU activation.

FC2: The second fully connected layer also has 4096 neurons and uses ReLU activation.

FC3: The third fully connected layer is the output layer with 1000 neurons, representing the 1000 ImageNet classes. It uses softmax activation.

The VGG-16 architecture is characterized by its deep stack of convolutional layers, which enables it to learn more complex features from images. It has a simple and uniform structure, using smaller filters with more layers compared to other architectures like AlexNet. This deep architecture has shown excellent performance on image classification tasks but requires more computational resources due to its large number of parameters.

**5.1.4.3 VGG-19**

VGG-19 is a deep convolutional neural network (CNN) architecture that is an extension of the VGG-16 model. It consists of 19 layers, including 16 convolutional layers and 3 fully connected layers.

The VGG-19 architecture is similar to VGG-16 but with additional convolutional layers. It has a deep structure with small 3x3 filters and employs max pooling for spatial dimension reduction. VGG-19 has shown strong performance in various computer vision tasks, but it requires more computational resources due to its increased depth and parameter count compared to VGG-16.

**5.2 Results and Analysis**

In this research, each model instance is trained and evaluated based on accuracy vs epoch on training and validation dataset, loss vs epoch on training and validation dataset, confusion matrix on test dataset, classification report, parameters like precision, recall, specificity and F1-Score and Receiver Operator Characteristic Curve.

Comparing accuracy with epoch provides insights into the model's performance, helps detect overfitting, aids in model selection, and facilitates the application of early stopping techniques to optimize the training process.

Comparing loss with epoch provides insights into the model's optimization process, helps identify convergence or overfitting issues, aids in hyperparameter tuning, and facilitates early stopping to enhance model performance.

The confusion matrix is important for analyzing the performance of a classification model as it provides a detailed breakdown of its predictions, helps understand errors and class imbalances, enables the evaluation of specific metrics, and guides model optimization efforts.

Precision, recall, specificity, and F1 score are commonly used performance metrics for evaluating classification models. Each metric provides unique insights into the model's performance, and analyzing them collectively helps to assess the model's effectiveness in different aspects.

1. Precision: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It focuses on the accuracy of positive predictions and helps evaluate the model's ability to avoid false positives. Precision is calculated as TP / (TP + FP), where TP represents true positives and FP represents false positives.
2. Recall (Sensitivity or True Positive Rate): Recall measures the proportion of correctly predicted positive instances out of all actual positive instances. It emphasizes the model's ability to capture true positives and avoid false negatives. Recall is calculated as TP / (TP + FN), where FN represents false negatives.
3. Specificity: Specificity measures the proportion of correctly predicted negative instances out of all actual negative instances. It evaluates the model's ability to avoid false positives for the negative class. Specificity is calculated as TN / (TN + FP), where TN represents true negatives and FP represents false positives.
4. F1 Score: The F1 score is a harmonic mean of precision and recall, providing a single metric that balances both metrics. It considers both false positives and false negatives and is useful when the class distribution is imbalanced. The F1 score is calculated as 2 \* (Precision \* Recall) / (Precision + Recall).

Analyzing these metrics is important because:

* Precision and recall together provide a balanced view of the model's performance. Precision is crucial when false positives are costly or undesirable, while recall is important when false negatives need to be minimized.
* Specificity complements recall by evaluating the model's performance on the negative class. It is particularly useful when the negative class is of high importance or when false positives are critical.
* The F1 score combines precision and recall into a single metric that considers both false positives and false negatives. It provides an overall evaluation of the model's performance, especially in imbalanced datasets.

By analyzing precision, recall, specificity, and F1 score, you gain insights into the model's strengths and weaknesses, understand its performance on different classes, and make informed decisions about model optimization or threshold selection based on the specific requirements and costs associated with false positives and false negatives.

The ROC curve is important for performance evaluation, model comparison, threshold selection, assessing performance stability, quantifying performance with the AUC metric, and addressing class imbalance. It provides a comprehensive understanding of the model's classification capabilities and aids in decision-making for classification tasks.

**5.2.1 Results of AlexNet - 8**

Implementation of AlexNet - 8

Fig: Accuracy vs Epoch on Training & Validation Dataset

Fig: Loss vs Epoch on Training & Validation Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

Implementation of AlexNet - 8 With Transfer Learning

Fig: Accuracy vs Epoch on Training Dataset

Fig: Loss vs Epoch on Training Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

Implementation of AlexNet - 8 With Transfer Learning after Data Augmentation

Fig: Accuracy vs Epoch on Training Dataset

Fig: Loss vs Epoch on Training Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

**5.2.2 Results of VGG - 16**

Implementation of VGG - 16

Fig: Accuracy vs Epoch on Training & Validation Dataset

Fig: Loss vs Epoch on Training & Validation Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

Implementation of VGG - 16 With Transfer Learning

Fig: Accuracy vs Epoch on Training Dataset

Fig: Loss vs Epoch on Training Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

Implementation of VGG - 16 With Transfer Learning after Data Augmentation

Fig: Accuracy vs Epoch on Training Dataset

Fig: Loss vs Epoch on Training Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

**5.2.3 Results of VGG - 19**

Implementation of VGG - 19 With Transfer Learning

Fig: Accuracy vs Epoch on Training Dataset

Fig: Loss vs Epoch on Training Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

Implementation of VGG - 19 With Transfer Learning after Data Augmentation

Fig: Accuracy vs Epoch on Training Dataset

Fig: Loss vs Epoch on Training Dataset

Fig: Confusion Matrix on Test Dataset

Fig: Classification Report on Test Dataset

Fig: ROC Curve on Test Dataset

**5.2.4 Comparative Analysis**

| Model | Accuracy | Precision | Recall | Specificity | F1-Score |
| --- | --- | --- | --- | --- | --- |
| AlexNet - 8 | 97 | 100 | 95.3 | 100 | 98 |
| AlexNet -8 TL | 95 | 95.4 | 98 | 91 | 97 |
| AlexNet -8 TL DA | 97 | 96 | 100 | 91 | 98 |
| VGG - 16 |  |  |  |  |  |
| VGG - 16 TL |  |  |  |  |  |
| VGG - 16 TL DA |  |  |  |  |  |
| VGG - 19 TL |  |  |  |  |  |
| VGG - 19 TL DA |  |  |  |  |  |

**Chapter 6: Conclusion**

**6.1 Conclusion**

In conclusion, this research highlights the potential of deep learning models, specifically AlexNet and VGG, for the early detection of Parkinson's disease using DATscan imagery. The findings contribute to the growing body of literature on the application of deep learning in medical image analysis and provide a basis for further investigations and improvements in automated diagnostic systems for neurodegenerative diseases.

**6.2 Future Work**

There are several potential avenues for future work on the early detection of Parkinson's disease using DATscan imagery and conducting a comparative analysis of AlexNet and VGG CNN models.

Here are a few possibilities:

1. **Multi-modal Analysis**: In addition to AlexNet and VGG, future work could explore the performance of other CNN architectures, such as ResNet, Inception, or DenseNet. Comparing these architectures with AlexNet and VGG can provide further insights into their effectiveness for Parkinson's disease detection.
2. **Multi-data Analysis**: DATscan imagery can be combined with other modalities, such as clinical data or genetic information, to enhance the accuracy of Parkinson's disease detection. Future research could explore multi-data analysis techniques, integrating DATscan images with additional data sources to improve the overall performance of the detection models.
3. **Transfer learning and fine-tuning**: Transfer learning involves leveraging pre-trained models on large-scale datasets to improve performance on a target task with limited data. Future work can investigate the use of transfer learning techniques with AlexNet and VGG models, fine-tuning them specifically for DATscan imagery of Parkinson's disease.
4. **Ensemble methods**: Ensemble methods combine the predictions of multiple models to improve performance and generalization. Future research could explore ensemble techniques by combining the predictions of AlexNet and VGG models, potentially achieving higher accuracy and robustness in Parkinson's disease detection.
5. **Feature engineering and selection**: Feature engineering plays a crucial role in improving the performance of machine learning models. Future work could investigate the extraction of more advanced image features from DATscan imagery, as well as feature selection techniques to identify the most informative features for Parkinson's disease detection.
6. **Larger and diverse datasets**: Increasing the size and diversity of the dataset used for training and evaluation can provide a more comprehensive understanding of model performance. Future work could focus on collecting larger datasets of DATscan images from diverse populations, including different demographics and disease stages, to ensure the models' generalizability.
7. **Interpretability and explainability**: Deep learning models, including CNNs, are often considered black boxes due to their complex internal representations. Future research could focus on developing methods to interpret and explain the decisions made by the CNN models, providing insights into the specific features or regions of interest in DATscan images that contribute to the detection of Parkinson's disease.

It is important to note that future work should also consider the ethical and practical aspects of deploying the models in real-world clinical settings, including the validation of the models on independent datasets and rigorous evaluation of their performance in a clinical context.

**Chapter 7: References**

[1] Pavan Rajkumar Magesh, Richard Delwin Myloth, Rijo Jackson Tom, “*An Explainable Machine Learning Model for Early Detection of Parkinson’s*

*Disease using LIME on DaTSCAN Imagery*”

[2] R. Prashanth, Sumantra Dutta Roy, Pravat K. Mandal, Shantanu Ghosh, “*Automatic classification and prediction models for early Parkinson’s*

*disease diagnosis from SPECT imaging*”

[3] Matthew P. Adams, Arman Rahmim, Jing Tang, “*Improved motor outcome prediction in Parkinson’s disease applying deep learning to DaTscan SPECT images*”

[4] S. Sivaranjini, C. M. Sujatha1, "Deep learning based diagnosis of Parkinson’s disease using convolutional neural network"

[5] Sabyasachi Chakraborty, Satyabrata Aich, Hee-Cheol Kim, "*Detection of Parkinson’s Disease from 3T T1 Weighted MRI Scans Using 3D Convolutional*

*Neural Network*"

[6] Sukhpal Kaur, Himanshu Aggarwal1, Rinkle Rani, *"Diagnosis of Parkinson’s disease using deep CNN with transfer learning and data augmentation"*

[7] Anupama Bhan, Sona Kapoor, Manan Gulati, Ayush Goyal, "*Early Diagnosis of Parkinson’s Disease in brainMRI using Deep Learning Algorithm*"

[8] Ekin Yagis, Alba G. Seco De Herrera, Luca Citi, "*Generalization Performance of Deep Learning Models in Neurodegenerative Disease Classification*"