Deep Transfer Learning Based Parkinson's Disease Detection Using Optimized Feature Selection

[1]Dillikar Santosh,[2]Dr. M. Raju, [3]Dr. N. Lakshmi Priya

[1]Department of CSE, Nalla Malla Reddy Engineering College, Telangana, India.

[2]Associate Professor, Department of CSE, Nalla Malla Reddy Engineering College, Telangana, India.

[3] Assistant Professor, Department of CSE, Nalla Malla Reddy Engineering College, Telangana, India.

Abstract: Parkinson's Disease (PD) is difficult to identify early on since there are no physical diagnostics. This initiative addresses the critical need for a non-invasive Parkinson's disease detection method. This study uses deep learning, primarily CNNs, to discover Parkinson's disease patients by analyzing handwriting trends. In classification jobs, deep learning has shown to be very accurate, and it has also been useful in medical areas for analyzing data like X-rays and MRI scans. The project's main goal is to improve accuracy by taking traits from handwriting, teaching machine learning models, and checking how well they work against old-fashioned methods. This project boosts its skills by adding the Xception algorithm for better feature extraction, in addition to VGG19, InceptionV3, and ResNet50 for feature extraction. A strong voting algorithm is added to the classification process to make it stronger. A Flask system that works with SQLite has been created to make user testing and real-world use easier. This project not only makes the project bigger, but it also makes sure that the interface is easy to use so that it can be tested and proven in clinical situations.

Index terms - *Parkinson's disease, neurological disorder, handwritten records, transfer learning, deep learning.*

1. Introduction

No cure exists for Parkinson's disease (PD), a neurological disorder. This occurs when brain dopamine levels plummet. One neurotransmitter is dopamine. It sends signals to the basal ganglia, which governs balance and movement. When cells in the basal ganglia that make dopamine die or get damaged, the amount of dopamine in the brain drops. Parkinson's signs can include shakes, stiff muscles, trouble with balance and posture, uncontrollable movements (dyskinesia), speech and writing changes, and movement that is limited or slow (Bradykinesia) [1].

Clinical testing like blood tests are few, making Parkinson's disease difficult to diagnose. PD is more common in those over 60, however it may start early and go undiagnosed. Early detection makes it simpler to manage symptoms and halt disease progression [2]. Early-onset PD may induce finger twitches and speech and movement pauses. Parkinson's patients' finger spasms create constricted handwriting. Mycrographia is the name for this type of handwriting, and it can be very helpful in finding people with PD early on. People who have PD can be identified by looking for certain patterns in their writings that show they have mycrographia or other deformations.

Deep learning techniques have recently become very good at solving classification problems. CNNs and other deep learning systems have been shown to be the most accurate ways to classify things. CNNs have been used a lot to sort pictures, sounds, and movies into groups. CNNs take given data and find unique patterns that are then used to make the final classification. CNNs are straightforward to discover and utilize, making them ideal for categorization. Studies have revealed that deep learning approaches outperform machine learning methods due to transfer learning. After training CNNs for new tasks, one or more layers are added [3].

Examples include ResNets, EfficientNets, MobileNets, and other deep learning models. Medical practitioners have long employed deep learning methods. Deep learning models evaluate X-rays and MRIs to identify. AI has been more useful in medicine over the last decade as it has improved. AI is already used to diagnose and forecast some diseases in medicine. Deep learning works better than traditional approaches, according to studies [4].

Using deep learning techniques to find PD in handwriting data can be helpful, as these techniques have hit very high levels of accuracy.

One way to get results is to feed a deep learning model picture data that includes handwriting samples from people who are affected and people who are not affected. In our model, we suggested using genetic algorithms, k-nearest neighbors, and deep transfer learning models to create a system that can quickly tell whether a patient is healthy or has Parkinson's disease by reading handwriting records and pulling features.

2. Literature Survey

Millions suffer from Parkinson's disease, a neurological disorder that can't be cured. One of the toughest medical tasks is automatically detecting Parkinson's disease in characteristics. Many characteristics in these datasets are useless or include noise that makes them difficult to utilize. This makes learning and calculations tougher. A superior discrete artificial bee colony technique is used to propose a mixed feature selection approach [2]. This ensures categorization efficiency. The algorithm combines the best filters and wraps to remove most unrelated or noisy data and discover the best set of features. The filter pre-ranks features using three variable ranking algorithms. Based on the importance of the re-ranked qualities, the artificial bee colony population is established. The artificial bee colony technique scores individuals (feature subsets) in the wrapping section depending on how effectively they categorize using the classifier to get the optimal subset [2]. The first way to automatically choose the quickest best predictor in the search framework is also shown. The recommended strategy outperforms state-of-the-art algorithms and extracts less valuable features than free datasets.

This study talks about how deep learning can be used to make medical diagnoses [4]. A full study of different scientific writings has been done on the topic of using deep neural networks in medicine. There were more than 300 study pieces found, and 46 of them were shown in more detail after going through a series of selection steps. The outcomes show that convolutional neural networks (CNN) [7, 12, 15] are the most common type used for deep learning and medical picture analysis. Also, this piece shows that deep learning technology is used in many areas, but most of them are related to science, medical analysis, and other similar fields.

Older people's second most prevalent progressive condition is Parkinson's disease (PD). The most crucial issue about PD is brain function loss. This greatly impacts body function. Clinical indications are the main method physicians detect PD early, thus their experience matters. For an additional Parkinson's disease diagnosis system [5], this research focuses on machine learning. For PD group choices, the KNN, Random Forest, and Naive Bayesian algorithms are utilized. Also offered is an enhanced KNN algorithm with information entropy. The comparative findings from planned clinical data experiments suggest that this strategy improves prediction accuracy and is feasible.

People with Parkinson's disease have movement problems that make their quality of life worse. The disease is a worsening brain disease. At the moment, there is no cure for this disease. This study [8] suggests a way to find people with Parkinson's disease by mixing voice and written data and using multimodal analysis. We show how to use picture transformer designs to tell the difference between people with Parkinson's disease [1, 5, 6, 7, 8] and healthy people. Finding Parkinson's disease early is important for managing movement symptoms. A data-efficient picture transformer using self-supervised learning on DINO was able to get more than 90% correct on a set of spiral and meander drawings from the NewHandPD dataset. On the other hand, an audio spectrogram generator got results of more than 80% accuracy on the continuous vowel sounds of /a/ and /o/ from the PC-GITA collection. This study looks at how to use a mixed method to find Parkinson's disease and how transformer designs can be used in tasks involving picture and audio spectrogram classification.

Parkinson's disease (PD) affects millions of people around the world, mostly older people. It is one of the most common neurological illnesses. Handwriting problems can be a good early sign of this disease, as has been shown. The point of this study [9] was to suggest a quick and easy way to tell the difference between people with PD and healthy controls by giving them drawing tasks that use machine-learning methods. Our telemonitoring system was based on an easy-to-use app for drawing tablets. It let us see where the digital pen was placed, how much pressure it was applying, and how inclined it was while the experiment was going on. At the same time, we showed the subject what was happening on the screen. We made a routine with drawing and writing tasks, some in Italian, and got information from 22 healthy people and 9 people with Parkinson's disease [9, 11, 13]. Using signs and data from an existing database, we created a machine-learning model that can instantly tell the difference between 77.5% of PD cases and healthy test subjects.

The world's most prevalent neurological movement disorder is Parkinson's. A loss of dopaminergic neurons in the substantia nigra indicates it. Current PD diagnosis based on Parkinsonism symptoms may lead to incorrect diagnoses. Electroencephalographic (EEG) recordings of non-surgical PD patients may be utilized as an alternative. This work suggests a deep learning-based automated PD detection methodology [11]. The research examined EEG data from 16 healthy and 15 PD patients. The Gabor transform created spectrograms from EEG [11, 14]. The recommended 2D-CNN model was trained using these spectrograms. The proposed model achieved 99.46% (±0.73) classification accuracy for three classes: healthy controls, PD patients with and without medication, and PD patients. This suggests that the model might immediately discover and assess PD patients' medication states. The proposed model may now be validated on a larger database before being utilized as a CAD tool to assist physicians make choices.

3. Methodology

i) Proposed Work:

Our cutting-edge method for finding Parkinson's disease uses handwriting analysis with Convolutional Neural Networks (CNNs) [7, 12, 15] in a way that is non-invasive and easy for anyone to use. Our method improves the accuracy of PD recognition by putting the comfort of the patient first and using deep learning innovations like VGG19, InceptionV3, and ResNet50 algorithms. For feature extraction, these deep learning methods are used. Adding machine learning models like K-Nearest Neighbors, Support Vector Machines, and others that use features taken from deep learning models and tested on the NEWHANDPD dataset makes it even better at changing the way Parkinson's disease is diagnosed. In addition to using VGG19, InceptionV3, and ResNet50 [16] for feature extraction, the Xception method is added to this project to make it even more useful. A strong voting algorithm is added to the classification process to make it stronger. A Flask system that works with SQLite has been created to make user testing and real-world use easier. This addition not only makes the project bigger, but it also makes sure that the interface is easy to use for testing and validating in clinical situations.

ii) System Architecture:

The method starts by choosing the NEWHANDPD dataset [7, 22], which has a variety of handwriting examples from people with Parkinson's disease. Using an image data creator and image processing methods, the written pictures are improved and prepared for use. Deep learning methods, such as VGG19, InceptionV3, ResNet50, ensemble, and the extension model xception, take processed pictures and pull out traits that are useful. The features that were taken are then used to create and train machine learning models that use methods such as K-Nearest Neighbors (KNN) [9], Support Vector Machines (SVM), the extension-voting classifier, and others. Performance measures are used to judge how well the system works, which makes sure that the labeling of Parkinson's disease is correct and reliable. This unified method uses picture processing, deep learning, and regular machine learning to find everything.

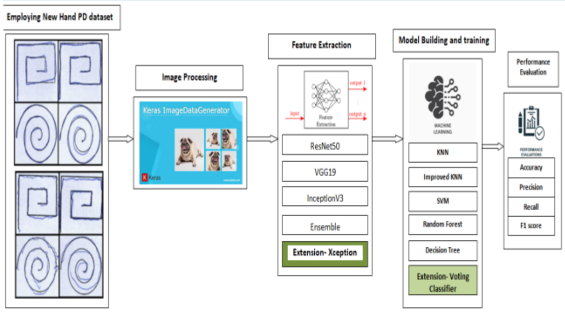


Fig 1 Proposed Architecture

iii) Dataset collection:

This step, we really look into the NEWHANDPD dataset [7, 22] to fully understand its structure and what it contains. This research is important for making smart choices during the next steps of data processing; it gives us useful information that guides our analysis.

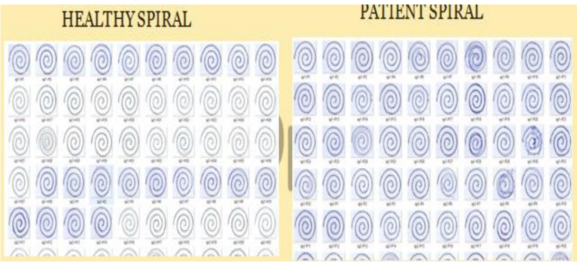


Fig 2 Dataset images

iv) Image Processing:

Several important steps are involved in image processing, which is a key part of how autonomous driving systems find objects. In the first step, the original picture is turned into a blob object, which makes it easier to analyze and change later. After that, the classes of items that need to be found are set, which makes it clear what groups the method is trying to find. At the same time, bounding boxes are set up to show where the items are supposed to be in the image's areas of interest. The data that has been handled is then turned into a NumPy array. This is an important step for quickly computing and analyzing numbers.

In the next step, information from large datasets is used to load a model that has already been trained. This includes reading the network levels of the pre-trained model, which have learned values and traits that are necessary for accurate object recognition. Also, output layers are taken out, which gives final forecasts and makes object separation and classification work well.

In the image processing chain, the picture file and the annotation file are also added together, which makes sure that there is enough information for further analysis. By switching from BGR to RGB, the color space is changed, and a mask is made to draw attention to important parts. Lastly, the picture is shrunk so that it can be used more efficiently for research and processing. This all-around image processing approach builds a strong base for reliable and accurate object recognition in the changing environment of self-driving cars, which improves road safety and the ability to make decisions.

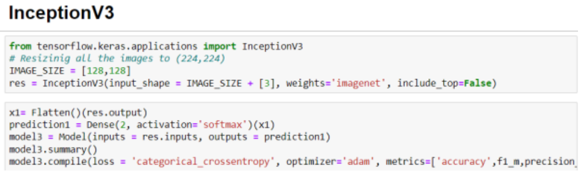
v) Feature Extraction:

When you turn raw data into number features that can be handled, you're still keeping the information in the original data set. This is called feature extraction. When you use machine learning on the raw data, this method gives you better results. Feature extraction can be done by hand or automatically:

To do manual feature extraction, you have to find the features that are important for a problem, describe them, and then set up a way to separate those features. Having a good understanding of the background or subject can help you choose which features might be useful in many cases. Engineers and scientists have worked for decades to come up with ways to pull features from pictures, signals, and text. The mean of a window in a signal is an example of a simple trait. Specialized algorithms or deep networks are used in automated feature extraction to pull out features from data or images without any help from a person. When you want to quickly go from raw data to making machine learning models, this method can be very helpful. Automated feature extraction can be seen in wavelet scattering. With the rise of deep learning, the first layers of deep networks have mostly taken the place of feature extraction, but mostly for picture data. For signal and time-series uses, feature extraction is still the first problem that needs a lot of skill to solve before prediction models can be built that work well.

vi) Algorithms:

InceptionV3 is a deep convolutional neural network model made to sort images into different groups. It quickly records complex patterns in pictures by using a mix of parallel and multi-scale convolutional processes. This makes it perfect for complex feature extraction in our project to find Parkinson's disease. Its ability to handle different kinds of spatial information lets it accurately analyze drawing patterns, which helps the system find signs of Parkinson's disease [16].



InceptionV3

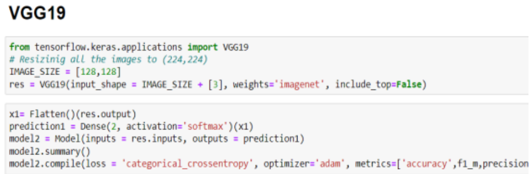
```
from tensorflow.keras.applications import InceptionV3
# Resizinig all the images to (224,224)
IMAGE_SIZE = [128,128]
res = InceptionV3(input_shape = IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

```
x1= Flatten()(res.output)
prediction1 = Dense(2, activation='softmax')(x1)
model3 = Model(inputs = res.inputs, outputs = prediction1)
model3.summary()
model3.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy',f1_m,precision_
```

Fig 3 Inception V3

VGG19 is a deep convolutional neural network design that is well-known for how well it works at recognizing images. With its 19 layers and use of small 3x3 convolutional filters and max-pooling layers, it can pull out complex data from pictures.

The deep structure of VGG19 makes it easier to capture hierarchical features, which are necessary for studying complex patterns in writings from people with Parkinson's disease. VGG19 is a good choice for improving the efficiency of our Parkinson's disease identification system through effective feature extraction [16]. It has been shown to work well in picture classification tasks and can be used with a variety of datasets.



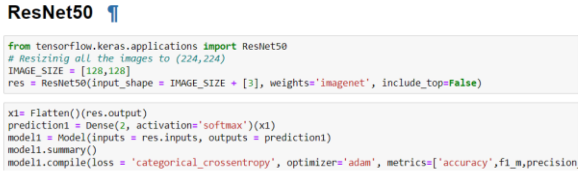
VGG19

```
from tensorflow.keras.applications import VGG19
# Resizinig all the images to (224,224)
IMAGE_SIZE = [128,128]
res = VGG19(input_shape = IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

```
x1= Flatten()(res.output)
prediction1 = Dense(2, activation='softmax')(x1)
model2 = Model(inputs = res.inputs, outputs = prediction1)
model2.summary()
model2.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy',f1_m,precision
```

Fig 4 VGG19

ResNet-50, which stands for Residual Network with 50 layers, is a deep convolutional neural network design that is well-known for how well it works at classifying images. It uses residue learning to help the network figure out how to deal with the disappearing gradient problem that comes up in deep networks. When skip links are used, information can move directly between layers, which makes training much more efficient. ResNet-50 [16] was picked for this project because it is very good at pulling out complex features from handwriting patterns, which is needed for accurate Parkinson's disease diagnosis.



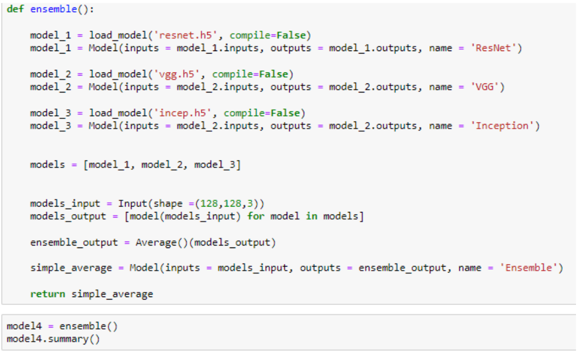
ResNet50 ¶

```
from tensorflow.keras.applications import ResNet50
# Resizinig all the images to (224,224)
IMAGE_SIZE = [128,128]
res = ResNet50(input_shape = IMAGE_SIZE + [3], weights='imagenet', include_top=False)
```

```
x1= Flatten()(res.output)
prediction1 = Dense(2, activation='softmax')(x1)
model1 = Model(inputs = res.inputs, outputs = prediction1)
model1.summary()
model1.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy',f1_m,precision
```

Fig 5 ResNet50

Ensembling in this project means combining the ability to predict outcomes of three different Convolutional Neural Networks (CNNs): ResNet50, VGG19, and InceptionV3. Each CNN shows a different part of the handwriting of people with Parkinson's disease. The ensemble's goal is to improve accuracy, stability, and extension by combining these models. ResNet50 is great at understanding things in depth, VGG19 is great at recognizing patterns, and InceptionV3 is great at recording spatial orders. This mixed-group method improves feature extraction and the system's ability to find small but important patterns in different handwriting samples that are indicative of Parkinson's disease [7, 25].

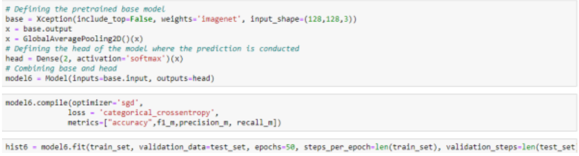


```
def ensemble():

    model_1 = load_model('resnet.h5', compile=False)
    model_1 = Model(inputs = model_1.inputs, outputs = model_1.outputs, name = 'ResNet')

    model_2 = load_model('vgg.h5', compile=False)
    model_2 = Model(inputs = model_2.inputs, outputs = model_2.outputs, name = 'VGG')

    model_3 = load_model('incep.h5', compile=False)
    model_3 = Model(inputs = model_2.inputs, outputs = model_2.outputs, name = 'Inception')


    models = [model_1, model_2, model_3]


    models_input = Input(shape =(128,128,3))
    models_output = [model(models_input) for model in models]

    ensemble_output = Average()(models_output)

    simple_average = Model(inputs = models_input, outputs = ensemble_output, name = 'Ensemble')

    return simple_average
```

```
model4 = ensemble()
model4.summary()
```

Fig 6 Ensemble

Xception, which is an extension of the Inception design, was picked for this project because it is very good at jobs that involve images. It uses depthwise separable convolutions, which improves speed by making the computations simpler. This makes it possible to get accurate features from handwriting patterns for finding Parkinson's disease, which fits with the project's focus on accuracy and using computers as little as possible [16].



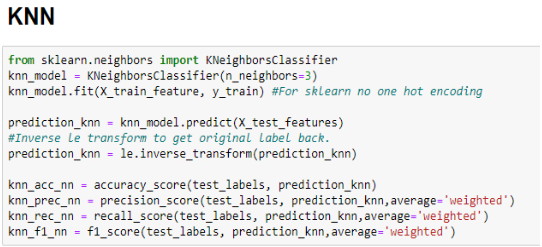
```
# Defining the pretrained base model
base = Xception(include_top=False, weights='imagenet', input_shape=(128,128,3))
x = base.output
x = GlobalAveragePooling2D()(x)
# Defining the head of the model where the prediction is conducted
head = Dense(2, activation='softmax')(x)
# Combining base and head
model6 = Model(inputs=base.input, outputs=head)
```

```
model6.compile(optimizer='sgd',
               loss = 'categorical_crossentropy',
               metrics=["accuracy",f1_m,precision_m, recall_m])
```

```
hist6 = model6.fit(train_set, validation_data=test_set, epochs=50, steps_per_epoch=len(train_set), validation_steps=len(test_set
```

Fig 7 Xception

K-Nearest Neighbors (KNN) is a machine learning method that was used to find Parkinson's disease in this project. KNN sorts data points into groups based on how close they are to other points in the feature space. When it comes to handwriting analysis, KNN uses the closeness of features taken from scribbled patterns to find patterns that are linked to Parkinson's disease. Because it is easy to use and good at finding complex connections in data, it works well for our non-invasive testing method, which means we can make accurate and easy-to-find statements about Parkinson's.

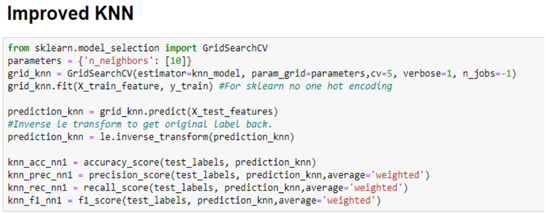


KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_knn = knn_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_knn = le.inverse_transform(prediction_knn)

knn_acc_nn = accuracy_score(test_labels, prediction_knn)
knn_prec_nn = precision_score(test_labels, prediction_knn,average='weighted')
knn_rec_nn = recall_score(test_labels, prediction_knn,average='weighted')
knn_f1_nn = f1_score(test_labels, prediction_knn,average='weighted')
```

Fig 8 KNN

In this project, the **Improved K-Nearest Neighbors** (KNN) method with Grid Search Cross-Validation is used to improve accuracy and find the best parameters. KNN works by putting a data point in the class that has the most members among its k close neighbors. Grid Search Cross-Validation carefully checks out different sets of hyperparameters to discover the best setup. This makes the model more stable and better at correctly identifying Parkinson's disease from handwriting patterns.

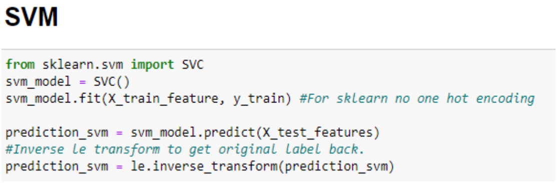


Improved KNN

```
from sklearn.model_selection import GridSearchCV
parameters = {'n_neighbors': [10]}
grid_knn = GridSearchCV(estimator=knn_model, param_grid=parameters,cv=5, verbose=1, n_jobs=-1)
grid_knn.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_knn = grid_knn.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_knn = le.inverse_transform(prediction_knn)

knn_acc_nn1 = accuracy_score(test_labels, prediction_knn)
knn_prec_nn1 = precision_score(test_labels, prediction_knn,average='weighted')
knn_rec_nn1 = recall_score(test_labels, prediction_knn,average='weighted')
knn_f1_nn1 = f1_score(test_labels, prediction_knn,average='weighted')
```

Fig 9 Improved KNN

SVM is a guided machine learning method that is used to classify things and figure out what happened in the past. SVM is used in this project because it is good at working with high-dimensional data, like the features that can be taken from handwriting patterns. SVM works by finding the best hyperplane that divides different groups the most. This makes it a good tool for figuring out trends in handwriting that could be signs of Parkinson's disease. It is a good choice for accurate disease forecast in this case because it can handle large datasets and give strong classification [14].

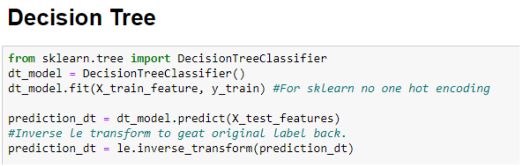


SVM

```
from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_svm = svm_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_svm = le.inverse_transform(prediction_svm)
```

Fig 10 SVM

Decision trees are predictive models that divide data into groups based on feature values over and over again, making a structure that looks like a tree. Each core node is a choice based on a trait, and each outer node is a possible result. This project is a good fit for decision trees because they are easy to understand and can find complicated connections in handwriting traits. They are great at sorting data into groups and finding patterns, which makes them useful for telling the difference between drawing patterns that show Parkinson's disease symptoms.

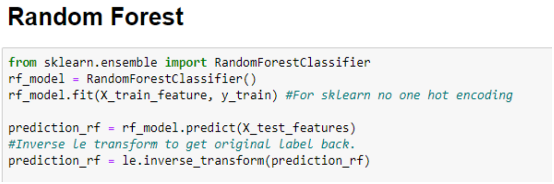


Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_dt = dt_model.predict(X_test_features)
#Inverse le transform to geat original label back.
prediction_dt = le.inverse_transform(prediction_dt)
```

Fig 11 Decision tree

Random Forest is an ensemble learning method that joins the forecasts of several decision trees to make the system more accurate and stable. Random Forest is a good choice for finding Parkinson's disease because it can handle complicated, non-linear connections in data. By combining results from different trees, it reduces overfitting and improves generalization performance. This makes it a good fit for the complex and varied patterns found in Parkinson's disease handwriting data. The method is useful for this project because it can be used in many ways and works well with datasets with a lot of dimensions.

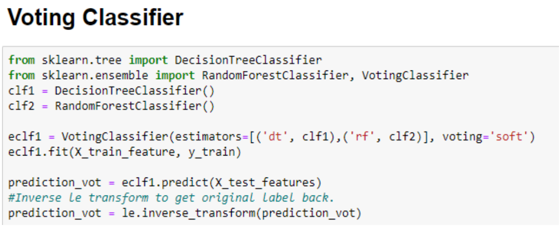


Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier()
rf_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_rf = rf_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_rf = le.inverse_transform(prediction_rf)
```

Fig 12 Random forest

The **voting classifier** is an ensemble learning method that improves performance by combining the results of several base classifiers. A soft voting classifier is used in this project. This type of classifier takes into account the weighted average of the expected values from each classifier. This method works well because it uses various models, including Decision Tree and Random Forest classifiers, each of which is good at what it does. By adding up all of their estimates, the vote classifier tries to make a stronger and more accurate system for finding Parkinson's disease. It does this by using the fact that the base classifiers work well together to make the total diagnosis accuracy better.



Voting Classifier

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
clf1 = DecisionTreeClassifier()
clf2 = RandomForestClassifier()

eclf1 = VotingClassifier(estimators=[('dt', clf1),('rf', clf2)], voting='soft')
eclf1.fit(X_train_feature, y_train)

prediction_vot = eclf1.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_vot = le.inverse_transform(prediction_vot)
```

Fig 13 Voting classifier

4. EXPERIMENTAL RESULTS

Precision: Precision is the percentage of correctly classified cases or samples compared to those that were correctly classified as hits. So, here is the method to figure out the precision:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

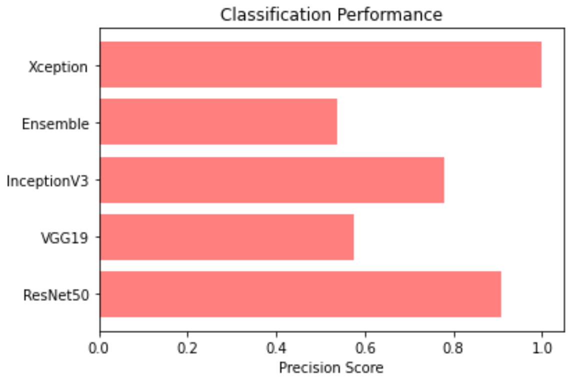


Fig 14 Precision comparison graph

Recall: In ML, recall measures how successfully a model finds all key examples of a class. This statistic measures a model's category prediction accuracy. Divide the number of accurately anticipated positive cases by the total positive instances.

$$Recall = \frac{TP}{TP + FN}$$

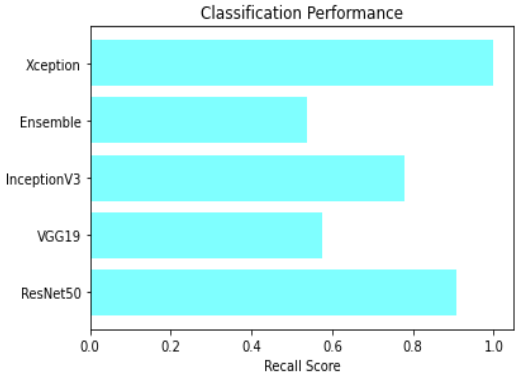


Fig 15 Recall comparison graph

Accuracy: Accuracy is the percentage of right guesses in a classification job. It shows how accurate a model's forecasts are generally.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

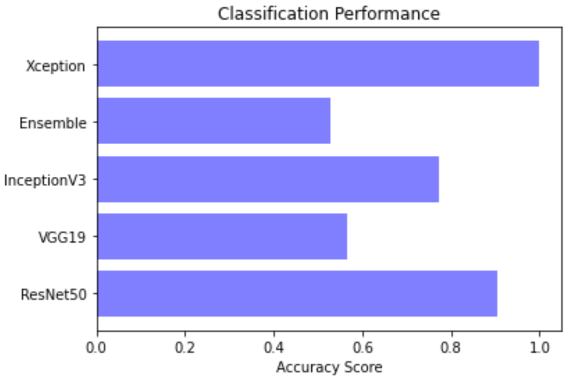


Fig 16 Accuracy graph

F1 Score: The F1 Score combines accuracy and recall into a single measure that considers both false positives and false negatives. This makes it suitable for evaluating datasets that are not evenly balanced.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

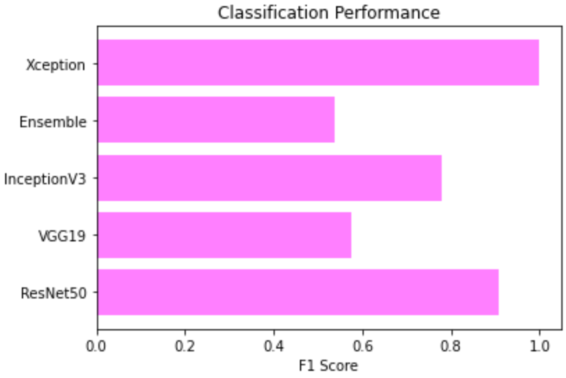
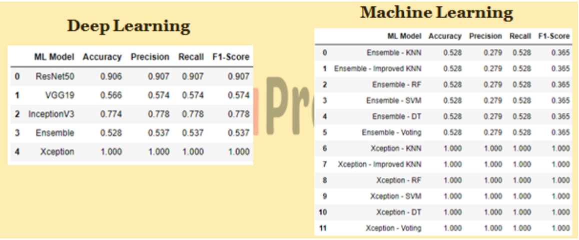


Fig 17 F1Score



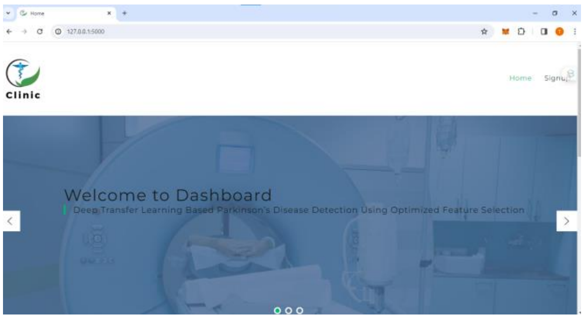
Deep Learning

	ML Model	Accuracy	Precision	Recall	F1-Score
0	ResNet50	0.906	0.907	0.907	0.907
1	VGG19	0.566	0.574	0.574	0.574
2	InceptionV3	0.774	0.778	0.778	0.778
3	Ensemble	0.528	0.537	0.537	0.537
4	Xception	1.000	1.000	1.000	1.000

Machine Learning

	ML Model	Accuracy	Precision	Recall	F1-Score
0	Ensemble - KNN	0.528	0.279	0.528	0.365
1	Ensemble - Improved KNN	0.528	0.279	0.528	0.365
2	Ensemble - RF	0.528	0.279	0.528	0.365
3	Ensemble - SVM	0.528	0.279	0.528	0.365
4	Ensemble - DT	0.528	0.279	0.528	0.365
5	Ensemble - Voting	0.528	0.279	0.528	0.365
6	Xception - KNN	1.000	1.000	1.000	1.000
7	Xception - Improved KNN	1.000	1.000	1.000	1.000
8	Xception - RF	1.000	1.000	1.000	1.000
9	Xception - SVM	1.000	1.000	1.000	1.000
10	Xception - DT	1.000	1.000	1.000	1.000
11	Xception - Voting	1.000	1.000	1.000	1.000

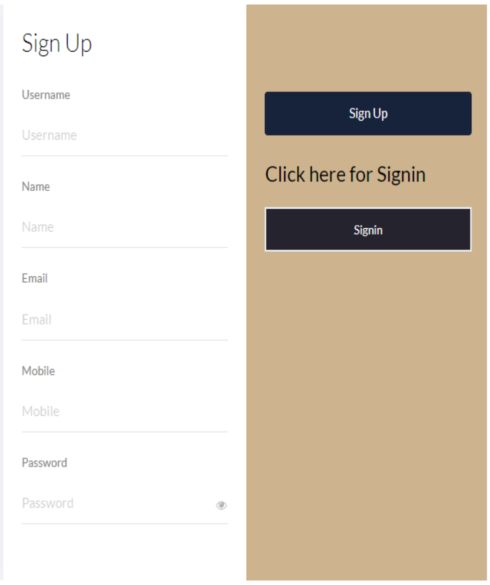
Fig 18 Performance Evaluation table



Welcome to Dashboard

Deep Transfer Learning Based Parkinson's Disease Detection Using Optimized Feature Selection

Fig 19 Home page



Sign Up

Username: Username

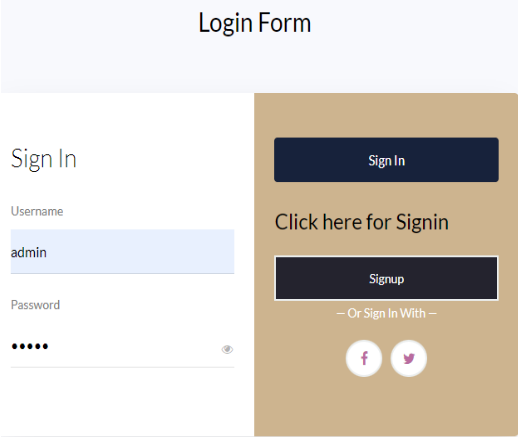
Name: Name

Email: Email

Mobile: Mobile

Password: Password

Fig 20 Registration page



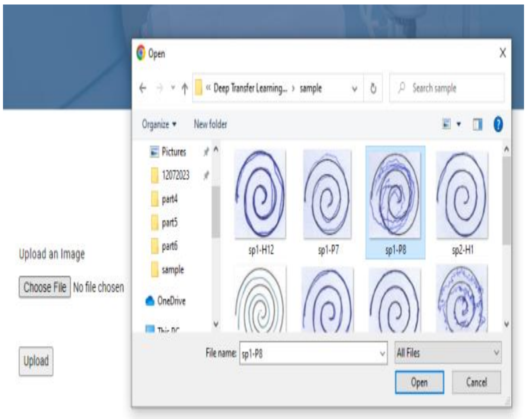
Login Form

Sign In

Username: admin

Password: •••••

Fig 21 Login page

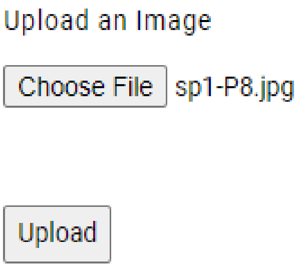


Upload an Image

Choose File No file chosen

File name: sp1-P8

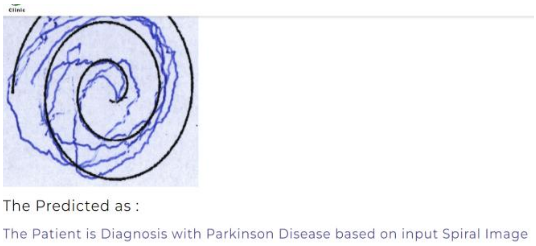
Fig 22 Input image folder



Upload an Image

Choose File sp1-P8.jpg

Fig 23 Upload input image



The Predicted as :

The Patient is Diagnosis with Parkinson Disease based on input Spiral Image

Fig 24 Predict result for given input

5. Conclusion

With the help of handwriting analysis, this project was made to help find Parkinson's disease earlier. Traditional ways of diagnosing problems have their limits, which is why people are looking for new, non-invasive ways to do things. The project's goal was to improve patient results by creating a more accurate and fast diagnosis tool for Parkinson's disease by focusing on small changes in handwriting as possible early warning signs. Deep learning models like VGG19, InceptionV3, ResNet50, and extension-Xception [16] were used in the project to get detailed information from handwriting data. Different ML algorithms are trained on the retrieved traits to make them work better. Evaluation measures were used to see how well these models could correctly diagnose Parkinson's disease, with the main goal of improving methods for the best results. All of the Xception-based models that were tested against the suggested ensemble model got perfect scores for KNN, Improved KNN, RF, SVM, DT, and Voting [14]. Xception's great performance suggests that it can pick up on minor trends in written data, which shows that it could be used to accurately diagnose Parkinson's disease. People who are at risk of or showing early signs of Parkinson's disease will benefit most from this project. The new tool could help doctors find problems early and without hurting the patient. This would allow them to act quickly and help the patient do better. The Flask framework makes sure that the front end is easy for users to understand by combining safe user registration with SQLite to protect privacy. The useful design makes it simple to enter data, see results, and make guesses. This makes the tool useful for finding Parkinson's early in hospital situations that have real-world effects.

6. Future Scope

The deep learning methods used to find Parkinson's disease will need to be improved and made even better in the future. This ongoing process aims to make the system more accurate and useful by constantly improving the algorithms [14, 16]. The merging of other types of data, like speech or gait analysis, will be looked into in future work as a way to improve testing skills. This addition is meant to give a more complete picture by including a wider range of Parkinson's disease [1, 2, 3, 4, 5, 6, 7] signs and symptoms for better accuracy. As part of future growth, real-time monitoring features will be added so that patients can be constantly checked on. This feature allows for quick actions and individual treatment plans, which can be changed as health conditions change and make managing Parkinson's disease better. In the future, large-scale clinical studies and validations will be done using a wide range of datasets that reflect different age groups and stages of disease. This thorough testing makes sure that the system works well in real life, which adds to its trustworthiness and makes it ready for broad clinical use.

REFERENCES

symptom classification using MRI images," Comput. Methods Programs Biomed., vol. 224, Sep. 2022, Art. no. 107030.

[7] M. Gazda, M. Hires, and P. Drotar, "Ensemble of convolutional neural networks for Parkinson's disease diagnosis from offline handwriting," Dept. Comput. Inform., Intell. Inf. Syst. Lab, Tech. Univ. Kosice, Ko†ice, Slovakia, Tech. Rep. 9, 2022.

[8] M. Mohaghegh and J. Gascon, "Identifying Parkinson's disease using multimodal approach and deep learning," in Proc. 6th Int. Conf. Innov. Technol. Intell. Syst. Ind. Appl. (CITISIA), Nov. 2021, pp. 1–6.

[9] M. Fratello, F. Cordella, G. Albani, G. Veneziano, G. Marano, A. Paffi, and A. Pallotti, "Classification-based screening of Parkinson's disease patients through graph and handwriting signals," Eng. Proc., vol. 11, no. 1, p. 49, 2021.

[10] A. Gold, "Understanding the mann-whitney test," J. Property Tax Assessment Admin., vol. 4, no. 3, pp. 55–57, 2007.

[11] H. W. Loh, C. P. Ooi, E. Palmer, P. D. Barua, S. Dogan, T. Tuncer, M. Baygin, and U. R. Acharya, "GaborPDNet: Gabor transformation and deep neural network for Parkinson's disease detection using EEG signals," Electronics, vol. 10, no. 14, p. 1740, Jul. 2021.

[12] S. Chakraborty, S. Aich, J. Seong-Sim, E. Han, J. Park, and H.-C. Kim, "Parkinson's disease detection from spiral and wave drawings using convolutional neural networks: A multistage classifier approach," in Proc. 22nd Int. Conf. Adv.

Commun. Technol. (ICACT), Feb. 2020, pp. 298–303.

[13] S. Nimm, S. Zarembo, K. Medijainen, P. Taba, and A. Toomela, ''Deep CNN based classification of the archimedes spiral drawing tests to support diagnostics of the Parkinson's disease,'' IFAC-PapersOnLine, vol. 53, no. 5, pp. 260–264, 2020.

[14] T. Tuncer, S. Dogan, and U. R. Acharya, ''Automated detection of Parkinson's disease using minimum average maximum tree and singular value decomposition method with vowels,'' Biocybernetics Biomed. Eng., vol. 40, no. 1, pp. 211–220, Jan. 2020.

[15] A. Das, H. S. Das, A. Choudhury, A. Neog, and S. Mazumdar, ''Detection of Parkinson's disease from hand-drawn images using deep transfer learning,'' in Proc. Congr. Intell. Syst. Singapore: Springer, Sep. 2020, pp. 67–84.

[16] L. S. Bernardo, A. Quezada, R. Munoz, F. M Maia, C. R. Pereira, W. Wu, and V. H. C. De Albuquerque, ''Handwritten pattern recognition for early Parkinson's disease diagnosis,'' Pattern Recognit. Lett., vol. 125, pp. 78–84, Jul. 2019.

[17] A. Johri and A. Tripathi, ''Parkinson disease detection using deep neural networks,'' in Proc. 12th Int. Conf. Contemp. Comput. (IC), Aug. 2019, pp. 1–4.

[18] T. Tuncer and S. Dogan, ''A novel octopus based Parkinson's disease and gender recognition method using vowels,'' Appl. Acoust., vol. 155, pp. 75–83, Dec. 2019.

[19] P. Khatamino, I. Canturk, and L. Ozyilmaz, ''A deep learning-CNN based system for medical diagnosis: An application on Parkinson's disease handwriting drawings,'' in Proc. 6th Int. Conf. Control Eng. Inf. Technol. (CEIT), Oct. 2018, pp. 1–6.

[20] C.-L. Liu, C.-H. Lee, and P.-M. Lin, ''A fall detection system using K-nearest neighbor classifier,'' Exp. Syst. Appl., vol. 37, no. 10, pp. 7174–7181, Oct. 2010.

[21] S. Katoch, S. S. Chauhan, and V. Kumar, ''A review on genetic algorithm: Past, present, and future,'' Multimedia Tools Appl., vol. 80, no. 5, pp. 8091–8126, 2021.

[22] C. R. Pereira, S. A. Weber, C. Hook, G. H. Rosa, and J. P. Papa, ''Deep learning-aided Parkinson's disease diagnosis from handwritten dynamics,'' in Proc. 29th Conf. Graph., Patterns Images (SIBGRAPI), Oct. 2016, pp. 340–346.

[23] C. R. Pereira, D. R. Pereira, F. A. Silva, J. P. Masieiro, S. A. Weber, C. Hook, and J. P. Papa, ''A new computer vision-based approach to aid the diagnosis of Parkinson's disease,'' Comput. Methods Programs Biomed., vol. 136, pp. 79–88, Nov. 2016.

[24] R. Ribani and M. Marengoni, ''A survey of transfer learning for convolutional neural networks,'' in Proc. 32nd SIBGRAPI Conf. Graph., Patterns Images Tuts. (SIBGRAPI-T), 2019, pp. 47–57.

[25] S. Xu and Z. Pan, ''A novel ensemble of random forest for assisting diagnosis of Parkinson's disease on small handwritten dynamics dataset,'' Int. J. Med. Informat., vol. 144, Dec. 2020, Art. no. 104283.

[26] M. Moetesum, I. Siddiqi, N. Vincent, and F. Cloppet, ''Assessing visual attributes of handwriting for prediction of neurological disorders—A case study on Parkinson's disease,'' Pattern Recognit. Lett., vol. 121, pp. 19–27, Apr. 2019.

[27] A. Parziale, C. A. Della, R. Senatore, and A. Marcelli, "A decision tree for automatic diagnosis of Parkinson's disease from offline drawing samples: Experiments and findings," in Proc. Int. Conf. Image Anal. Process. Cham, Switzerland: Springer, 2019, pp. 196–206.

[28] J. P. Folador, M. C. S. Santos, L. M. D. Luiz, L. A. P. De Souza, M. F. Vieira, A. A. Pereira, and A. De Oliveira Andrade, "On the use of histograms of oriented gradients for tremor detection from sinusoidal and spiral handwritten drawings of people with Parkinson's disease," Med. Biol. Eng. Comput., vol. 59, no. 1, pp. 195–214, Jan. 2021.

[29] L. Parisi, D. Neagu, R. Ma, and F. Campean, "Quantum ReLU activation for convolutional neural networks to improve diagnosis of Parkinson's disease and COVID-19," Exp. Syst. Appl., vol. 187, Jan. 2022, Art. no. 115892.