International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 5, 2025
https://theaspd.com/index.php

MRI And Environmental Data Fusion For Accurate Alzheimer's Stage Identification Via Advanced Learning Models

[1]Anusha K L, [2]Shankara Gowda S R, [3]Keerti Patil, [4]Pooja Nayak S, [5]Anoop G L, [6]Saritha D, [7]Anusha Preetham, [8]Ramya R S
[1]Assistant Professor, Department of Computer Science and Engineering,,BMS Institute of Technology and Management, Bengaluru, Karnataka, India, anushakl@bmsit.in
[2]Associate Professor, Department Of ISE,Don Bosco Institute Of Technology, Bangalore, shankaracademics@gmail.com
[3]Assistant Professor, Department of CSE (IoT,Cybersecurity including Blockchain Technology), Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India, keertipatil47@gmail.com
[4]Associate Professor, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bengaluru, Karnataka, India, Pooja-ise@dsatm.edu.in
[5]Assistant Professor, Dept of CSE, CHRIST University, Bengaluru, Karnataka, India
[6]Assistant Professor, Department of Computer Science and Engineering,,BGS Institute of Technology, Adichunchanagiri University, BG Nagara, Mandaya, Karnataka India, sarithaswamy@gmail.com
[7]Associate Professor, Department of Computer Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India, raoanushapreetham@gmail.com
[8]Associate Professor, Department of Computer Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India, dr.ramya-cse@dayanandasagar.edu

Abstract: *Alzheimer's disease is a progressive brain disorder that leads to a decline in cognitive functions. Although its progression cannot be reversed once initiated, early prediction offers an opportunity to manage and slow down the disease by targeting specific protein functions associated with its development. Despite numerous efforts using machine learning techniques for early-stage diagnosis, many studies have faced challenges in achieving reliable and accurate classification results. The efficiency of transfer learning techniques for classifying the different stages of AD is examined in this case using the ensemble stacking approach where diverse transfer learning exits. Applying the Markov random field approach to the brain tissues has an impact on AD instead of feed image extraction. The brain tissues that have been harvested are used to train the base models. The second level classification meta-model is then trained by combining the predictions of the base models. The suggested model was successful in achieving a 96% accuracy rate for disease early detection.*
Keywords: *MRI, Ensemble, Alzheimer's, Markov Random Field.*

I INTRODUCTION

Better diagnostic tools must be developed, and this thesis tackles that requirement. These tools can be effectively given by a number of recently developed machine learning algorithms that seek to increase the accuracy of AD prediction so that the patient can receive the right treatments. Using the Kaggle MRI dataset, an ensemble stacking model for the automatic detection of AD was proposed in this phase of work. Three basic models of three distinct convolutional neural networks make up the suggested model. Different MRI picture features that impact AD are used to train each model. Gray matter, white matter, and cerebrospinal fluid (CSF) are the primary brain regions impacted by AD [1,2]. In one study, the grey matter volumes of persons with AD disease and those with dementia were analyzed. When comparing dementia patients to cognitively normal people, they found that while there was a decrease in grey matter, the decline was far less than that of AD patients[3,4].

In order to distinguish AD from a variety of other neurodegenerative dementias, we sought to ascertain the diagnostic value of an expanded panel of CSF biomarkers. Using a Markov Random Field (MRF) and a Gaussian distribution, these three key characteristics that impact the disease are retrieved in this stage of work rather than feeding the complete MRI dataset into a model [5-6]. Three distinct convolutional networks are trained using the segmented brain image.

The stacking technique is an ensemble machine learning method [7-8]. Combining the predictions from many machine learning models on the same dataset involves the use of strategies like bagging and boosting. To solve this issue, a different machine learning model will be employed, which will learn when to use or trust each model in the ensemble. Unlike bagging, stacking frequently employs many models that are appropriate for the same dataset (for example, not all decision trees) (for example, in place of training dataset samples). Instead of using a series of models that correct the predictions of previous models, stacking uses a single model to determine the best way to combine the predictions from the contributing models, as opposed to boosting[9].
A stacking model's architecture is made up of two or more base models, also referred to as level-0 models, and a meta-model, sometimes called a level-1 model, which combines the base models' predictions. models whose predictions are compiled after being fitted to the training data, as opposed to Level-0 Models (Base-Models). The Level 1 (Meta-Model) model finds the best strategy to combine the predictions of the basic models[10]. To train the meta-model, the basic models' extrapolations from non-sample data are utilized. To put it another way, the base models function as input and output pairs of the training dataset that the meta-model is fitted to, using data that wasn't used to train the base models. The base models then provide the predicted outcomes and make predictions. The meta-model was fed the outputs of the underlying models. As a result, the meta-learner model will outpredict the submodel. The authors[11] used five CNN architectures for the basic model while implementing ensemble models. Every CNN model under consideration has the same structure and a 93.18% accuracy rate.

2 Proposed Model

In figure 1, the suggested ensemble stacking model is displayed. The preprocessing steps for the NIFTY MRI brain images include segmentation, normalization, bias correction, and skull stripping. Train and test datasets are created from additional preprocessed segmented images[12,13]. Three base models of convolutional neural networks with varying configurations are trained using the train dataset. In the end, the three models' findings are combined by the meta model, which also learns from the test dataset and categorizes the input images into AD, MCI, and HC. Lastly, determine statistical parameters to assess the system's performance.

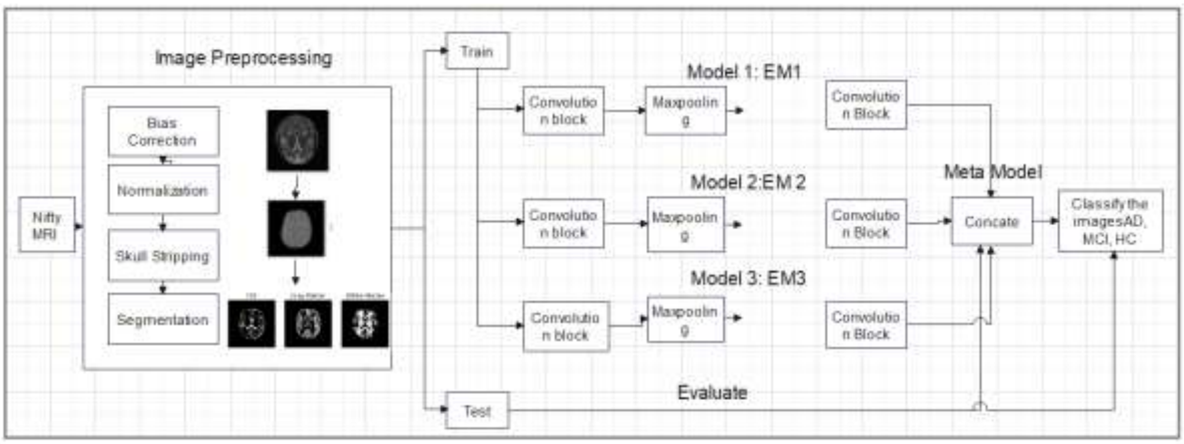


Figure 1: Proposed Ensemble Model

The proposed work's procedure is provided in Algorithm 1. First and foremost, use Gaussian and MRF algorithms to pre-process the input photos and extract the three distinct features. The model is trained using additional extracted features. The meta model is trained using the underlying model's prediction output. Lastly, use the test dataset to evaluate the model.

Algorithm 1: AD disease classification using ensemble stacking method
Input: RGB images
Output: classified and recognized images
Step 1: Prepare the input data
Step 2: Use a Gaussian distribution and a Markov Random Field to divide WM, GM, and CSF.
Step 3: Create three meta models and base models.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 5, 2025
https://theaspd.com/index.php

Step 3: Use the features that were extracted to train the base model, then use the base model's prediction to train the meta model.
Step 4: Develop a stacking classifier for the foundational models
Step 5: Use the stack ensemble model's prediction output to train the meta learner in step four.
Step 6: Accept test photos in step five and note the outcome.

2.1 Image preprocessing

In this projected work considered the MRI images collected from ADNI. Before feed images into proposed ensemble model pre-process the image to improve the process performance. In this section focus on segmenting processed brain images into WM, GM and CSF[14].

In the proposed study, WM, GM, and CSF were extracted using the Gaussian distribution and MRF. When paired with the provided data, the MRF—a stochastic process that describes a picture's local characteristics—recreates the original image. Even basic modeling of this kind can yield valuable information for the segmentation process, and the MRF of prior contextual information is a potent technique for modeling spatial continuity and other properties. A voxel's probability is dependent on its neighborhood in the MRF, which is a conditional probability model[15].

It is comparable to an energy function-based Gibbs joint probability distribution . Compared to the local conditional probabilities of the MRF, this energy function offers a more practical and organic technique for describing contextual information. On the other hand, the MRF is the best technique for sampling the probability distribution. An image's local attributes can be modeled using Markov random field (MRF)[16] theory, in which the global image properties are determined by the local interactions.

Four nearest nodes in a 2D image and six nearest nodes in a 3D image make up the first order neighborhood, whereas eight nearest nodes in a 2D image and eighteen nearest nodes in a 3D image make up the second order neighborhood. A graph G=(P,N) can be used to describe the Markov random field model, where P stands for the nodes and N for the links (also known as edges) that, based on the neighborhood relationship, connect the nodes. An picture is represented by such a network structure, in which pixels (or voxels) are represented by nodes, and the contextual dependency between pixels (or voxels) is represented by the links linking the nodes. Here, the package TissueClassifierHMRF[17,18] constructed a Gaussian Markov field.

2.2 Constructing the training model

In this experiment, three different CNN architectures for the level 0 base model were considered. EM1, EM2, or EM3 are the designations for each model. The top, middle, and bottom networks are denoted by EM1, EM2, and EM3, accordingly. The meta model takes into consideration the deep learning network. Since these architectures have already been trained on ImageNet, they perform well in classification. The suggested ensemble stacking model is shown in the figure. The proposed method considered the preprocessed images. The preprocessed images are divided into a train dataset and a test dataset with a ratio of 0.2.

The validation and train subsets of the train dataset are created. For the training dataset, the three base models are each trained independently. The ensemble predictions are then made using the stacking classifier ensemble approach with meta-learner. The detailed architecture of proposed ensemble stack model is shown in figure 2.

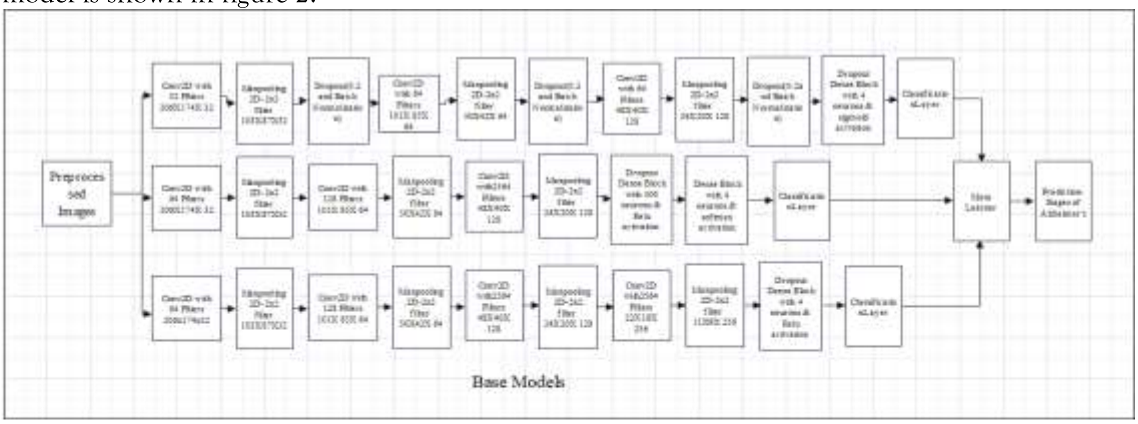


Figure.2: Ensemble Stacking Model

EM1: The EM1 algorithm consists of three 2D convolutional layers with 32, 64, and 128 filters of size 2X. Each convolutional layer was followed by maxpooling[19], dropout, and batch normalization layers. Here, remove the 0.2 percent of neurons to avoid a model being overfit. The categorization part of the system consists of flat, dense layers with four neurons for the four different classes of AD disease: non-demented, very lightly demented, mildly demented, and moderately demented. For convolutional and dense layers, use ReLU and sigmoid activation, respectively. The EM1 basic model and a screenshot of its layer summary are shown in figure 3.

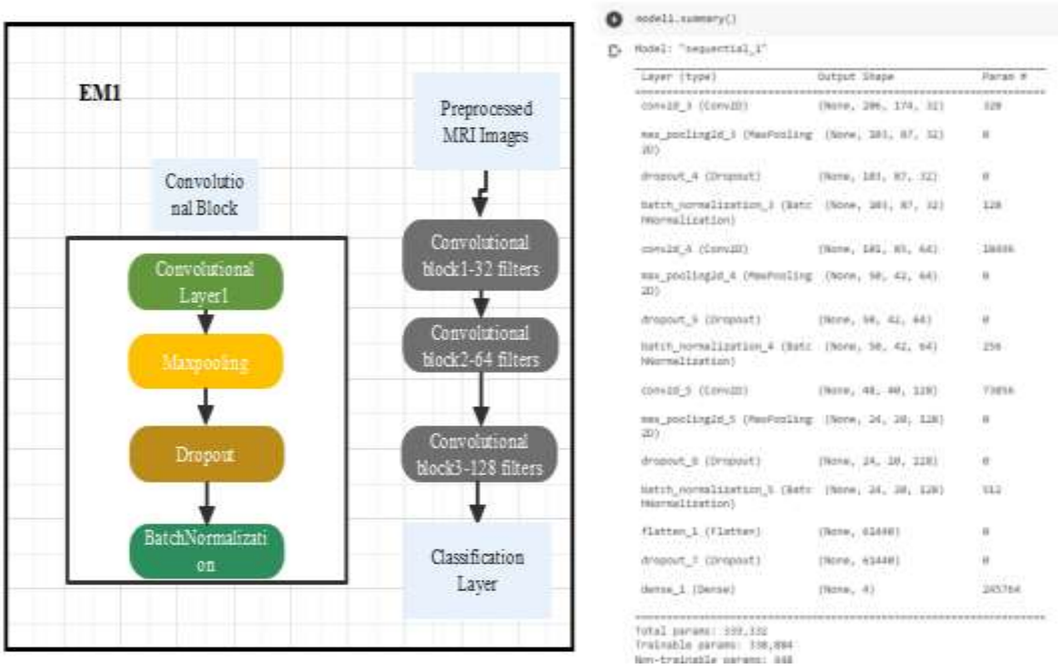


Figure 3: EM1 Model and its layer summary screenshot

EM2 Three 2D convolutional layers with 64, 128 and 256 3X3-sized filters each make up the EM2. Each convolutional layer was followed by maxpooling layers. The dropout layer is added at the conclusion of the feature extraction layers. The two flat, dense layers that make up the categorization component of the system have 500 and 4 neurons apiece. Relu and Softmax[20] are used in convolutional and dense layers. Figure 4 displays the EM2 design and an overview of its layers.

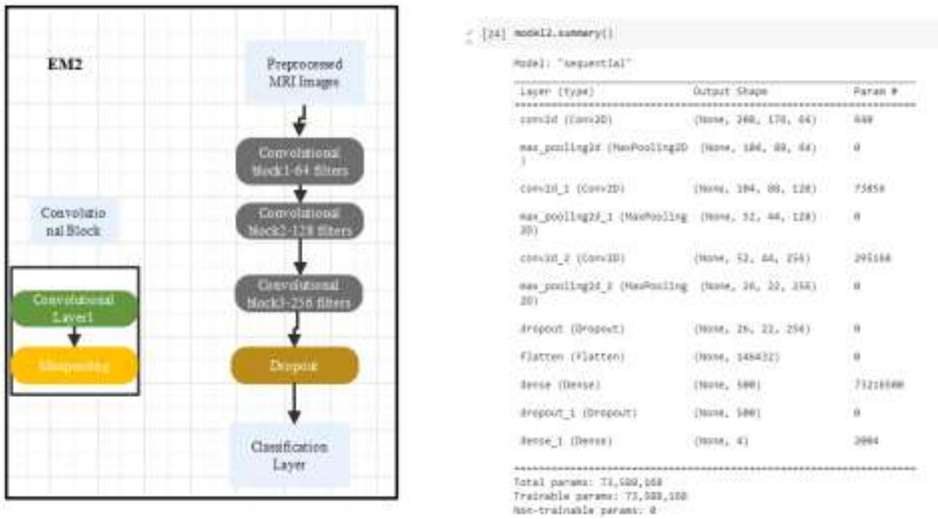
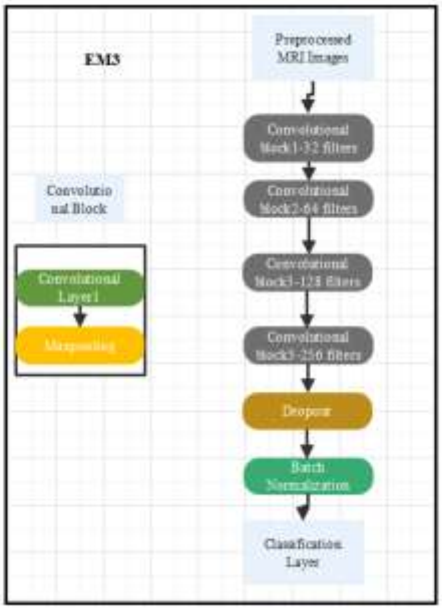
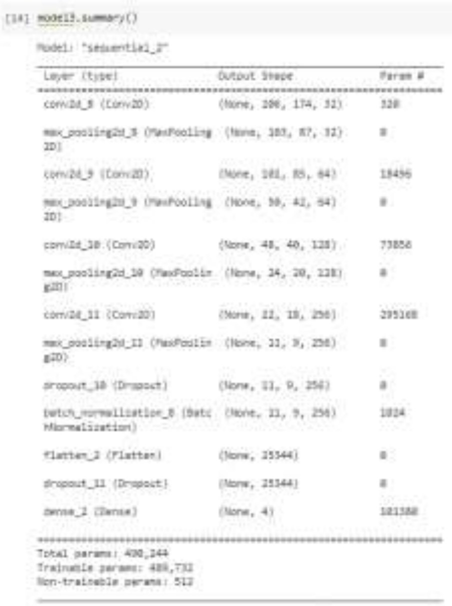


Figure 4: EM2 Model and its layer summary screenshot

EM3 Following the two convolutional layers with 16 filters in the EM3's base layer are four convolutional blocks with 32, 64, 128 and 256 filters. The convolutional blocks consist of four convolutional layers with a ReLU[21] activation function. The model's classification component consists of four dense layers with 512, 256, 128 neurons and four neurons with sigmoid activation function. Figure 5 displays the EM3 architecture and layer summary screenshot.

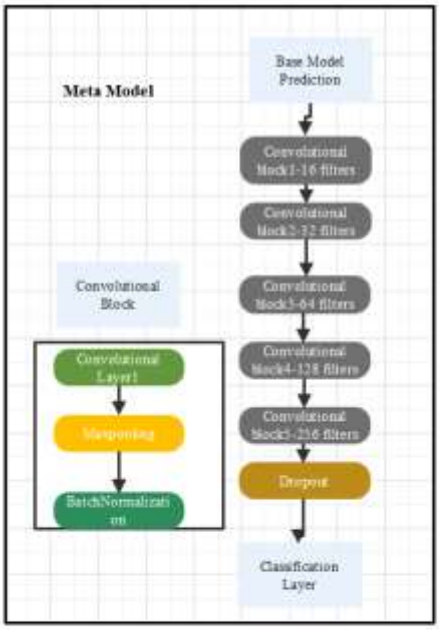




```
[14] model3.summary()

Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_8 (Conv2D)           (None, 206, 174, 32)      320

 max_pooling2d_8 (MaxPooling  (None, 103, 87, 32)      0
 2D)

 conv2d_9 (Conv2D)           (None, 101, 85, 64)       18496

 max_pooling2d_9 (MaxPooling  (None, 50, 42, 64)       0
 2D)

 conv2d_10 (Conv2D)          (None, 48, 40, 128)       73856

 max_pooling2d_10 (MaxPoolin  (None, 24, 20, 128)      0
 g2D)

 conv2d_11 (Conv2D)          (None, 22, 18, 256)       295168

 max_pooling2d_11 (MaxPoolin  (None, 11, 9, 256)       0
 g2D)

 dropout_10 (Dropout)        (None, 11, 9, 256)        0

 batch_normalization_8 (Batc  (None, 11, 9, 256)       1024
 hNormalization)

 flatten_2 (Flatten)         (None, 25344)             0

 dropout_11 (Dropout)        (None, 25344)             0

 dense_2 (Dense)             (None, 4)                 101380

=================================================================
Total params: 490,244
Trainable params: 489,732
Non-trainable params: 512
_________________________________________________________________
```

Figure 5: EM3 Model and its layer summary screenshot

The Meta Model Four convolutional blocks, each with three convolution layers and 32, 64, 128 or 256 filters, make up the meta model of the proposed system. The maxpooling layers came after each convolutional block, and then the convolutional layers. To lower the system loss, run the model for various optimizers, including Adam, Adagrad, and Rmsprop. The Adam's 97% accuracy rate shows excellent performance. Figure 6 displays the layers descriptions and the design of the meta model.



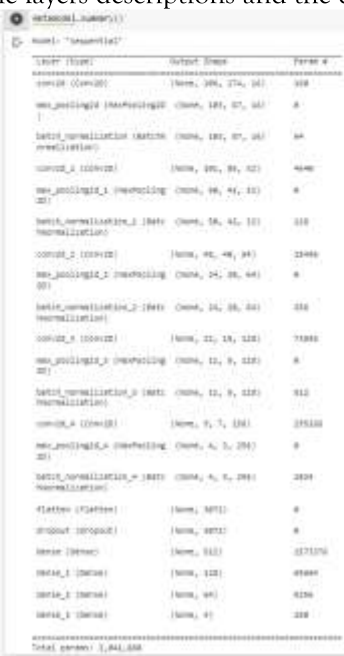


Figure 6:Meta Model and its layer summary screenshot

After setup the models , fitting each of the base model to train dataset. The sample snipet for model fitting is given below.

EM1.fit(x_traindataset,y_traindataset)
EM2.fit(x_traindataset,y_traindataset)
EM3.fit(x_traindataset,y_traindataset)

After train the base models evaluate their accuracy by using following code sample. Call prediction of EM1,EM2 and EM3 as PrEM1,PrEM2 and PrEM3 respectively.

PrEM1=EM1.predict_accuracy(x_test)

PrEM2=EM2.predict_accuracy(x_test)
PrEM3=EM3.predict_accuracy(x_test)

Finally compute the average prediction by taking average of PrEM1,PrEM2 and PrEM3.
Avgpred=(PrEM1+PrEM2+PrEM3)/3

Further train the meta model by AvgPred and Test original dataset and it is given in below.
Meta.fit(Avgpred,x_ traindataset)

At last evaluate the meta model by using test dataset.
pred3=EM3.predict_accuracy(Avgpred ,x_test)

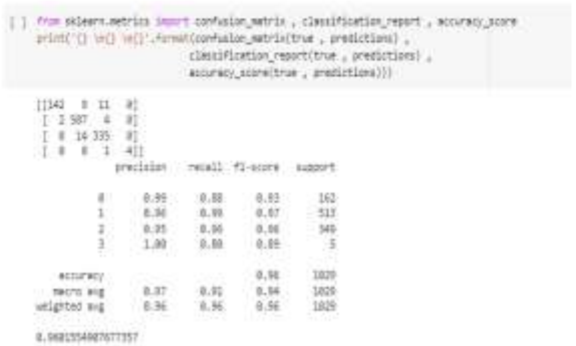
2.3 Results

Google Colab was used in the execution of this suggested project. In the experiment, four different classes of kaggle MRI data were used: non-demented, mildly demented, very mildly demented, and moderately demented. There are two distinct datasets: the train dataset and the test dataset. The base models are trained using the train dataset over 10 iterations at a learning rate of 0.0001. Following their training using AD illness images using the Adam optimizer, the level 0 models' network weight is modified using the categorical cross entropy function. Table 1 lists the parameters that were taken into consideration for this experiment.

Table .1: Parameters

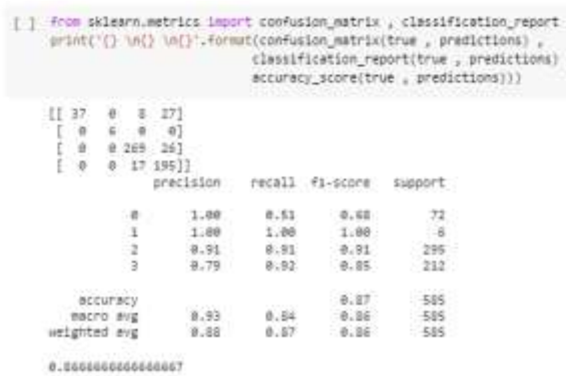
Parameters	**Model-1**
Optimizer	Adam, Rmsprop, Adagrad,
Activation Function	ReLU, Softmax and Sigmoid
Loss function	Categorical cross entropy
Batch size	15
Dataset	ADNI
Epoch	10
Learning Rate	0.0001
Normalization	Batch Normalization
Pooling	Maxpooling

Screenshots of the EM1, EM2, EM3, and meta learner's statistical performance are displayed in Figures 7, 8, 9, and 10, respectively. For EM1, EM2, and EM3, the corresponding accuracy was 96%, 86.6%, and 94%. The meta learner receives as input the stackclassifier's output, which stacks these three model predictions. Ten iterations of the meta learner should be run using different optimizers, activation functions, and cross entropy loss functions so that it can learn from the original dataset's test data as well as the prediction output of the basic model.



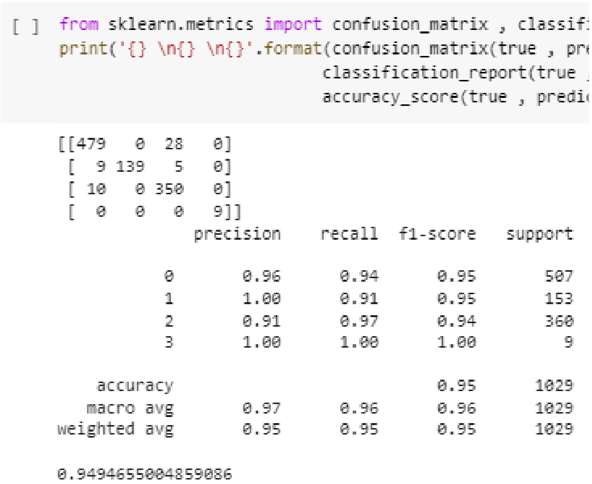
```
[ ] from sklearn.metrics import confusion_matrix , classification_report , accuracy_score
    print('{} \n{} \n{}'.format(confusion_matrix(true , predictions) ,
                                classification_report(true , predictions) ,
                                accuracy_score(true , predictions)))

[[142   8  11   0]
 [  2 507   4   0]
 [  0  14 335   0]
 [  0   0   1   4]]
              precision    recall  f1-score   support

           0       0.99      0.88      0.93       162
           1       0.96      0.99      0.97       513
           2       0.95      0.96      0.96       349
           3       1.00      0.80      0.89         5

    accuracy                           0.96      1029
   macro avg       0.97      0.91      0.94      1029
weighted avg       0.96      0.96      0.96      1029

0.9601554907677357
```

Figure 7:. Statistical performance screenshot of EM1



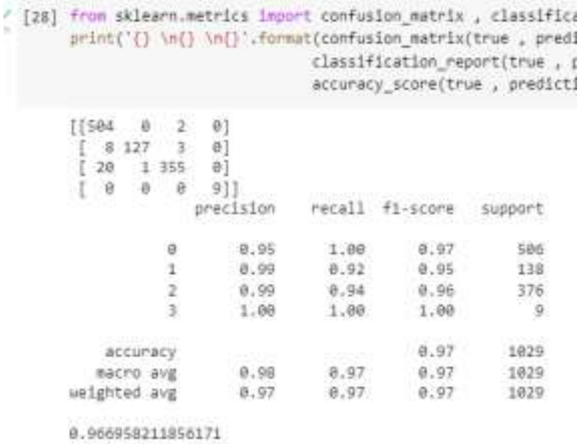
```
[ ] from sklearn.metrics import confusion_matrix , classification_report
    print('{} \n{} \n{}'.format(confusion_matrix(true , predictions) ,
                                classification_report(true , predictions)
                                accuracy_score(true , predictions)))

[[ 37   0   8  27]
 [  0   6   0   0]
 [  0   0 269  26]
 [  0   0  17 195]]
              precision    recall  f1-score   support

           0       1.00      0.51      0.68        72
           1       1.00      1.00      1.00         6
           2       0.91      0.91      0.91       295
           3       0.79      0.92      0.85       212

    accuracy                           0.87       585
   macro avg       0.93      0.84      0.86       585
weighted avg       0.88      0.87      0.86       585

0.8666666666666667
```

Figure 8: Statistical performance screenshot of EM2



```
[ ] from sklearn.metrics import confusion_matrix , classifi
    print('{} \n{} \n{}'.format(confusion_matrix(true , pre
                                classification_report(true
                                accuracy_score(true , predi

[[479   0  28   0]
 [  9 139   5   0]
 [ 10   0 350   0]
 [  0   0   0   9]]
              precision    recall  f1-score   support

           0       0.96      0.94      0.95       507
           1       1.00      0.91      0.95       153
           2       0.91      0.97      0.94       360
           3       1.00      1.00      1.00         9

    accuracy                           0.95      1029
   macro avg       0.97      0.96      0.96      1029
weighted avg       0.95      0.95      0.95      1029

0.9494655004859086
```

Figure 9: Statistical performance screenshot of EM3



```
[28] from sklearn.metrics import confusion_matrix , classifica
     print('{} \n{} \n{}'.format(confusion_matrix(true , predi
                                 classification_report(true , p
                                 accuracy_score(true , predicti

[[504   0   2   0]
 [  8 127   3   0]
 [ 20   1 355   0]
 [  0   0   0   9]]
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       506
           1       0.99      0.92      0.95       138
           2       0.99      0.94      0.96       376
           3       1.00      1.00      1.00         9

    accuracy                           0.97      1029
   macro avg       0.98      0.97      0.97      1029
weighted avg       0.97      0.97      0.97      1029

0.966958211856171
```

Figure 10: Statistical performance screenshot of Meta Model

For EM1, EM2, EM3, and the meta model, the confusion matrix findings are shown in tables 2,3,4, and 5, respectively. The EM1 model yields 93.6 percent f1 score, 90.75 percent recall, 97.5% precision, and 96% accuracy, per the performance table. The F1 score of the EM2 is 85%, its accuracy is 875, its precision is 92.4%, and its recall is 83.25 percent. EM3 exhibits 96% F1 score, 95% recall, 965 precision, and 95% accuracy.

The suggested ensemble model's performance metrics are shown in the table below. With 98.25% precision, 96.5% recall, and 97% f1 score, the meta model's accuracy grew significantly as it was trained utilizing ensemble models for both prediction and test data.

Table .2: EM1 Confusion matrix

Non Demented	Very Mild Demented	Mild Demented	Moderate demented	Accuracy	Precision	Recall	F1 Score
142	9	11	0	96%	99%	88%	93%
2	507	4	0	96/%	96%	99%	97%
0	14	335	0	96%	95%	96%	96%
0	0	1	9	96%	100%	80%	89%
Average				96%	97.5%	90.75%	93.75%

Table 3: EM2 Confusion matrix

Non Demented	Very Mild Demented	Mild Demented	Moderate demented	Accuracy	Precision	Recall	F1 Score

479	0	28	0	94.01%	100%	51.38%	67.88%
9	139	5	0	100%	100%	100%	100%
10	0	350	26	91%	91%	91%	91%
0	0	17	195	87.9%	78.6%	91.9%	84.78%
Average				**87%**	**92.4%**	**83.25%**	**85%**

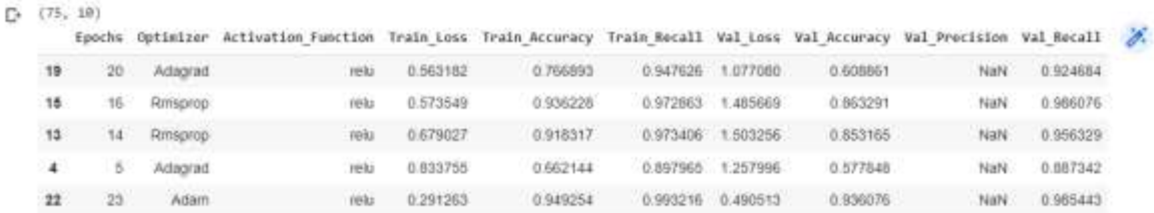
Table.4: EM3 Confusion matrix

Non Demented	Very Mild Demented	Mild Demented	Moderate demented	Accuracy	Precision	Recall	F1 Score
504	0	2	0	95%	96%	94%	95%
8	127	3	0	98.67%	100%	91%	95%
20	1	355	0	95.82%	91%	97%	94%
0	0	0	9	100%	100%	100%	100%
Average				**95%**	**96%**	**95%**	**96%**

Table 5: Meta model Confusion matrix

Non Demented	Very Mild Demented	Mild Demented	Moderate demented	Accuracy	Precision	Recall	F1 Score
504	0	2	0	97%	95%	100%	97%
8	127	3	0	97%	99%	92%	95%
20	1	355	0	97%	99%	94%	96%
0	0	0	9	100%	100%	100%	100%
Average				**97%**	**98.25%**	**96.5%**	**97%**

The meta model is iterated across 25 epochs using a variety of optimizers, such as Adam, Rmsprop, and Adagrad, in order to reduce the loss efficiency. Adam performs better in this area, and the parameter values for f1 score, accuracy, recall, and precision for random epochs with different optimizers are shown in figure 11.


(75, 10)

	Epochs	Optimizer	Activation_Function	Train_Loss	Train_Accuracy	Train_Recall	Val_Loss	Val_Accuracy	Val_Precision	Val_Recall
19	20	Adagrad	relu	0.563182	0.766893	0.947626	1.077080	0.608861	NaN	0.924684
15	16	Rmsprop	relu	0.573549	0.936228	0.972863	1.485669	0.863291	NaN	0.986076
13	14	Rmsprop	relu	0.679027	0.918317	0.973406	1.503256	0.853165	NaN	0.956329
4	5	Adagrad	relu	0.833755	0.662144	0.897965	1.257996	0.577848	NaN	0.887342
22	23	Adam	relu	0.291263	0.949254	0.993216	0.490513	0.936076	NaN	0.965443

Figure 11:Screenshot generated for different optimizers

The screenshots of accuracy and loss history for ensemble and meta models are shown in figure 12 through 19 .

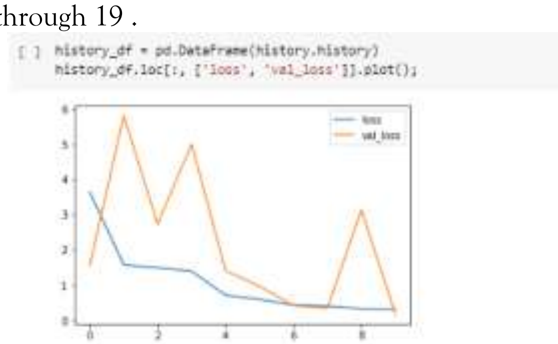


Figure.12: EM1 Loss History graph

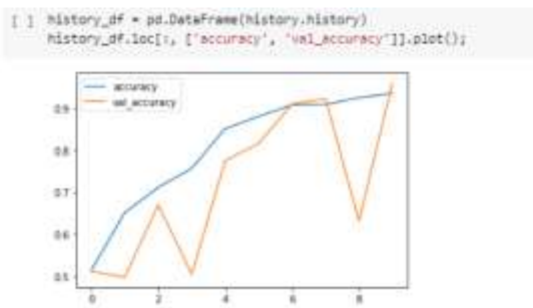


Figure 13: EM1 Accuracy History graph

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 5, 2025
https://theaspd.com/index.php

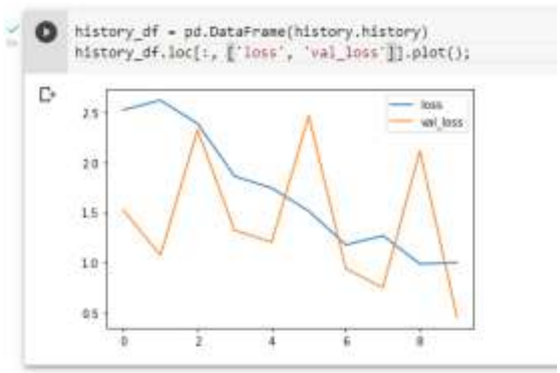


Figure 14: EM2 Loss History graph

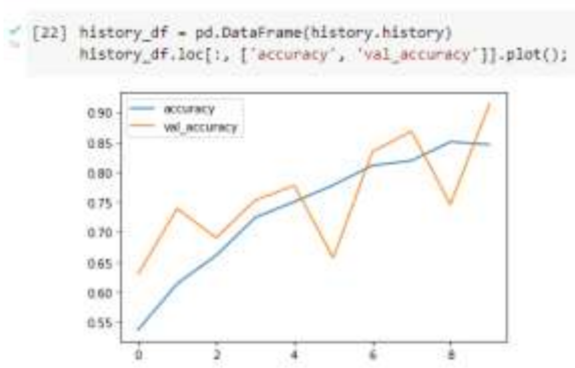


Figure 15: EM2 Accuracy History graph

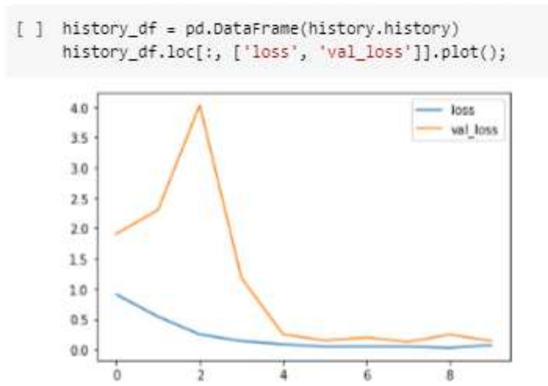


Figure 16: EM3 Loss History graph

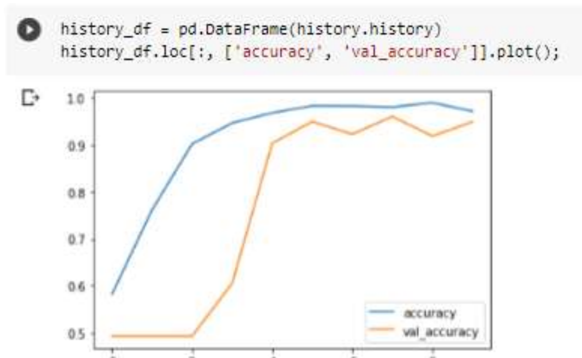


Figure 17: EM3 Accuracy History graph

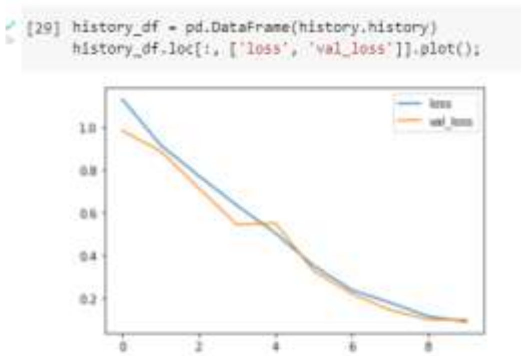


Figure 18: Meta Loss History graph

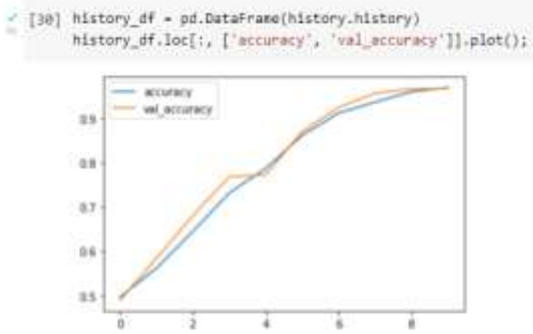


Figure 19: Meta Learner Accuracy History graph

CONCLUSION

AD is a degenerative disease of the neurons. There is currently no cure for AD or way to halt its progression. On the other hand, early dementia detection may assist families consider their own financial future. Neural network applications have been developed to use the MRI modality to classify the various phases of AD. A method is created to automatically extract features from input photos in order to classify illness stages. A small number of MRI dataset samples were used to evaluate the proposed technique. By extracting the features of input photos at two classification levels, the ensemble stacking model was implemented to enhance system performance.Three convolutional models are used to create the base model at the first level of prediction, and a meta model is used for the final prediction at the last level. The model's precision is 98.25% and its accuracy is 97%. The projected model is implemented using publicly available Python. The Intel ® Center ™ i7-8750H, 16GB RAM, 64-bit operating system, and NVIDIA GPU were used to conduct this analysis.

REFERENCES

1. Preetham, A., Ahmad, S. S., Wattar, I., Singh, P., Rout, S., Alqahtani, M. A., & Tetteh Amoatey, E. (2022). Instinctive Recognition of Pathogens in Rice Using Reformed Fractional Differential Segmentation and Innovative Fuzzy Logic-Based Probabilistic Neural Network. Journal of Food Quality, 2022(1), 8662254.
2. Hayath, S., Prathibha, G., Chaitra, P. C., Murgod, T. R., Veena Mohan, M., Hussana Johar, R. B., & Kishore, H. N. (2025). Design And Evaluation Of The System Performance Of SICA-Based Optimized Social Intelligent Agents. International Journal of Environmental Sciences, 1298-1306.
3. Nagarathna, C. R., Kusuma, M., & Seemanthini, K. (2023). Classifying the stages of Alzheimer's disease by using multi layer feed forward neural network. Procedia computer science, 218, 1845-1856.
4. Nagarathna, C. R., & Kusuma, M. (2022). Automatic Diagnosis of Alzheimer's disease using Hybrid Model and CNN. Journal of Soft Computing Paradigm, 3(4), 322-335.
5. Nagarathna, C. R., & Kusuma, M. (2021, November). Comparative study of detection and classification of Alzheimer's disease using Hybrid model and CNN. In 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON) (Vol. 1, pp. 43-46). IEEE.
6. Nagarathna, C. R., & Kusuma, M. M. (2023). Early detection of Alzheimer's Disease using MRI images and deep learning techniques. Alzheimer's & Dementia, 19, e062076.
7. Huliyappa, H. (2024). Ensemble Stacking Method of Classifying the Stages of Alzheimer's Disease by using MRI Dataset. International Journal of Artificial Intelligence, 11(2), 62-69.
8. Preetham, A., & Almas, M. (2023). Alzheimer's Classification from EGG Signals Employing Machine Learning Algorithms. Journal of Electronics and Informatics, 5(4), 386-404.
9. Nagarathna, C. R., & Mohanchandra, K. (2023). An overview of early detection of Alzheimer's disease. International Journal of Medical Engineering and Informatics, 15(5), 442-457.
10. Rangegowda, N. C., Mohanchandra, K., Preetham, A., Almas, M., & Huliyappa, H. (2023). A Multi-Layer Perceptron Network-Based Model for Classifying Stages of Alzheimer's Disease Using Clinical Data. Revue d'Intelligence Artificielle, 37(3), 601.
11. Nagarathna, C. R., Bhagya, K. G., Devaraju, H. B., Somashekara, A. S., & Kishore, H. N. (2025). Machine Learning Techniques for EEG-Based Alzheimer's Disease Classification. Mathematical Modelling of Engineering Problems, 12(3).
12. CR Jr, N., & MM Sr, K. (2022). Ensemble approach for early detection and classification of stages of Alzheimer's disease. Alzheimer's & Dementia, 18, e059124.
13. Chandana, S., Nagarathna, C. R., Amrutha, A., & Jayasri, A. (2024, January). Detection of image forgery using error level analysis. In 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) (pp. 1-5). IEEE.
14. Bhavyashree, H. L., Nagarathna, C. R., Preetham, A., & Priyanka, R. (2019, March). Modified cluster based certificate blocking of misbehaving node in MANETS. In 2019 1st international conference on advanced technologies in intelligent control, environment, computing & communication engineering (ICATIECE) (pp. 155-161). IEEE.
15. Nagarathna, C. R., Jayasri, A., Chandana, S., & Amrutha, A. (2023). Identification of Image Forgeries using Machine Learning-A Review. Journal of Innovative Image Processing, 5(3), 323-336.
16. CR Jr, N., & MM Sr, K. (2022). Ensemble approach for early detection and classification of stages of Alzheimer's disease. Alzheimer's & Dementia, 18, e059124.
17. Preetham, A., Vyas, S., Kumar, M., & Kumar, S. N. P. (2024). Optimized convolutional neural network for land cover classification via improved lion algorithm. Transactions in GIS, 28(4), 769-789.
18. Preetham, A., & Battu, V. V. (2023). Soil Moisture Retrieval Using Sail Squirrel Search Optimization-based Deep Convolutional Neural Network with Sentinel-1 Images. International Journal of Image and Graphics, 23(05), 2350048.
19. Nagamalla, V., Kumar, B. M., Janu, N., Preetham, A., Parambil Gangadharan, S. M., Alqahtani, M. A., & Ratna, R. (2022). Detection of adulteration in food using recurrent neural network with internet of things. Journal of Food Quality, 2022(1), 6163649.
20. Debnath, S., Preetham, A., Vuppu, S., & Kumar, S. N. P. (2023). Optimal weighted GAN and U-Net based segmentation for phenotypic trait estimation of crops using Taylor Coot algorithm. Applied Soft Computing, 144, 110396.
21. Rao, A., & Kulkarni, S. B. (2017, February). An improved technique of plant leaf classificaion using hybrid feature modeling. In 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 5-9). IEEE.