
CHAPTER 7

ENSEMBLE BOOTSTRAP AGGREGATING WITH NEURO EVOLUTION OF AUGMENTING TOPOLOGIES BASED HYBRID TECHNIQUES FOR CLASSIFYING STAGES OF AD

7.1 Introduction

The several categorization methods for AD classification are covered in this chapter. In this study, the ensemble bootstrap aggregating using NEAT is suggested. The CNN model is utilized for feature extraction in order to improve the performance of AD stage classification. As a base network, an already-trained AlexNet has been employed. For comparison, conventional feature extraction techniques such the GLCM, GLDM, and GLRLM are employed. Chapter 5 provides a thorough explanation of the AlexNet, which is utilized for feature extraction. The proposed BAGGING_NEAT classifier and the other classifiers are defined in the subsequent section of this chapter. Performance measures are used to analyze the experimental outcomes, and a final evaluation of the proposed method is provided.

7.2 Classifiers for AD stage classification

The different classifiers, including DT, NEAT, and the proposed BAGGING_NET, are covered in this section.

7.2.1 Decision Tree (DT)

A multipurpose, interpretable algorithm used for predictive modelling. The decisions are performed using the input data which is appropriate for classification. DT is a tree like arrangement with consists of an internal node, branch and leaf node. All internal nodes have attributes that relates to attribute value and all leaf node denotes prediction. The various terminologies used in DT are given below:

- i) The highest node, known as the root node, indicates the original characteristic from which the tree splits.

-
- ii) Internal node: is the decision node in the tree. The selection is based on the attribute values with branches that go to other nodes.
 - iii) Leaf node: is the terminal node with branches that predicts the outcome and there are no branches in this node.
 - iv) Branches: It is the edges that connects between nodes to display the how choices are performed in reply to certain situations.
 - v) Splitting: process of separating a node into sub-nodes with respect to the decision criteria. It consists of feature selection based on the given threshold to create data subsets.
 - vi) Parent node: it splits into child nodes from the original node.
 - vii) Child node: is produced as an outcome of a split from parent node.
 - viii) Decision condition: is to regulate, how data should be fragmented at a decision node with respect to a threshold.
 - ix) Pruning: is the elimination of nodes from a DT to enhance its overview and avoid overfitting.

In DT the key challenge is selection of attribute from root node at all level. The two key measures for attribute selection are:

- i) The amount of change in entropy is known as information gain. "S" stands for the set of instances, "A" for the attribute, "S_v" for the subset of "S," and "v" for the individual value of attribute "A." The degree of ambiguity in a random variable is known as its entropy. There is more information present when the entropy value is high.

$\text{Gain}(S,A)=\text{Entropy}(S)-\sum_{v \in \text{Values}(A)} \text{Entropy}(S_v)$ in this case. Entropy (S_v) (Payam Hosseinzadeh Kasani 2021).

The algorithm begins the entire categorization process at the tree's root node. The feature that divides the feature space most effectively is the root. To categorize testing (unknown) data, the classes are determined based on the weights computed on the features during the learning phase. The DT is built with information gain that begins with

all training instances related to the root node. Then selecting the attribute to label all nodes. Then recursively create a sub tree on the training instances. The nodes are labelled “yes” or “no” based on the positive or negative training instances. The following are the steps involved:

Step 1: Build a tree with the entire dataset covered by the root node "S."

Step 2: The Attribute Selection Measure (ASM) is used to determine which attribute in the dataset is the best.

Step-3: Divide the S into subsets that covers possible values for best attributes.

Step-4: Create decision tree node, which covers best attribute.

Step 5: Use the dataset subsets generated in Step 3 to recursively develop new decision trees. The process keeps going until there are no more node classifications. It's referred to as a leaf node or a final node.

- ii) Gini Index is the extent of inequality of distribution used in DT. It ranges [0, 0.5], where 0 denotes pure set and 0.5 denotes impure set. Calculated by subtracting the result from 1 and adding up the squared probabilities of each possible outcome in a distribution. A more homogenous or pure distribution is indicated by a low Gini index, whereas a more heterogeneous or impure distribution is indicated by a high Gini index. By calculating the difference between parent and child nodes, the Gini Index is utilized in DT to determine split quality.

By dividing the dataset according to information gain, the best attribute is identified. calculated using Entropy (parent) – [Weighted average] * Entropy (children) is the formula for information gain, where entropy represents the impurity set and the weighted average represents the number of child nodes.

7.2.2 NEAT classifier

NEAT classifier combines an evolutionary algorithm principle to create NN and change complex network structures. It requires a fitness function and a user-defined hyper-parameters. These parameters control the crossover and mutation operations. The fitness function evaluated using the genomic solutions. Initially the population is

initialized in NEAT then the input and output layers are generated. NEAT iterations consists of a different NNs where a random structure classifies the NNs. Each generation has sub-populations with similar node networks. Each class acquires NN weights consistently to its fitness function. NN weights are trained without back propagation (Sebastian Lang et al., 2021). NEAT optimizes the structure and weights of NN. The NEAT algorithm works as shows below:

- i) Initialization: initial population is generated with minimal structure.
- ii) Genetic operators: Mutation and crossover operators to create new NNs and make tiny changes to the structure and weights of NNs.
- iii) Historical markings: To have structural innovations overtime and evade structural loss during the evolutionary procedure.
- iv) Speciation: aids to prevent premature convergence and enhance the population diversity. The grouping is based on the structural connection. Each solution evolves independently in the search space to promote exploration in diverse regions.
- v) Fitness evaluation: is evaluated based on the performance. A better fitness score has a chance to move to the next generation.
- vi) Reproduction: Individuals with better fitness score are chosen to create offspring in the next generation. The structures and weights are inherited from the parent population.

These iterative steps of the NEAT classifier aids to enhance the outcome of the given task.

7.2.3 The proposed BAGGING_NEAT classifier

Ensemble learning is a subfield of ML is increasing in the recent years due to many diverse strategies for training heterogeneous components. It enhances the overview ensemble learning by categorizing each approach (Ganaie and Tanveer 2022). In this study BAGGING_NEAT is applied to classify AD into different stages. The objective of this classifier is to concurrently inspire diverse viewpoints and individual accuracy. The ensemble bagging algorithm is given below: (Donghwan Kim and Jun-Geol Baek 2022).

The combination of NEAT and BAGGING techniques enhance the classification performance in identifying AD. The bagging_NEAT algorithm applied in this study is given below:

Algorithm 1: Ensemble Bagging

Input: $[(x,y)];j=1,2,\dots,m];$
training set S

Rate of learning L

The number of machine learning classifiers T for $I=1, 2,\dots, T$

Take the nth sample out of S and learn L from S_k . $N = L(S_k)$

The result of merging classifier
 $N(x)=\operatorname{argmax}_{y \neq x} \sum y \in x 1$

End for

Ensemble N(x)

Algorithm : Bagging NEAT

Data Preparation:

Input: Data (X) and Labels (Y)

NEAT Evolution:

Initialize NEAT parameters and population

Repeat until convergence criteria are met:

Generate new neural network topologies using NEAT

Evaluate the fitness of each network using the provided dataset.

Select parents and perform genetic operations (mutation, crossover) to create offspring

Replace the population with the offspring

Bootstrap Sampling:

Input: NEAT-evolved networks (NEAT_networks)

For each bootstrap iteration:

Create a bootstrap sample by randomly selecting data points with replacement

For each NEAT network in NEAT_networks:

Train the network on the bootstrap sample

Prediction and Aggregation:

Input: Test data (X_test)

For each NEAT network in NEAT_networks:

Obtain predictions for X_test using the trained network

Ensemble Combination:

Apply an aggregation technique (e.g., majority voting or averaging) to combine the predictions of NEAT networks

Evaluation and Performance Analysis:

Input: Combined predictions and accurate labels for test data (ensemble_predictions, Y_test)

Calculate evaluation measures using ensemble_predictions and Y_test

Output: Evaluation results

7.3 Experimental result and analysis for BAGGING_NEAT classifier

The BAGGING_NEAT classifier's performance metrics are covered below. ADNI and OASIS datasets are used to evaluate the BAGGING_NEAT classifier. The proposed and the existing classification methods comparison are elaborated. Table 7.1 shows the performance metrics formulas.

Table 7.1 Performance measures formulas

Performance Measures	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Sensitivity or recall	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$
Precision	$\frac{TP}{TP + FP}$
F1-Score	$2X \frac{Precision * Recall}{Precision + Recall}$

7.3.1 Result analysis of BAGGING_NEAT classifier on the ADNI dataset

Table 7.2 and 7.3 shows the performance of GLCM, GLDM and GLRLM with BAGGING_NEAT using k=5, k=10 fold cross validation. The proposed GLCM, GLDM and GLRLM with BAGGING_NEAT method maintains a good level of performance over both k=5 and k=10 fold cross-validation. It has a k=5 accuracy of 95.8% and a k=10 accuracy of 90.5%. The classification accuracy of the proposed method which involves GLCM, GLDM and GLRLM with BAGGING_NEAT obtained accuracy of 90.5%, recall of 94.7%, specificity of 96.9%, precision of 94% and F1-Score of 94.1% when k=10-fold cross-validation. The BAGGING_NEAT method successfully outperforms existing algorithms across all performance parameters in k=5 fold cross-validation, demonstrating its effectiveness in handling fused features.

Table 7.2 Performance Metrics with Fused Features (GLCM, GLRLM, and GLDM) with BAGGING_NEAT - k=5 Fold Cross-Validation on ADNI Dataset

Algorithm	Accuracy %	Sensitivity %	Specificity %	Precision %	F1-Score %
J48	61.7	61.5	54.8	71.4	57.6
KNN	63.2	58.3	45.1	75	63.2
SVM	86.3	85.4	93.5	88.1	85.5
NEAT	92.4	92.3	81	92.5	92.8
BAGGING NEAT	95.8	96.3	98.5	95.6	95.6

Table 7.3 Performance Metrics with Fused Features (GLCM, GLRLM, and GLDM) with BAGGING_NEAT - k=10 Fold Cross-Validation on ADNI Dataset

Algorithm	Accuracy %	Sensitivity %	Specificity %	Precision %	F1-Score %
J48	58.6	61.1	52.4	68	56.4
KNN	60.9	58.1	45.1	73	63.7
SVM	84.2	85.5	92.6	88	85.7
NEAT	89	92	79.2	92.4	92.2
BAGGING NEAT	90.5	94.7	96.9	94	94.1

The performance of the several categorization techniques using BAGGING_NEAT in the same ADNI setup is compared in Table 7.4. The classification accuracy of method which involves GLCM, GLDM and GLRLM with BAGGING_NEAT obtained accuracy of 95.8% when k=5 fold cross-validation. A classification accuracy of 97.5% was achieved by the method that uses the pre-trained AlexNet with BAGGING_NEAT. The accuracy percentage obtained by the pre-trained AlexNet with BAGGING_NEAT method is more compared to the accuracy percentage obtained by GLCM, GLDM and GLRLM with BAGGING_NEAT method when k=5 fold cross-validation. This demonstrates how well the pre-trained AlexNet model with BAGGING_NEAT performs on images of brain neurons from the ADNI dataset.

Table 7.4 Performance comparison of the various classification methods with BAGGING_NEAT for AD stage classification on the ADNI dataset

Classification Method	Accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1-Score (%)
DT	61.7	71.4	54.8	61.5	57.6
KNN	63.2	75	45.1	58.3	63.2
SVM	86.3	88.1	93.5	85.4	85.5
NEAT	92.4	92.5	81	92.3	92.8
GLCM, GLDM and GLRLM + BAGGING_NEAT	95.8	95.6	98.5	96.3	95.6
AlexNet + BAGGING_NEAT	97.5	96.9	99.2	94.1	97.8

Table 7.5 compares BAGGING_NEAT model performance with the existing classification methods under the same setting on the ADNI dataset. Accuracy metric is the frequently used metric, and the other metrics that are not stated in the literature are represented as '-.' The classification accuracy of the proposed method which involves GLCM, GLDM and GLRLM with BAGGING_NEAT obtained accuracy of 95.8% when k=5 fold cross-validation. A classification accuracy of 97.5% was achieved by the method that uses the pre-trained AlexNet with BAGGING_NEAT. The accuracy percentage received by the pre-trained AlexNet with BAGGING_NEAT method is more compared to the accuracy percentage obtained by GLCM, GLDM and GLRLM with BAGGING_NEAT method when k=5 fold cross-validation. This demonstrates how well the pre-trained AlexNet model with BAGGING_NEAT performs on images of brain neurons from ADNI dataset.

Table 7.5 Performance comparison of BAGGING_NEAT classification method with the existing classification methods for AD stage classification on the ADNI dataset

Authors	Classification Method	Accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1-Score (%)
Lin et al.,(2018)	CNN	88.79	-	-	-	-
Albright et al., (2019)	CNN	86.60	-	-	-	-
Alickovic and Subasi (2020)	KNN	84.27	-	-	-	-
Shahbaz et al., (2019)	DT	74.22	-	-	-	-
Alickovic and Subasi (2020)	SVM	83.15	-	-	-	-
Chithra & Vijayabhanu (2023)	GLCM, GLDM and GLRLM + BAGGING_NEAT	95.8	95.6	98.5	96.3	95.6
(Proposed)	AlexNet + BAGGING_NEAT	97.5	96.9	99.2	94.1	97.8

7.3.2 Result analysis of BAGGING_NEAT classifier on the OASIS dataset

Tables 7.6 and 7.7 show GLCM, GLDM, and GLRLM performance metrics with BAGGING_NEAT using k=5, k=10-fold cross-validation on OASIS dataset. The proposed GLCM, GLDM, and GLRLM with BAGGING_NEAT method maintains a good level of performance over both k=5 and k=10 fold cross-validation. It has a k=5 accuracy of 94.6% and a k=10 accuracy of 87.9%. The classification accuracy of the proposed method which involves GLCM, GLDM and GLRLM with BAGGING_NEAT obtained an accuracy of 87.9%, recall of 86.9%, specificity of 88.7%, the precision of

89%, and F1-Score of 87.4% when k=10-fold cross-validation. The BAGGING_NEAT method successfully outperforms current algorithms across all performance parameters in k=5 fold cross-validation, demonstrating its effectiveness in handling fused features.

Table 7.6. Performance Metrics of GLCM, GLDM, and GLRLM with BAGGING_NEAT k=5 fold cross-validation on OASIS dataset

Algorithm	Accuracy %	Sensitivity %	Specificity %	Precision %	F1-Score %
DT	71.4	77.2	62.7	67.2	66.1
KNN	76.2	79.3	65.7	68.9	62.6
SVM	89.7	89.6	88.5	89.3	87.3
NEAT	92.6	91.6	91.4	93.7	91.7
BAGGING_NEAT	94.6	93.7	93.9	92.1	93.8

Table 7.7. Performance Metrics of GLCM, GLDM, and GLRLM with BAGGING_NEAT k=10 fold cross-validation on OASIS dataset

Algorithm	Accuracy %	Sensitivity %	Specificity %	Precision %	F1-Score %
DT	67.4	74.2	60.7	65.2	64.1
KNN	73.8	77.5	63.7	66.9	59.6
SVM	87.7	87.5	84.5	86.3	85.3
NEAT	90.6	89.6	89.1	91.3	89.8
BAGGING_NEAT	87.9	86.9	88.7	89%	87.40%

Table 7.8 compares the performance of the various classification methods with BAGGING_NEAT under the same setting on the OASIS dataset. When k=5 fold cross-validation was used, the classification accuracy of the approach that uses GLCM, GLDM, and GLRLM with BAGGING_NEAT was 94.6%. The classification accuracy of method which involves the pre-trained AlexNet with BAGGING_NEAT obtained accuracy of 95.8% respectively. The accuracy percentage obtained by the pre-trained AlexNet with BAGGING_NEAT method is more compared to the accuracy percentage obtained by GLCM, GLDM and GLRLM with BAGGING_NEAT method when k=5 fold cross-validation. This demonstrates how well the pre-trained AlexNet model with BAGGING_NEAT performs on images of brain neurons from OASIS dataset.

Table 7.8 Performance comparison of the various classification methods with BAGGING_NEAT for AD stage classification on OASIS dataset

Classification Method	Accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1-Score (%)
DT	71.4	77.2	62.7	67.2	66.1
KNN	76.2	79.3	65.7	68.9	62.6
SVM	89.7	89.6	88.5	89.3	87.3
NEAT	92.6	91.6	91.4	93.7	91.7
GLCM, GLDM and GLRLM + BAGGING_NEAT	94.6	93.7	93.9	92.1	93.8
AlexNet + BAGGING_NEAT	95.8	94.9	97.6	98.4	94.6

Table 7.9 compares BAGGING_NEAT model performance with the existing classification methods under the same setting on the OASIS dataset. Accuracy metric is the frequently used metric, and the other metrics that are not stated in the literature are represented as '-.' The classification accuracy of the proposed method which involves GLCM, GLDM and GLRLM with BAGGING_NEAT obtained accuracy of 94.6% when k=5 fold cross-validation. A classification accuracy of 95.8% was achieved by the method that uses the pre-trained AlexNet with BAGGING_NEAT. The accuracy percentage obtained by the pre-trained AlexNet with BAGGING_NEAT method is more compared to the accuracy percentage obtained by GLCM, GLDM and GLRLM with BAGGING_NEAT method when k=5 fold cross-validation. This demonstrates how well the pre-trained AlexNet model with BAGGING_NEAT performs on images of brain neurons from OASIS dataset.

Table 7.9 Performance comparison of BAGGING_NEAT classification method with the existing classification methods for AD stage classification on the OASIS dataset

Authors	Classification Method	Accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1-Score (%)
Alroobaea et al., (2021)	Random Forest	83.92	-	-	-	-
Alroobaea et al., (2021)	Logistic Regression	84.33	-	-	-	-
Islam and Zhang (2018)	DNN	93.18	-	-	-	-
Islam and Zhang (2017).	SVM	71.25	-	-	-	-
Chithra & Vijayabhanu (2023) (Proposed)	GLCM, GLDM and GLRLM + BAGGING_NEAT	94.6	93.7	93.9	92.1	93.8
	AlexNet + BAGGING_NEAT	95.8	94.9	97.6	98.4	94.6

7.3.3 Performance comparison of the proposed AlexNet with BAGGING_NEAT method on ADNI and OASIS dataset

Table 7.10 compares the performance of the AlexNet with BAGGING_NEAT method on the ADNI and OASIS datasets. The AlexNet with BAGGING_NEAT method on the ADNI dataset achieved 97.5% accuracy, 96.9% precision, 99.2% specificity, 94.1% recall, and 97.8% F1-Score. The AlexNet with BAGGING_NEAT method on the OASIS dataset has obtained 95.8% accuracy, 94.9% precision, 97.6% specificity, 98.4% recall, and 94.6% F1-Score. The ADNI dataset outperforms the OASIS dataset in terms of performance.

Table 7.10 Performance comparison of AlexNet with BAGGING_NEAT on ADNI and OASIS dataset

Datasets	Classification Method	Accuracy (%)	Precision (%)	Specificity (%)	Recall (%)	F1-Score (%)
ADNI	AlexNet + BAGGING_NEAT	97.5	96.9	99.2	94.1	97.8
OASIS	AlexNet + BAGGING_NEAT	95.8	94.9	97.6	98.4	94.6

7.3.4 Performance comparison of stage-wise analysis of AD detection using the proposed AlexNet with BAGGING_NEAT method on ADNI and OASIS dataset

Figure 7.1 illustrates the stage wise accuracy analysis of the proposed AlexNet with BAGGING_NEAT method on ADNI and OASIS datasets. Normal, stage 1, stage 2, and stage 3 are the labels assigned to the datasets. The accuracy attained for the classifications normal, stage 1, stage 2, and stage 3 in the ADNI dataset is 97.3%, 98.8%,

96.7%, and 96.8%. The accuracy attained for the classifications normal, stage 1, stage 2, and stage 3 in the OASIS dataset is 94.5%, 97.7%, 98.3%, and 89.2%, respectively. The plot demonstrates that, for the ADNI dataset, stage 1 performs better than the other stages. When it comes to the OASIS dataset, stage 2 performs better than the other stages.

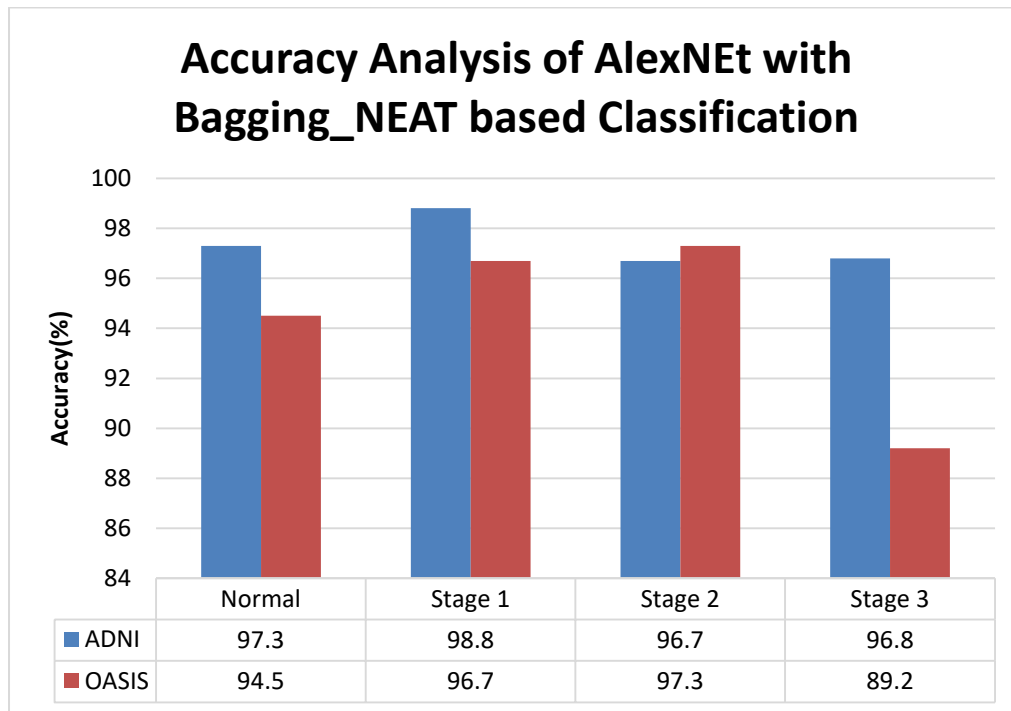


Figure 7.1 Stage wise accuracy analysis of classification using AlexNet with BAGGING_NEAT on ADNI and OASIS datasets

Figure 7.2 shows the confusion matrix for AlexNet with BAGGING_NEAT using ADNI dataset. The confusion matrix shows the distribution of predictions across 4 classes. For Normal, 1632 images were correctly classified, with 44 misclassified. Stage 1 had 884 correct classifications, with 36 misclassified. Stage 2 had 63 correct, with 10 misclassified. Stage 3 had 1120 correct, with 38 misclassified.

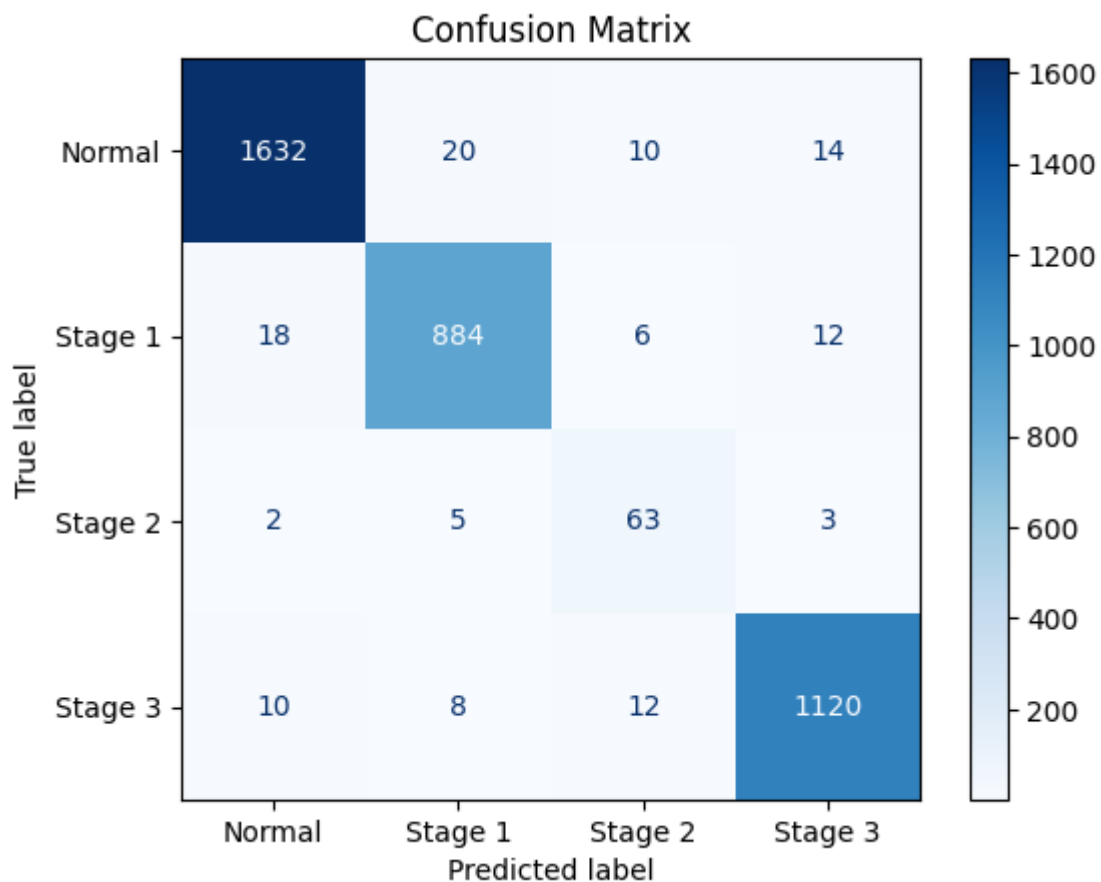


Figure 7.2 Confusion matrix for AlexNet with BAGGING_NEAT using ADNI dataset

Figure 7.3 shows the confusion matrix for AlexNet with BAGGING_NEAT using OASIS dataset. With true positives of 1,134 for Normal, 1,160 for Stage 1, 1,168 for Stage 2, and 174 for Stage 3, the classification model successfully detects the phases of AD. Thirty were incorrectly classified as Normal, twenty as Stage 1, ten as Stage 2, and three as Stage 3 among the false positives. By offering a thorough summary of the model's performance, this confusion matrix makes it possible to evaluate both its advantages and disadvantages.

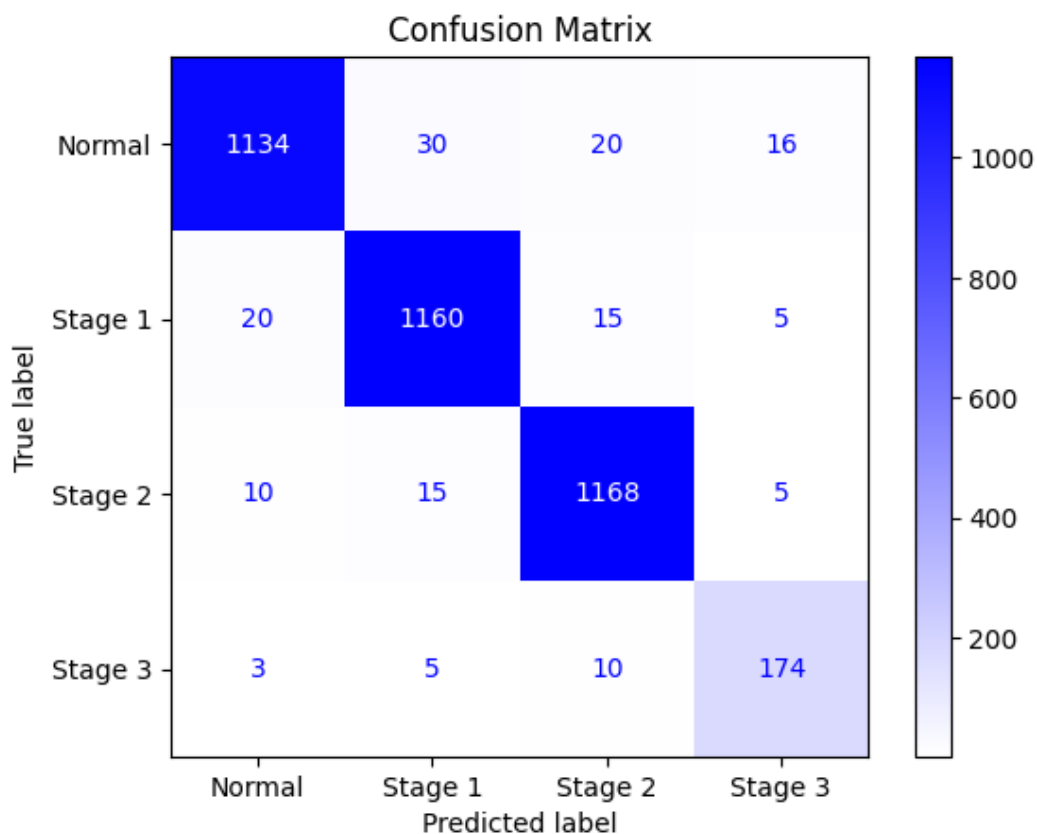


Figure 7.3 Confusion matrix for AlexNet with BAGGING_NEAT using OASIS dataset

7.4 Summary

The BAGGING_NEAT method was presented in this chapter as a way to categorize AD phases according to their severity. In this work, the ensemble of bagging with NEAT is suggested. To improve the performance in categorizing the stages of AD, a CNN model that has already been trained is used for feature extraction. The traditional feature extraction methods such as the GLCM, GLDM, and GLRLM are used for comparison. The findings of the experiment are verified and examined using performance metrics, and the evaluation and outcome analysis of the BAGGING_NEAT method are compared with previous research. Thus the accuracy of 97.5% and 95.8% is achieved in the AlexNet with the BAGGING NEAT method on ADNI and OASIS datasets respectively.