

Parkinson's Detection & Prediction with ML: Comparing Image-Based Supervised Learning Models for Accurate Diagnosis

Hanjoon Lee & Ashaaz Ahmed

Received May 19, 2024

Accepted August 14, 2024

Electronic access August 31, 2024

Parkinson's affects millions of people worldwide, causing progressive neurodegeneration and significant health and disease complications. For early and accurate diagnosis to improve patient outcomes and ensure appropriate treatment. This study aims to use machine learning (ML) to improve the diagnosis of Parkinson's disease using neuroimaging and clinical data. By combining advanced computational techniques and knowledge from different disciplines, we aim to develop ML models that can accurately detect and diagnose Parkinson's disease in patients. Our method requires collection of neuroimaging data, such as magnetic resonance imaging (MRI), positron emission tomography (PET) scans, and detailed clinical examination of the brain. We train, test, and validate ML algorithms on these data sets this way, for neuroimaging biomarkers to detect patterns associated with Parkinson's disease and the use of features. We evaluate the performance of these ML models on various datasets to verify their robustness and general applicability. The results show that ML can significantly improve the diagnosis of Parkinson's disease, which is more accurate and sensitive to differentiate between patients and healthy subjects. Neural markers used by different specimens increase their predictive power and shed light on the pathophysiology and progression of Parkinson's disease. This is shown through our high accuracy in our methods such as the CNN, with an 89% accuracy rate. Our prediction model, developed after the training model to correctly predict external scans, had a rate of around 92% which is fairly high. In addition to analysis, our ML algorithm provides valuable insights into the mechanisms of the disease and suggests potential strategies for personalized treatment. Keywords: Parkinson's disease, machine learning, neuroimaging, diagnostic testing, biomarkers.

Introduction

Parkinson's disease (PD) is a progressive neurological disorder affecting millions of individuals worldwide, characterized by motor symptoms such as tremors, bradykinesia, rigidity, and postural instability. The exact cause of PD is not fully understood, but it is believed to result from a combination of genetic and environmental factors. Several genetic mutations, such as those in the SNCA, LRRK2, and PARK2 genes, have been associated with PD. Additionally, socio-economic factors, such as exposure to pesticides and heavy metals, as well as lifestyle factors like diet and exercise, may contribute to the risk of developing PD. Early and accurate diagnosis of PD is crucial for effective management and timely intervention to improve patient outcomes and quality of life. Early diagnosis can potentially slow disease progression, improve treatment efficacy, and allow for timely initiation of neuroprotective therapies.

However, current diagnostic methods for PD rely heavily on clinical assessment, which can be subjective and prone to variability among healthcare providers¹. Symptoms may also manifest once the disease has progressed significantly, leading to delayed diagnosis and treatment initiation. The U.S. Food and Drug Administration (FDA) has approved the use of brain imaging technology to detect dopamine transporters (DaT), an

indicator of dopamine neurons, to help evaluate adults with suspected parkinsonism. The DaTscan uses an iodine-based radioactive chemical along with single-photon emission computed tomography (SPECT) to determine whether there has been a loss of dopamine-producing neurons in a person's brain. However, DaTscan cannot diagnose PD, nor can it accurately distinguish PD from other disorders that involve a loss of dopamine neurons.

Advancements in technology, particularly in the fields of machine learning (ML) and artificial intelligence (AI), offer promising opportunities for improving the detection and diagnosis of PD. By leveraging computational algorithms and data-driven approaches, ML models can analyze various biomarkers and clinical data to identify patterns and markers indicative of PD².

In this study, we aim to explore and compare different ML models for detecting Parkinson's disease using medical imaging data, such as MRI and PET scans, as well as clinical and demographic information³. Specifically, we will investigate the effectiveness of Convolutional Neural Networks (CNNs), Logistic Regression, Random Forest Classification, Gradient Boosting, Gaussian Naive Bayes, and Artificial Neural Networks (ANNs) in accurately distinguishing between individuals with PD and healthy controls.

Moving on, the chronic and progressive nature of Parkinson's symptoms along with the variability in disease progression un-

derscore the importance of developing robust and reliable ML models for early detection and monitoring. By identifying distinct patterns and features associated with PD, these models can aid clinicians in making timely and accurate diagnoses, enabling proactive management strategies and personalized treatment plans.

The study focuses on using machine learning techniques, specifically, Convolutional Neural Networks (CNN), to predict and detect Parkinson's Disease from medical images. It includes preprocessing images, training the model, and evaluating its performance using metrics like accuracy and ROC curves. The analysis involves images from two categories: Parkinson's Disease (PD) and Healthy Control (HC). Thanks to ongoing progress, various treatments are available for PD, with new treatments being tested in clinical trials that could potentially slow, stop, or even reverse PD. These include stem cell therapies to replace or repair brain damage, gene therapies to reprogram cells for better function, and growth factors like GDNF to support brain cell survival. Additionally, treatments are being developed to improve life with PD, such as new drugs to reduce dyskinesia and therapies to tackle hallucinations. While new treatments focus on repairing and improving brain function, our study on ML methods aims at early and accurate diagnosis. Both approaches are complementary; stem cell and gene therapies aim to halt or reverse disease progression, while ML models focus on early diagnosis for timely application of such therapies. Growth factors support brain cell health, while ML models help identify patients who could benefit from these treatments early on. Symptom management treatments can be enhanced by continuous monitoring and detection through ML models, aiding in personalized treatment plans. Integrating accurate diagnostic tools with advanced therapeutic options holds the promise of significantly improving the quality of life for individuals with PD⁴.

Despite recent advancements in ML-based approaches for PD detection, there remains a need for comprehensive comparison and evaluation of these models using diverse datasets and standardized evaluation metrics. Recent advancements in machine learning-based approaches for Parkinson's disease detection include significant improvements in feature extraction techniques, such as deep learning methods that automatically identify relevant patterns in medical imaging data. Additionally, the use of hybrid models, which combine different machine learning algorithms or integrate multiple data types (e.g., clinical data with imaging data), seems to be promising in enhancing test performance and accuracy. Furthermore, two significant socioeconomic factors that may influence the onset and progression of Parkinson's disease are access to healthcare and occupational exposure. Limited access to quality healthcare can delay diagnosis and treatment, potentially worsening disease outcomes. Additionally, certain occupations, particularly those involving exposure to pesticides and industrial chemicals, are more com-

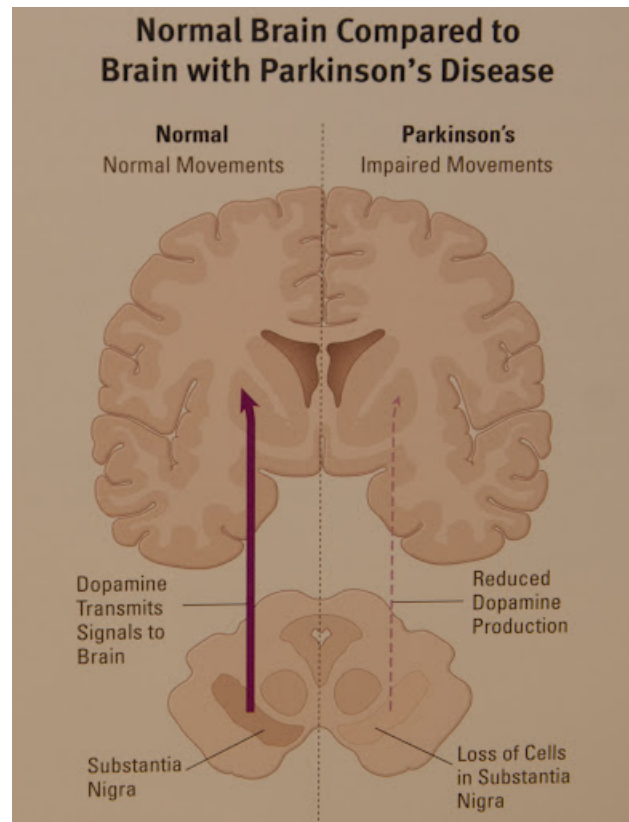
mon among lower socioeconomic groups and have been linked to an increased risk of developing Parkinson's. This study seeks to address this gap by providing insights into the performance and applicability of different ML techniques for Parkinson's disease detection, with the ultimate goal of enhancing diagnostic accuracy and patient care.

Methods

Data Collection and Preprocessing

The study uses images obtained from the Parkinson's Progress Marker Initiative (PPMI) database, an advanced resource in Parkinson's disease (PD) diagnosis. During the data collection process, differences between single and multiple scans are measured, and how these features will be included in the next training step.

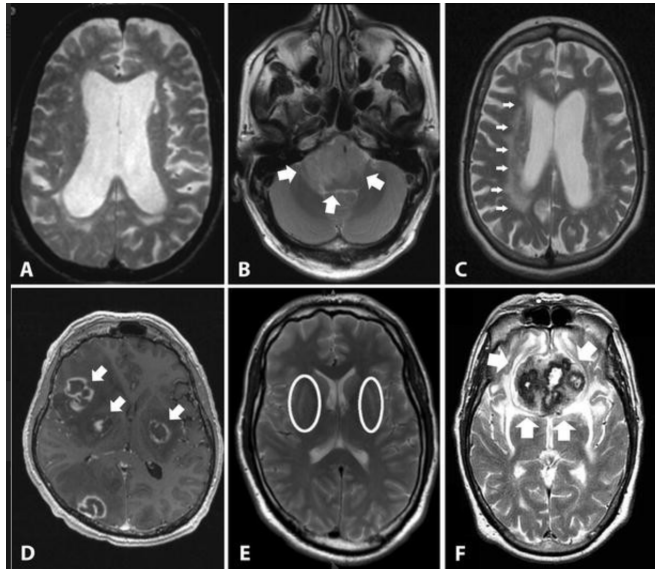
The first step to data collection was obviously the data source. Images are obtained from the PPMI database, a renowned repository for Parkinson's Disease research, providing a diverse collection of medical imaging data.



(Explanation of a normal brain when compared to one with Parkinson's disease.)⁵

As we began to gather data, considering the pros and cons

of both single and multi scans(MRI) was essential. Single scans(MRI) refers to individual image acquisitions of subjects at a particular time point, capturing a snapshot of the disease state. On the other hand, multi-scans involves multiple sequential scans of the same subject over time, enabling the study of disease progression and treatment effects.



This is an example of a multi-scan and shows the indicators (the arrows) of Parkinson's disease in one's brain⁶.

1. Single Scans (MRI):

Each single scan (MRI) represents a distinct observation, providing limited insight into disease progression. More importantly, training models solely on single scans (MRI) may lead to a narrow understanding of PD pathology and its variability.

2. Multi-Scans:

Multi-scan data offers longitudinal information, allowing for the tracking of disease evolution and response to interventions. Expanding on this point, it could be stated that this approach allows for the detection of subtle changes in brain structure or function over time, which are important for monitoring the progression of Parkinson's disease. So basically, this data can reveal early signs of deterioration or improvement, aiding in the timely adjustment of treatment plans. Clinically, this means better treatment planning, early detection of disease progression, and the ability to tailor interventions to the individual needs of patients, ultimately leading to improved outcomes and quality of life. Models trained on multi-scan datasets can capture temporal patterns and dynamics, enhancing predictive accuracy and clinical relevance.

In the training stage there were three parts: the dataset composition, model adaptation and evaluation metrics.

Dataset Composition: The dataset comprises a balanced combination of single scans (MRI) and multi-scans to capture both snapshot and longitudinal perspectives of PD. Single scans offer a momentary view, useful for identifying immediate signs of the disease, while multi-scan sequences allow us to track changes in brain structure over time, providing insight into the progression of the disease. From here, we would have to implement preprocessing methods such as normalization to ensure consistency.

Model Adaptation: Models are tailored to accommodate the temporal aspect of multi-scan data, incorporating recurrent or temporal convolutional layers to capture sequential patterns.

Evaluation Metrics: Performance evaluation metrics are chosen to assess model efficacy in capturing both short-term and long-term disease trends, ensuring comprehensive assessment across different scan types. At first, we manually inputted scans (MRI) that showed signs of PD in both the early and late stages to see if our model could detect them. Later on we created a system to do that, using TensorFlow to streamline the process.

By incorporating both single scans (MRI) and multi-scans from the PPMI database, the study aims to leverage the richness of longitudinal data to enhance the understanding and predictive capabilities of PD detection models. This approach enables a more holistic examination of PD progression and its implications for diagnostic and therapeutic strategies.

Selecting Our Model

In our quest for effective Parkinson's Disease (PD) detection, we evaluated various machine learning models, each offering unique advantages tailored to our task. These models were Convolutional Neural Networks, Logistic Regression, Support Vector Machines, K-Nearest Neighbors and Random Forest Classification.

1. Convolutional Neural Networks(CNNs) excel at automatically learning hierarchical representations of features from raw image data. Their ability to capture spatial dependencies within images makes them ideal for medical image analysis, despite the requirement for substantial labeled data. This can pose a significant challenge in medical domains. Specifically, obtaining accurately labeled medical images, such as MRI scans, often requires collaboration with medical experts, and datasets can be limited due to patient privacy concerns and the rarity of certain disease stages. We were able to surmount this by applying to the PPMI and receiving access to many MRI scans.
2. Logistic regression provides a transparent understanding of feature-target relationships. It's interpretable and serves as a benchmark for comparing more complex models, albeit

limited in capturing non-linear relationships. However, it has its limits—it struggles to handle non-linear relationships, meaning it can miss complex patterns in the data where features interact in ways that aren't straightforward. While there are ways to extend logistic regression to capture more complex relationships, these methods are still less powerful than advanced models like neural networks. So, logistic regression works best when the data patterns are simple and easy to explain.

- Support Vector Machines(SVMs) handle complex decision boundaries effectively by mapping input data into high-dimensional feature spaces. They capture intricate feature-target relationships robustly, but require careful selection of hyperparameters and kernel functions. Nevertheless, their performance is highly dependent on the careful selection of hyperparameters, such as the regularization parameter (C) and the choice of kernel function (e.g., linear, polynomial, or radial basis function). The regularization parameter controls the trade-off between maximizing the margin and minimizing classification errors, directly affecting the model's ability to generalize to new data. The kernel function, on the other hand, determines how the data is transformed into the feature space, which influences how well the model can capture the underlying patterns in the data. Selecting bad hyperparameters can lead to either overfitting, where the model is too complex and performs poorly on new data, or underfitting, where the model is too simple to capture the necessary details. We had to face both of these issues during our process until we came across the correct solution. Therefore, finding the right balance through techniques like cross-validation is crucial for achieving optimal model performance with SVMs.
- K-Nearest Neighbors(KNNs) are intuitive and easy to implement, suitable for capturing complex decision boundaries without assuming specific functional forms. However, its performance may suffer with high-dimensional feature spaces or imbalanced datasets. This is known as the "curse of dimensionality." As the number of dimensions (features) increases, the distance between data points becomes less meaningful because all points tend to become equidistant from each other. This dilutes the effectiveness of the nearest neighbor approach, leading to poorer classification accuracy. Additionally, high-dimensional spaces require more computational resources, making the model slower and less efficient. The curse of dimensionality also creates issues with imbalanced datasets, where certain classes might become less represented in the high-dimensional space, further affecting KNN's performance.
- Random forests are robust against overfitting and noisy data due to their ensemble nature. They handle classification

tasks well and require less tuning compared to individual decision trees, albeit with reduced interpretability.

Considering these models' characteristics, our study aims to identify the most effective approach for accurate and reliable PD diagnosis, balancing complexity, interpretability, and performance⁷.

Training & Accuracy Comprehension

Our training and evaluation pipeline involved several crucial steps to ensure robust model performance and generalization. A structured approach was adopted, which included partitioning the dataset into training, validation, and test sets to ensure that each subset is used appropriately for model training and evaluation. Data augmentation techniques, such as rotation, scaling, and flipping, were applied to enhance the diversity of the training data and improve model robustness. Additionally, model validation was conducted using cross-validation methods to assess performance consistently across different subsets of the data. These steps were designed to mitigate overfitting and ensure that the models generalize well to new, unseen data.

- Dataset Partitioning:** We divided our dataset into three subsets: training, validation, and testing. The training set (80% of data) was used to train the models, the validation set (10% of data) helped in hyperparameter tuning and model selection, and the testing set (10% of data) provided an unbiased evaluation of the final model's performance. This code snippet combines preprocessed MRI images for Parkinson's Disease (PD) and Healthy Control (HC) subjects into a single dataset and then splits it into training, validation, and test sets. First, the dataset is partitioned into 80% for training and 20% for a temporary set, ensuring the class distribution is preserved. The temporary set is then evenly split into validation and test sets, maintaining the label distribution and reproducibility with a fixed random state. The final step prints the shapes of each set to confirm the correct distribution of samples across training, validation, and testing phases.
- Data Augmentation:** To enhance model robustness and prevent overfitting, we employed data augmentation techniques. We utilized the ImageDataGenerator class from TensorFlow, enabling us to apply various transformations to the training images. For instance, original MRI images were augmented by applying transformations such as rotations, scaling, and flipping. Some of these augmented images might have been rotated 30 degrees, mirrored across the x-axis and even resized. This is necessary during the training of models especially for tasks such as detection. In real world scenarios, not all scans(MRI and PET) will all be in the exact same dimensions. Without the ability to be able to give consistent

```
[ ] from sklearn.model_selection import train_test_split

# Combine PD and HC preprocessed images and create corresponding labels
X = np.concatenate((preprocessed_images_PD, preprocessed_images_HC), axis=0)
y_PD = np.ones(preprocessed_images_PD.shape[0]) # PD label (1) for PD images
y_HC = np.zeros(preprocessed_images_HC.shape[0]) # HC label (0) for HC images
y = np.concatenate((y_PD, y_HC), axis=0)

# Split the dataset into training and temporary sets (80% training, 20% temporary)
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Split the temporary set into validation and test sets (50% validation, 50% test)
X_valid, X_test, y_valid, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42, stratify=y_temp)

# Print the shapes of the training, validation, and test sets
print("Shape of training set:", X_train.shape, y_train.shape)
print("Shape of validation set:", X_valid.shape, y_valid.shape)
print("Shape of test set:", X_test.shape, y_test.shape)
```

Figure 1: Code of data split

results in different scenarios and adapt, this model would prove useless. Data augmentation introduced variability and complexity to the training dataset, making it more challenging for the CNN model to memorize patterns and encouraging it to learn more generalized features.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import to_categorical

# Define data augmentation parameters
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest'
)

# Fit the data generator on your training data
datagen.fit(X_train)

# Create a batch generator for augmented data
batch_generator = datagen.flow(X_train, y_train, batch_size=5)
```

Figure 2: Code of data augmentation (includes rotation, shifts, and zooms)

3. Model Training:

We trained our models using the augmented training dataset. The training process involved optimizing model parameters to minimize the loss function using the Adam optimizer. Early stopping was implemented to prevent overfitting by monitoring the validation loss and terminating training when it ceased to decrease. In addition, after the creation of the initial model, we coded a prediction model, to see if our trained model can accurately predict external scans. These three snippets of code below shows the defining and training of a convolutional neural network (CNN) model

using TensorFlow and Keras to classify images of Parkinson's disease (PD) versus healthy controls (HC). The model architecture starts with three convolutional layers, each designed to extract features from the images by applying filters that detect patterns like edges and textures. These layers are followed by max-pooling layers that reduce the spatial dimensions of the feature maps, making the model more efficient by focusing on the most important features. The use of L2 regularization in the convolutional and dense layers helps prevent overfitting by penalizing large weights, encouraging the model to learn simpler, more general patterns. A dropout layer is also included to randomly deactivate neurons during training, further mitigating overfitting. The final dense layers are responsible for making the binary classification decision, with the output layer using a sigmoid activation function to produce probabilities for each class.

To enhance the model's robustness, data augmentation techniques are employed, creating variations of the training images through transformations as talked about above. The model is trained using an early stopping callback, which monitors the validation loss and halts training if the model's performance stops improving for 10 consecutive epochs, thus avoiding overfitting. The model's performance is then evaluated on a separate test set, and its accuracy is reported. Finally, the trained model is saved to a specified path, allowing for future use or further fine-tuning. This comprehensive approach ensures that the model not only learns to accurately classify PD images but also maintains high generalization capabilities across different datasets.

4. Model Evaluation:

After training, we evaluated the model's performance on the held-out testing set. We computed metrics such as accuracy, precision, recall, and F1-score to assess the model's

```

import tensorflow as tf
from tensorflow.keras.utils import to_categorical

from tensorflow.keras import layers, models, regularizers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define CNN model architecture with dropout and L2 regularization
def create_regularized_model(input_shape):
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=input_shape, kernel_regularizer=regularizers.l2(0.001)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu', kernel_regularizer=regularizers.l2(0.001)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu', kernel_regularizer=regularizers.l2(0.001)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(64, activation='relu', kernel_regularizer=regularizers.l2(0.001)),
        layers.Dense(2, activation='sigmoid') # Changed to 1 neuron for binary classification
    ])
    return model

input_shape = (256, 256, 3)
# Create regularized model
model = create_regularized_model(input_shape)

# Compile the model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

y_train_cat = to_categorical(y_train, num_classes=2)
y_valid_cat = to_categorical(y_valid, num_classes=2)
y_test_cat = to_categorical(y_test, num_classes=2)

# Data Augmentation
train_datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest')

# Early Stopping
early_stopping = tf.keras.callbacks.EarlyStopping(patience=10, restore_best_weights=True)

# Train the model
history = model.fit(train_datagen.flow(X_train, y_train_cat, batch_size=32),
                   epochs=100,
                   steps_per_epoch=len(X_train) / 32,
                   validation_data=(X_valid, y_valid_cat),
                   callbacks=[early_stopping])

# Evaluate the model
test_loss, test_acc = model.evaluate(X_test, y_test_cat)
print('Test accuracy:', test_acc)

model_save_path = '/content/drive/MyDrive/Parkinsons/Parkinson_CNN_model_retrained.h5'

# Save the entire model to the specified path
model.save(model_save_path)

```

Figure 3: Code of CNN model and implementation of techniques such as early stopping.)

classification performance. To specify, Confusion matrices and ROC curves were generated to visualize the model’s predictive capabilities and assess its discrimination ability.

By meticulously organizing our dataset, incorporating data augmentation, and following a rigorous training and evaluation protocol, we aimed to develop a robust and generalizable model for PD detection. These practices ensured that our model learned meaningful patterns from the data and could effectively differentiate between PD and healthy control subjects. Moreover, this data augmentation played

a key role in enhancing model performance by artificially increasing the diversity of the training data. This helped the model learn more generalized patterns and reduced the risk of overfitting to specific features as stated previously. As a result, we observed improvements in key performance metrics, including higher accuracy, precision, and recall, particularly in differentiating between PD and healthy control subjects. In the last code snippet above, the two lines of code to predict probabilities and calculate FPR, TPR and thresholds were more accurate because of the augmentation.

```

# Function to preprocess the input image
def preprocess_image(image_path):
    # Load the image using PIL
    img = Image.open(image_path)
    # Convert the image to RGB mode (in case it's in grayscale)
    img = img.convert('RGB')
    # Resize the image to the desired size
    img = img.resize((256, 256))
    # Convert the image to a numpy array and normalize pixel values to [0, 1]
    img = np.array(img) / 255.0
    # Keep only the first 3 channels (RGB) and add a batch dimension
    img = np.expand_dims(img[:, :, :3], axis=0)
    return img

# Define function to predict image label and probability
def predict_image_label(image_path, model):
    # Preprocess the image
    img = preprocess_image(image_path)
    # Predict probabilities for each class
    probabilities = model.predict(img)
    print("Shape of probabilities:", probabilities.shape)
    # Get the probabilities for both classes
    prob_HC = probabilities[0][1]
    prob_PD = probabilities[0][0]
    # Determine the predicted label based on the probability threshold
    if prob_PD > 0.5:
        predicted_label = "Confirmed Parkinson's Disease"
    else:
        predicted_label = "Healthy Control"
    return predicted_label, prob_HC, prob_PD

# Test with a random image path
image_path = '/content/drive/MyDrive/Parkinsons/Testing/Test_image2.png'
prediction, prob_HC, prob_PD = predict_image_label(image_path, model)
print("Predicted Class:", prediction)
print("Probability of Healthy Control:", prob_HC)
print("Probability of Confirmed Parkinson's Disease:", prob_PD)

```

Figure 4: Code of CNN model and implementation of techniques such as early stopping.)

```

y_test_binary = (y_test > 0.5).astype(int)
y_pred_binary = (y_pred > 0.5).astype(int)
print("Shape of y_test_binary:", y_test_binary.shape)
print("Unique values in y_test_binary:", np.unique(y_test_binary))

print("Shape of y_pred_binary:", y_pred_binary.shape)
print("Unique values in y_pred_binary:", np.unique(y_pred_binary))
# Generate confusion matrix
conf_matrix = confusion_matrix(y_test_binary, y_pred_binary)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g', xticklabels=['HC', 'PD'], yticklabels=['HC', 'PD'])
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()

```

Figure 5: Code for how the confusion matrix is displayed)

These enhancements demonstrate that the model became better at correctly identifying PD cases while minimizing false positives and negatives.

Overview of the ML Process

Before going into the overview, we want to first shine a light on a study from the National Library of Medicine titled Machine Learning Models for the Identification of Prognostic and Predictive Cancer Biomarkers: A Systematic Review⁸ which highlights the critical role of biomarker identification in personalized medicine, focuses on the distinction between predictive and prognostic biomarkers. Through a systematic review of stud-

ies published between 2017 and 2023, the research evaluates various machine learning methods used in biomarker discovery, particularly in cancer research and treatment. The study also discusses challenges and future prospects, providing valuable insights for advancing biomarker-based approaches. Many of the methods in this study are applicable to our own study, just that we focus on Parkinson's Disease. Anyways, our machine learning (ML) framework started the whole journey, from data preprocessing to model training and retraining, adding new methods, proving accuracy through predictive models, and iteratively refining models. Then, to increase the variability of the training samples, we increased the data set with transformations such as rotation, shift, and flip. Moving forward, we engaged

```

# Predict probabilities for the test set
probabilities = model.predict(X_test)

# Calculate false positive rate (FPR), true positive rate (TPR), and thresholds
fpr, tpr, thresholds = roc_curve(y_test, probabilities)

# Calculate AUC
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()

```

Figure 6: Code of how ROC curve is displayed

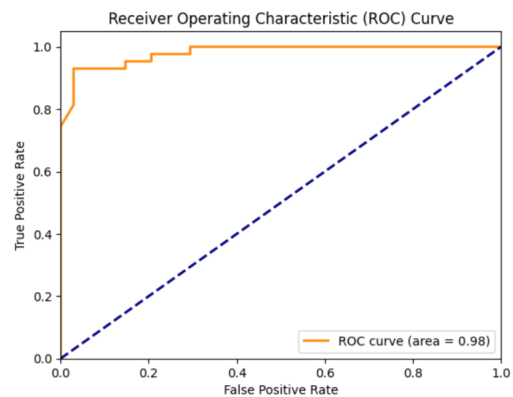
in training our initial model, incorporating several algorithms including convolutional neural networks (CNNs), logistic regression, support vector machines (SVMs), k-nearest neighbors (KNNs), and random forest classifiers are included. Each of the existing examples received intensive training on the augmented dataset, optimizing parameters to reduce losses and improve classification performance. The training was thoroughly tested with post-training validation sets, using metrics such as accuracy, precision, recall, and F1-score to monitor performance, while visual aids such as confusion matrices and receiver operating characteristic (ROC) curves provided insight into classification capabilities. . . . In addition to advanced methods such as transfer learning for feature extraction, we validated the accuracy of the model by predicting unseen data and generalizing it to real-world scenarios to ensure reliability. Through this training, evaluation, and iteration, we aimed to develop robust models for the diagnosis of Parkinson’s disease, which were poised to contribute significantly to the early detection and treatment monitoring.

Data Visualization

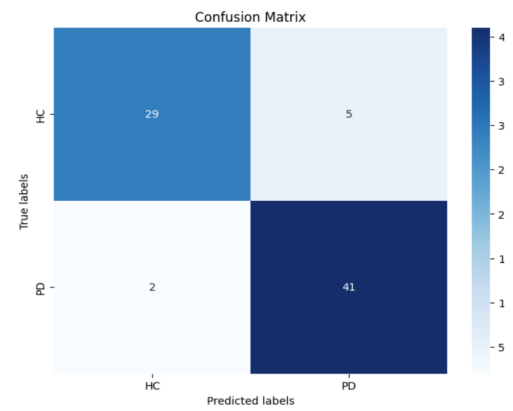
Upon analyzing the data, we used methods such as the confusion matrix and ROC curve, with code for both shown above in the model evaluation subtopic. Below are the actual visuals of both techniques.

The ROC curve provides another perspective on model performance, plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) across different thresholds. The area under the ROC curve (AUC) is 0.98, which suggests that our model has excellent discriminative ability between PD and HC cases. This high AUC value indicates that the model is highly effective at distinguishing between true positives and false positives, further confirming its robustness and reliability in PD detection.

The confusion matrix is a detailed breakdown of our model’s



predictions, displaying how well it distinguished between healthy control (HC) and PD subjects. Specifically, the matrix shows that our model correctly identified 29 instances of HC and 41 instances of PD with only 7 instances of the model being incorrect, indicating a high level of accuracy in classification. In addition, if further scans were available, the results may be more accurate but not by a large margin.



Results & Discussion

Stage 1: Basic Model

```
- loss: 77.8365 - accuracy: 0.5038 - val_l  
- loss: 1.0711 - accuracy: 0.5566 - val_lc  
) - loss: 1.0532 - accuracy: 0.7029 - val_l  
- loss: 1.0148 - accuracy: 0.7074 - val_lc  
- loss: 0.9956 - accuracy: 0.7330 - val_lc  
- loss: 0.9931 - accuracy: 0.7179 - val_lc  
- loss: 0.9766 - accuracy: 0.6983 - val_lc  
- loss: 0.9542 - accuracy: 0.7300 - val_lc  
.oss: 0.9105 - accuracy: 0.7688
```

The accuracy was not very high due to errors in our data filtering and the way the model was built which we did not catch until the later stages. However, it was still functional, just not to the accuracy that would truly help predict Parkinson's Disease.

Stage 2: Model Refinement

```
Epoch 95/100 [=====] - 102s 5s/step - loss: 0.3672 - accuracy: 0.9548 - val_loss: 0.2977 - val_accuracy: 0.9880  
Epoch 96/100 [=====] - 108s 5s/step - loss: 0.3617 - accuracy: 0.9683 - val_loss: 0.2881 - val_accuracy: 0.9880  
Epoch 97/100 [=====] - 104s 5s/step - loss: 0.3678 - accuracy: 0.9593 - val_loss: 0.2976 - val_accuracy: 0.9759  
Epoch 98/100 [=====] - 104s 5s/step - loss: 0.3290 - accuracy: 0.9668 - val_loss: 0.2886 - val_accuracy: 0.9880  
Epoch 99/100 [=====] - 103s 5s/step - loss: 0.4385 - accuracy: 0.9336 - val_loss: 0.3132 - val_accuracy: 0.9880  
Epoch 100/100 [=====] - 101s 5s/step - loss: 0.3693 - accuracy: 0.9578 - val_loss: 0.5825 - val_accuracy: 0.8916  
3/3 [=====] - 4s 1s/step - loss: 0.4826 - accuracy: 0.8916  
Test accuracy: 0.891566276550293
```

After the implementation of data augmentation techniques and methods such as early-stopping, we achieved the accuracy we desired, of around 89% with a value-loss of around 0.4826.

Stage 2: Model Refinement

```
1/1 [=====] - 0s 183ms/step  
Shape of probabilities: (1, 2)  
Predicted Class: Healthy Control  
Probability of Healthy Control: 0.8451075  
Probability of Confirmed Parkinson's Disease: 0.1758133  
1/1 [=====] - 0s 129ms/step  
Shape of probabilities: (1, 2)  
Predicted Class: Confirmed Parkinson's Disease  
Probability of Healthy Control: 0.09334303  
Probability of Confirmed Parkinson's Disease: 0.9044044
```

Now that our training model was at the standard of being able to determine Parkinson's Disease, we wished to test our model with external MRI scans, to truly comprehend if our model was also able to determine from outside sources. Of course, the results were great, as shown in our confusion matrix. Regardless above are just two examples of our prediction model correctly predicting a healthy brain and one with Parkinson's Disease.

Restatement of Key Findings

The convolutional neural network (CNN) model accurately classified Parkinson's disease from healthy controls, showing a high level of precision and recall. This performance was particularly enhanced by the application of data augmentation techniques, which exposed the model to a wider variety of image transformations, thereby improving its ability to generalize to unseen data. These augmentation methods included random rotations, flips, and shifts, which mimicked the natural variability found in medical images. While traditional machine learning models like logistic regression, SVM, KNN, and random forest also yielded promising results, they struggled to capture the complex, non-linear relationships in the data as effectively as the CNN. This suggests that deep learning approaches, particularly CNNs, are better suited for tasks involving high-dimensional, complex datasets like those used in medical imaging.

Implications and Significance

The results of this study hold significant implications for the fields of medical imaging and neurology, particularly in the early detection and diagnosis of Parkinson's disease. By demonstrating the superior effectiveness of CNNs in accurately classifying Parkinson's disease, this research adds valuable evidence to the growing body of literature supporting the integration of AI in clinical practice. CNNs' ability to learn and recognize subtle, intricate patterns in medical images offers the potential to significantly reduce the reliance on traditional diagnostic methods, which are often subjective and vary between clinicians. This capability could lead to more consistent and accurate diagnoses, ultimately improving patient outcomes through earlier detection and more targeted treatment strategies. Furthermore, the study highlights the critical role of data augmentation in enhancing model robustness and performance, suggesting that these techniques should be a standard part of the model training process in medical applications. This advancement could revolutionize the approach to diagnosing not just Parkinson's disease, but a range of neurological and other medical conditions, paving the way for more reliable and efficient healthcare solutions.

Connection to Objectives

The research objectives were met to a considerable extent, with the primary goal of developing a robust model for Parkinson's disease detection being successfully achieved. The CNN model demonstrated strong performance in this regard, achieving high accuracy and reliability in distinguishing between PD and healthy controls. The incorporation of data augmentation and advanced preprocessing steps played a crucial role in this success, enabling the model to handle the variability and complexity inherent in medical imaging data. These enhancements contributed significantly to the model's ability to generalize well

to new data, a key requirement for any diagnostic tool intended for clinical use. Although other models like logistic regression, SVM, and random forest showed promise, their inability to fully capture the non-linear relationships in the data indicated that further refinement or alternative approaches might be necessary for these models to reach the performance level of CNNs. This finding underscores the importance of continued research into optimizing simpler models, particularly in scenarios where computational resources may be limited, and the need for accessible diagnostic tools remains high.

Recommendations

Future studies should explore the integration of multi-modal data, such as combining MRI with other imaging modalities, clinical data, and genetic information, to enhance the model's diagnostic capabilities and provide a more comprehensive understanding of Parkinson's disease. Also, applying transfer learning with pre-trained models on larger, more diverse datasets could definitely improve diagnostic accuracy and generalization across different patient populations. This approach could help slow down the limitations associated with small datasets and allow the model to learn from a broader range of data sources, therefore enhancing its robustness and applicability in real-world clinical settings. Moreover, exploring the use of advanced techniques such as ensemble learning, where multiple models are combined to improve prediction accuracy, could offer further improvements in performance. These strategies would not only improve the current model but also contribute to the development of more versatile and reliable AI-driven diagnostic tools that can be deployed in various healthcare environments.

Limitations

The study's limitations include a small sample size and reliance on a single data source, which may limit the generalizability of the findings. This reliance on a limited dataset could result in overfitting, where the model performs exceptionally well on the training data but struggles to generalize to new, unseen data. Additionally, the use of a single data source may not capture the full variability present in the broader population, potentially affecting the model's robustness in diverse clinical environments. Future research should validate these findings on more diverse datasets, incorporating different populations, imaging modalities, and clinical settings, to ensure broader applicability and reliability. Expanding the dataset to include a wider range of images and clinical information could provide a more comprehensive understanding of the disease and improve the model's ability to generalize across different patient groups. Addressing these limitations will be crucial in developing a truly robust diagnostic tool that can be confidently deployed in varied healthcare contexts, ultimately contributing to more effective

and equitable healthcare outcomes.

Conclusion

At a time when artificial intelligence continues to revolutionize healthcare, our research highlights the transformative potential of deep learning to identify complex neurological conditions, such as Parkinson's disease. Moving forward, the integration of advanced AI models with clinical knowledge not only increases the accuracy of diagnosis but also enables earlier intervention and more personalized treatment plans, potentially improving patient outcomes significantly. This study underscores the importance of continued innovation in this field, as well as the need for collaboration between AI researchers and healthcare professionals to develop tools that are both scientifically sound and clinically applicable. While this journey is challenging, it also represents a significant opportunity to reshape the future of healthcare, where technology and medicine work hand-in-hand to solve some of the most daunting challenges. As we continue to push the boundaries of what is possible with AI in healthcare, the potential for these technologies to revolutionize patient care becomes ever more apparent. This study lays the groundwork for future research and clinical applications, emphasizing the need for continued innovation and collaboration in this rapidly evolving field, ultimately striving towards a future where early diagnosis and tailored treatments are the norm, significantly improving the quality of life for patients with neurological disorders.

References

- 1 S. Chen, H. Chen, K. Wang, C. P. Association and C. N. Association, *Chinese Journal of Neurology*, 2021, **54**, 762–771.
- 2 Z. Zhang, G. Li, Y. Xu and X. Tang, *Diagnostics (Basel)*, 2021, **11**, 1402.
- 3 S. Marino, R. Ciurleo, G. D. Lorenzo, M. Barresi, S. D. Salvo, S. Giacoppo, A. Bramanti, P. Lanzafame and P. Bramanti, *Neural Regen Res*, 2012, **7**, 611–619.
- 4 P. U. Staff, *When Will There Be a Cure for Parkinson's?*, <https://www.parkinsons.org.uk/research/when-will-there-be-cure-parkinsons>, Accessed: 2024-08-09.
- 5 P. L. Alvarez, *Overview of Parkinson's Disease*, <http://www.valleyneurology.net/parkinsons.html>, Accessed: 2024-08-09.
- 6 G. Pagano, F. Nicolini and M. Politis, *Clin Med (Lond)*, 2016, **16**, 371–375.
- 7 P. Faragó, A. Ștefăniță, C. G. Cordoș, L. I. Mihăilă, S. Hinteș, A. S. Peștean, M. Beyer, L. Perju-Dumbravă and R. R. Ileașan, *Bioengineering (Basel)*, 2023, **10**, 531.
- 8 Q. Al-Tashi, M. B. Saad, A. Muneer, R. Qureshi, S. Mirjalili, A. Sheshadri, X. Le, N. I. Vokes, J. Zhang and J. Wu, *Int J Mol Sci*, 2023, **24**, 7781.