Decision Analytics Journal 9 (2023) 100348

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

An Alzheimer's disease classification model using transfer learning Densenet with embedded healthcare decision support system

Ahmad Waleed Saleh [a], Gaurav Gupta [a], Surbhi B. Khan [c,d,b,*], Nora A. Alkhaldi [e], Amit Verma [f]

[a] Yogananda School of Artificial Intelligence, Computers and Data Sciences Shoolini University Solan, 173229, HP, India
[b] Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon
[c] Department of Data Science, School of Science, Engineering and Environment, University of Salford, United Kingdom
[d] Department of Engineering and Environment, University of Religions and Denominations, Qom, Iran
[e] Department of Computer Science, College of Computer Science and Information Technology, King Faisal University, Al Ahsa, Saudi Arabia
[f] University Centre for Research and Development, Department of Computer Science, Chandigarh University, Gharuan, Mohali, Punjab, India

ARTICLE INFO

Keywords:
Convolutional Neural Networks
Transfer learning
Healthcare decision support system
Data augmentation
Machine learning
Deep learning

ABSTRACT

Training a Convolutional Neural Network (CNN) from scratch is time-consuming and expensive. In this study, we propose implementing the DenseNet architecture for classification of AD in three classes. Our approach leverages transfer learning architectures as the base model and showcases superior performance on the MRI dataset compared to other techniques. We use a variety of methodologies to provide a thorough study of our model. We first create a performance baseline without data augmentation, addressing the difficulties in classifying Alzheimer's disease (AD) caused by high-dimensional MRI brain scans. The improved model performance obtained through data augmentation is then highlighted, demonstrating its effectiveness in handling sparse data and assisting in generalization. We also investigate the impact of omitting particular transformations and modifying dataset split ratios, providing more insights into the behavior of the model. Through comprehensive evaluation, we demonstrate that our proposed system model achieves an accuracy of 96.5% and an impressive AUC of 99%, surpassing previous methods. This study mainly highlights the effectiveness of DenseNet architecture, current study limitations and future recommendation. Moreover, incorporating a healthcare decision support system further aid in providing valuable insights for AD diagnosis and decision-making in clinical settings.

1. Introduction

Millions of people throughout the world are afflicted by the common neurodegenerative ailment known as Alzheimer's disease [1]. Physicians typically monitor AD using the Clinical Dementia Rating (CDR) system based on signs and symptoms. Subjects are divided into three states using the CDR, Vemuri et al. [2] including CN (Cognitive Normal), MCI (Mild-Cognitive Impairment), and AD. Atrophy of the GM(Gray Matter) and WM (White Matter) and an increase in CSF (Cerebrospinal Fluid) volume are markers of AD progression. The cerebral cortex and hippocampus get smaller as the size of the ventricles in the brain grows in AD patients. Epistolaryngeal memory and spatial memory are harmed when the hippocampus's size is decreased. Planning, judgement, and short-term memory impairments are the result of this neuronal injury [3]. This decrease leads to dysfunctional synapses, damaged terminals, and further cell death.

Accurate early AD diagnosis is essential for prompting appropriate therapies and improving patient outcomes [4]. The development of machine learning and artificial intelligence has made decision support systems into potent instruments in clinical settings. These systems use AI algorithms to assess patient data, including clinical data and medical image data, and then deliver insightful recommendations to healthcare practitioners [5,6]. In order to identify illnesses and generate precise forecasts, machine learning systems examine patient data, including medical images [7–9]. This promotes early identification [5], individualized care, and better patient outcomes. Machine learning algorithms, for instance, may be trained on annotated pictures used in medical imaging to automatically detect and categorize diseases including cancer, heart issues, and neurological disorders [10]. Healthcare professionals can improve patient care by integrating AI-based decision support systems into clinical practice. These technologies can help diagnose patients more accurately, optimize treatment regimens, and optimize costs [11]. Additionally, healthcare decision support systems

* Corresponding author at: Department of Data Science, School of Science, Engineering and Environment, University of Salford, United Kingdom.
E-mail addresses: waleedsalehi@shooliniuniversity.com (A.W. Saleh), gaurav@shooliniuniversity.com (G. Gupta), s.khan138@salford.ac.uk (S.B. Khan), nalkhaldi@kfu.edu.sa (N.A. Alkhaldi), d_k.t1816@cumail.in (A. Verma).

https://doi.org/10.1016/j.dajour.2023.100348
Received 29 May 2023; Received in revised form 9 October 2023; Accepted 25 October 2023
Available online 29 October 2023

are essential for assisting medical practitioners in difficult decision-making procedures [12]. These systems offer practitioners individualized treatment suggestions and therapeutic alternatives by integrating clinical standards, evidence-based medicine, and real-time patient data [13].

One specific area of focus within healthcare decision support systems is AI-based decision support for diagnosis [14] such as leveraging machine learning algorithms where it has helped risk assessment, illness diagnosis, and categorizing patients as having a condition or not [15]. However, Deep Learning [16] on the other hand is subset of ML, presents even greater potential due to its capability to automatically learn hierarchical representations from even more complex data [17]. For instance, CNNs [18], a type of DL model, have achieved great success in medical image processing tasks, allowing for precise classification of disease. Deep learning offers considerable potential for boosting diagnostics and enhancing healthcare outcomes due to its greater capacity to extract complex patterns and characteristics from medical pictures. This ability is especially important in the realm of medical imaging [17], where high-dimensional data, such as MRI [19] or CT scans [20], calls for sophisticated processing methods. By enhancing diagnostic precision, streamlining treatment plans, and lowering healthcare costs, the use of AI-based decision support systems in clinical settings has the potential to revolutionize healthcare [21].

Another key element that further improves the capabilities of DL-based models is known as Transfer Learning (TL) [22]. Transfer learning enables the use of knowledge and representations acquired from one task or dataset on another task that is relevant. The use of pre-trained CNN models in medical imaging is made possible by transfer learning and large-scale [23] datasets like ImageNet. By leveraging the learned features and representations from these pre-trained models, healthcare practitioners can enhance the performance of their own diagnostic models, even with limited labeled medical imaging data. A pre-trained model is used as the foundation for a new model. For instance, a pre-trained model [24] is trained with large amounts of data like ImageNet [25] which consists of millions of images with more than 1000 labels. And its desirable and a good choice to transfer the domain knowledge from these models to another domain like classification task in medical area as a feature extractor of general features [26]. Since we are dealing with medical types of data its and the pre-trained models usually on different data like ImageNet, we cannot fully use the pre-trained model instead we need to add another layer of CNN as domain-specific feature extractor and modify the output layer as per domain-specific problem [27]. When compared to training with randomly initialized models, knowledge transfer requires fewer training instances to complete a different task, takes less time, and improves accuracy. Various state-of-the-art architectures such as Dense Net [28], AlexNet [29], Inception [30], VGG [31], ResNet [32] which can be fine-tuned to solve other problems with a much smaller dataset and in much less time. In our specific research, the focus lies on leveraging the capabilities of DenseNet, a prominent deep learning architecture, to develop an Alzheimer's disease classification.

This study introduces a method for classifying three stages of Alzheimer's Disease, DenseNet, a powerful DL-based architecture which can empower the decision support systems in clinical settings. By fine-tuning a DenseNet model, our goal is to extract and utilize relevant features from MR images, enabling effective differentiation between AD, NC, and MCI cases. The suggested approach has a number of benefits over conventional AD classification techniques. Firstly, it decreases the dependence on extensive labeled data, which is often scarce in medical domains. Furthermore, it makes use of the learned representations from a large-scale dataset, allowing the model to detect subtle anomalies and detailed patterns in medical images. Finally, it offers a comprehensive and precise categorization structure to support doctors in arriving at fast and correct diagnoses. This work demonstrates the use and effectiveness of our proposed model using the dataset we collected from an online repository comprising MRI scans from AD, MCI patients and healthy individuals. We evaluate the model's performance in terms of accuracy, sensitivity, specificity, and other relevant metrics, comparing it to existing approaches in the field. By contrasting it with current methods in the area, we assess the model's performance in terms of accuracy, sensitivity, specificity, and other pertinent metrics.

To thoroughly evaluate the model's effectiveness, we present four different evaluation methods. Starting with evaluation of models without applying any data augmentation technique, evaluation of Models with the use of Data Augmentation, Analysis on the basis of selective augmentation, and how the ratio of training to testing affects the performance. Our results demonstrate the potential of DenseNet-based transfer learning, offering a significant improvement in AD classification. The implications of this strategy go beyond AD diagnosis and treatment, promising improved patient care and administration. This study highlights the transformative potential of DenseNet-based transfer learning and contributes fundamentally to the field of AD classification.

The remainder of the paper is structured as follows. In Section 2, we highlight the motivation and significance of employing deep learning in AD classification. Section 3 presents the contributions of our work, emphasizing the use of DenseNet for image-based analysis and the significance of transfer learning. Section 4 provides a concise overview of related work in AD diagnosis, setting the context for our study. In Section 5, we detail the materials and methods used in our experiments. The results and analysis of these experiments are presented in Section 6. In Section 7, which is the last section, we provide the implications of our findings and provide concluding remarks.

2. Motivation

Alzheimer's disease poses a major challenge for clinicians due to its complex nature and the necessity for early identification. Traditional approaches and human intuition often fall short in accurately predicting and visualizing the progression of the disease. To overcome this issue, computationally demanding and unconventional approaches like deep learning are needed. Medical experts are increasingly relying on DL techniques for usage in both disease prediction and visualization in order to deliver foresighted and tailored treatment regimens [33]. This shift helps doctors decide on treatments and health economists analyze costs, both of which improve patients' lives.

Healthcare decision support systems in clinical settings play a vital role in leveraging deep learning techniques for accurate and effective models in AD diagnosis. Deep have emerged as leading machine learning approaches for visual object detection. While CNNs were introduced over two decades ago, despite the fact that they were first presented over 20 years ago, recent advancements in hardware, software, and network structures have enabled the training of truly deep CNNs [34]. Only lately have advancements in computer hardware, software, and algorithmic network structure made it possible to train truly deep CNNs [35]. However, as CNNs grows deeper, they encounter challenges such as information vanishing or loss of gradients along the network path [36], Addressing these challenges becomes crucial for developing accurate and effective models for Alzheimer's disease diagnosis. According to Andrew Ng in 2016 conference about [37] "Nuts and Bolts of Building AI Applications using Deep Learning" Transfer learning will succeed supervised learning as the next major factor in AI-based machine and DL commercial success. Conventional ML and DL algorithms were only designed to solve separate specific tasks, which require a lot of data. Transfer learning was developed to overcome the isolated learning paradigm. A study in 2017 [38] conducted by Lopez and Valiati, demonstrated the use of pre-trained CNNs as feature extractors for detection of tuberculosis which can produce reasonable results without a large dataset, an expensive, time-consuming training phase, or both.

In the context of AD classification, our motivation stems from the need to improve prediction accuracy and address the challenges

posed by limited training data. By exploring the application of transfer learning techniques, specifically the DenseNet architecture, we aim to leverage the learned features from pre-trained models and enhance the classification performance using MRI scans. We also extend our investigation to explore three different variants of the DenseNet architecture, evaluating their respective performances in AD classification. Our goal is to develop a robust and efficient AD classification model by DenseNet model and addressing the limitations of conventional approaches. By harnessing the power of deep learning and transfer learning, we strive to improve the accuracy of AD diagnosis, enable early identification, and enhance the decision-making process for clinicians.

3. Contribution

People with Alzheimer's disease are identified in our research work, and our goal is to identify people who may have the disease early. The dataset for AD is available on Kaggle which is used for training all patient's data using DenseNet architectures, to identify the affected people quickly and efficiently in a highly effective manner. We then examine another transfer learning paradigm to capture generic and domain-specific features and reserve using pre-trained DenseNet on ImageNet and learn new specialized representations by replacing top layers according to our output classifier. The dataset's tiny size is the biggest problem. The main obstacle that must be overcome is the amount of dataset which is small. We used data augmentation technique to help the model's function at their optimal level and dramatically enhance their effectiveness during classification. The paper can be used to infer the following contributions:

- Present an overview and significance of DenseNet architecture for image-based analysis.
- Implement three versions of DenseNet architecture, including DenseNet-121, DenseNet169, and DenseNet-201.
- We present four different evaluation methods to evaluate the model's effectiveness.
- On-fly data augmentation is used to avoid the overfitting problem.
- Incorporate a healthcare decision support system for diagnosis and decision-making in clinical settings.

4. Transfer learning

In the context of training of DL-based models, transfer learning is a powerful method that makes use of information acquired from one domain to enhance performance on a separate but related task or domain. It is especially helpful when obtaining labeled data is difficult or expensive. TL facilitates the creation of highly accurate and effective models with shorter training times by transferring acquired representations and information from pre-trained models [22]. TL has attracted a lot of interest and has shown itself to be quite successful in the field of medical imaging. CNN models which are also commonly known as ConvNet that have already been trained on large-scale datasets, models like VGG, ResNet, AlexNet, DenseNet [28] and etc. Used to capture rich visual features that are relevant across different image recognition tasks [24]. These pre-trained models have mastered the ability to recognize and extract both high-level characteristics like object patterns and structures as well as low-level elements like edges, textures, and forms [20].

Transfer learning shows considerable potential for the detection and prediction of various diseases such as Alzheimer's disease. Since it is unusual to have a dataset large enough to train an entire Convolutional Network from scratch (with random initialization), this approach is impractical World Alzheimer Report 2010 [39]. Another option is to use a Conv-Net that has already been trained on a massive dataset like ImageNet [40], as an initialization or a fixed feature extractor for the task at hand. We can use a pre-trained Convolutional Network that has been trained on a sizable dataset like ImageNet [40] rather of creating a Convolutional Network from scratch, which necessitates an impractically big dataset World Alzheimer Report 2010 [39]. Two main transfer learning scenarios typically exist:

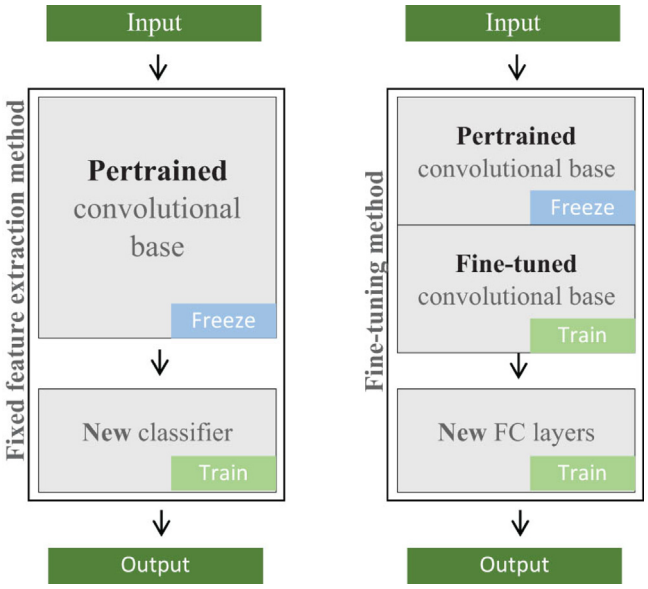

Fig. 1. Transfer learning strategies.

- **ConvNet as a fixed feature extractor**: Using a ConvNet that has been previously trained on ImageNet, we can remove the final fully connected layer and use the remaining structure as a fixed feature extractor for the target dataset. This approach capitalizes on the pre-learned representations from ImageNet, allowing us to extract relevant features without retraining the entire network [41].
- **ConvNet fine-tuning**: A different approach is to replace the classifier's final fully connected layer while also adjusting the parameters of the pretrained network. We are only able to finetune a few higher-level components of the network due to overfitting concerns. This idea is motivated by the observation that earlier features in a ConvNet contain more general features (such color blob detectors or edge detectors) that can be helpful for many tasks [42]. The network's later layers, however, become steadily more attentive on the specifics of the classes in the original dataset. Fig. 1 [43], shows the two different approaches and how they can be used, in the left we have feature extractor approach and on the right fine-tunning approach.

Fig. 2 shows the transformation of the base model to the target model. It illustrates how the pre-trained DenseNet model, which was trained on the dataset ImageNet, was modified. The pretrained model is used as a feature extractor and new layers are added on top for task-specific learning. In this method, only the weights of the newly added layers are trained for the particular task; the weights of the pre-trained model are left fixed during training. This is a typical transfer learning strategy, especially when there is a dearth of labeled data for the target task. Adding a new fully connected layer with three output nodes for the three categories of brain MRI images Mild-Demented (MID), Non-Demented (ND), and Very-Mild-Demented (VMD). This transformation enabled the model to perform the classification task on the new dataset of brain MRI images.

4.1. Significance and exploiting the potential of DenseNet architecture

CNN usage is fairly comparable to that of a standard neural network. A CNN has several layers: an input layer, an output layer, and one or more hidden layers. Convolutional, pooling, or fully connected layers serve as the hidden ones [47]. As the name suggests, ConvNet architectures are built on the explicit premise that the inputs are images, allowing us to hardcode certain characteristics. These then make it

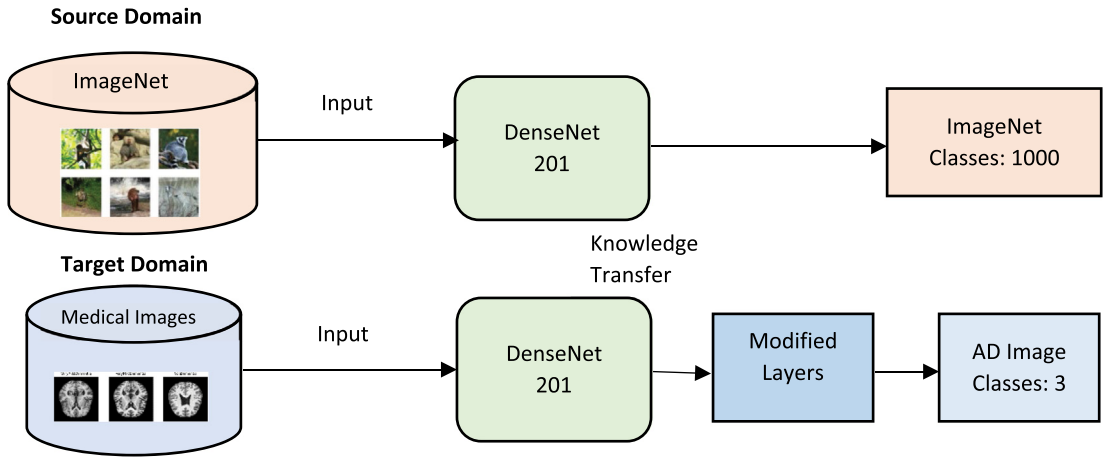

Fig. 2. Base model transformation to target model.

Table 1
DenseNet building blocks [44].

Types	Definition	Contain	Remarks
Dense block	A dense block is a collection of layers that are all related to each other or to their previous layer.	3 × 3 Convolution ReLU (Rectified Linear Unit) activation Batch Normalization	**Growth rate:** The growth rate of a layer is defined as the number of output feature maps. When the model is very deep, concatenating residuals rather than summing them has a disadvantage: It produce a large number of input channels [45].
Transition layer	In this layer the concatenation of all the feature maps happens. To make sure each layer has the same size for concatenation this is the job of transition layer.	1 × 1 Convolution Average pooling Batch Normalization	**Vanishing Gradient:** The longer path between the input layer and the output layer is the cause when information disappears before it reaches its destination; consequently, the main goal of utilizing DenseNet is to reduce the issue of vanishing gradient and improve the accuracy loss caused by vanishing gradient in deep [46].

easier to implement the forward function and greatly decrease the number of network parameters. CNNs have issues when they delve deeper. This is due to the fact that the distance between the input layer and the output layer and in the opposite direction performing gradient grows to such a size that information may be lost before it reaches the other side (Pablo [48]). Instead of relying on highly deep or wide architectures for representational strength, DenseNet makes use of the network's potential by reusing features. Huang et al. demonstrated the two major problem that can be solved with DenseNet are the following:

1. Contrariwise, by connecting in this manner, DenseNet architectures require fewer parameters than an equivalent CNN because redundant feature maps are not required to be learned.
2. In addition, some variations of ResNets have demonstrated that many layers barely contribute and can be eliminated. ResNets have a large number of parameters because each layer must learn its own weights. In contrast, DenseNet layers are extremely narrow and simply add a small number of new feature-maps [36].

DenseNets also differ from other architectural designs, such as Highway Networks [49], Residual Networks [50], and Fractal Networks [51], by simplifying the connectivity pattern between layers. While these other architectures rely on deep or wide structures to enhance representational power, DenseNets focus on maximizing information and gradient flow by directly connecting every layer to each other. This approach allows for effective feature reuse within the network, harnessing the network's potential without the need for excessive depth or width.

4.2. Overview of general DenseNet architecture

DenseNet is a deep CNN architecture that was introduced by Huang et al. in 2017. It is intended to address the vanishing gradient issue in very deep neural networks by feeding forward from one layer to the next, thereby creating densely connected blocks. The DenseNet architecture is composed of multiple dense blocks, where each block consists of a stack of convolutional layers with the same number of filters. The output of each layer is passed on as input to all subsequent layers in the block, creating dense connections. The dense connections concatenate the feature maps from all previous layers, which are then fed to the next layer in the block. Each block's final layer is connected to a transition layer, which is made up of a layer for batch normalization, a layer for 1x1 convolution, and a layer for pooling. The transition layer reduces the number of feature maps and the spatial dimensionality of the output (see Table 1).

DenseNet transforms the conventional CNN design by offering dense connections and effective feature use. As a result of the direct connections made possible by these dense connections, gradient flow and information propagation have improved.

The dense block, which comprises of several convolutional layers stacked together (Pablo [48]), is the core component of DenseNet model. Let us use the symbols X for the dense block's input and H_l for its output, where l stands for the dense block's layer index. A dense block's computation can be seen as [36]:

$$H\_I = H\_\{I-1\} \otimes F\_1([H\_0, H\_1, \ldots ..H\_\{I\_1\}]) \quad (1)$$

Here, [H_0, H_1, . . . , H_l-1] symbolizes the concatenation of the feature maps from all preceding layers, while [f_l] represents the composite function made up of batch normalization, ReLU activation, and convolution operations. And this symbol ⊕ denotes the concatenation operation.

By introducing these thick connections, DenseNet encourages the network to learn more intriguing and varied representations while also increasing feature reuse. The skip connections enable gradients to flow directly from the output layer to the input layer.

The notion of transition layers, which are in charge of down sampling the feature maps and lowering the number of channels, is also introduced by the DenseNet design. A batch normalization layer, a 1x1

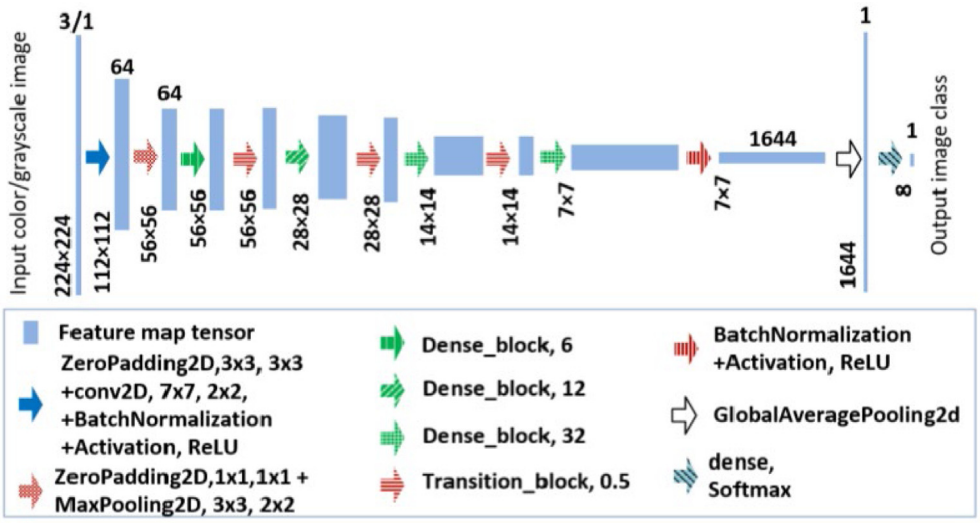

Fig. 3. Illustrates the DenseNet architecture, which is built upon the DenseNet169 model [52]. The input tensor size for color images is 224 × 224 × 3, while for grayscale images it is 224 × 224 × 1. The architecture incorporates a pattern of alternating Dense and Transition blocks to form its structure [53].

convolution layer, and an average pooling layer make up the transition layer. The transition layer's output, T_l, may be computed as follows:

$$T\_I = \theta\_I([H\_0, H\_1, \ldots ..H\_\{I\_1\}]) \quad (2)$$

In this case, θ_l stands for the composite function of the average pooling, batch normalization, ReLU activation, and 1x1 convolution procedures. DenseNet's general structure may be represented as a stack of dense blocks with transition layers put between them to regulate the channel size and spatial dimensions. To get the appropriate output predictions, the final global average pooling layer and fully connected layer are combined (see Fig. 3).

5. Related work

Systems that support healthcare decisions have drawn a lot of attention in clinical settings as useful tools for improving patient care overall, treatment planning, and diagnostic accuracy. To assess complicated patient data, such as medical imaging, laboratory findings, electronic health records, and clinical guidelines, these systems make use of cutting-edge technologies, such as machine learning and deep learning for diagnosis of Alzheimer's disease. An introduction to the various approaches in identification and classification of the disease is evaluated in light of the subsequent literature review. The classification of medical images has benefited from the development of a wide variety of efficient feature engineering approaches throughout the years. According to Suk et al. [54] which has used multi-model fusion with hierarchical features to detect AD in 2014 and it was tested on Positron Emission Tomography (PET) and MRI of 398 scans this includes 93, 104, 101 (AD MCI, and NC, respectively). Using the tissue densities from the MRI patch and the voxel intensities from the PET patch, they created the Multi Model patch-level feature learning architecture and VGG-16 and 19 was used as pre-trained model. Next, they trained the Restricted Boltzmann Machine as a preprocessor, which can convert real values observations into binary form. The multilevel classifier received the extracted features and classified them accordingly such as MCI - CN and AD - CN with accuracy of 85.67%, 95.35%, respectively.

In their study, the authors [55] suggest three successful approaches for generating visual representations to classify AD using 3D convolutional neural networks (3DCNN). Specifically, they employ the 3D-ResNet and 3D-VGGNet techniques to classify MRI scans from both AD and NC, utilizing ADNI brain MRI images. The performance of the classification models is assessed using metrics such as the area under the AUC and classification accuracy (ACC). Among the methods, the 3D-ResNet achieves the highest accuracy of 79.4%.

The authors in this study [56] propose a method for diagnosing AD by employing a CNN along with a combination of sMRI and Diffusion Tensor Imaging (DTI) techniques. They focus on analyzing a specific region of interest (ROI) in the hippocampus using ADNI data. The study includes 214 individuals, consisting of 48 AD cases, 108 cases with MCI, and 58 NC patients. Each patient undergoes both a T1-weighted sMRI and a DTI scan. The authors also address the challenge of imbalanced class sizes and investigate the impact of ROI size on classification outcomes. The proposed approach achieves an impressive classification accuracy of 96.7%.

Yosra Kazemi [57] and his colleagues used AlexNet architecture for classification of different stages of the AD using fMRI scans. The model's average accuracy was 97.63%. Using DL algorithms, they have successfully classified five different stages such as AD, NC EMCI (Early-Mild-Cognitive-Impairment), SMC (Significant-Memory-Concern), andLMCI (Late-Cognitive-Mild-Impairment). The tested accuracy for AD, NC, EMCI, SMC and LMCI were 94.97%, 98.34%, 95.89%, 94.55%, 98.34% respectively.

The author [53] in this experiment utilized Computed Tomography (CT) brain images. The study employed three pre-trained models, namely ShuffleNet, DenseNet, and NASNet-mobile, in combination with CNN. Among the three models, DenseNet demonstrated the highest performance, achieving an accuracy of 87.36% on three stages of AD such as MID, Moderate Demented (MOD), and VMD.

In this research [1], various pre-trained models including ResNet18, AlexNet, SqueezeNet, VGG16, InceptionV3 and DenseNet are employed to classify AD into four categories. The experiments are conducted using MRI images from the OASIS database. The evaluation of the implemented models reveals that the pre-trained SqueezeNet model achieves the highest validation accuracy of 82.53% for the multiclass classification of AD. While DenseNet 121 achieved training accuracy and validation accuracy of 67.31% and 67.72%, respectively.

The goal of this study [58] was to improve AD images classification using DCNN (Deep Convolutional Neural Networks) also known as deep ConvNets, which incorporate VGG16 and 19 transfer learning and CNN. The classification AD images are done into four classes such as MID, MOD, VMD, and ND. The results accuracy for CNN, VGG-16 and 19 are 0.710%, 0.770% and 0.776% respectively. Part of the problem with this research was that the experiment could not use real-world or local datasets. Also, the system used to implement the techniques showed that it took a long time to do computations. Both of these things could affect the results of the work.

This study [59] introduces a method using CNN with the TL-based ResNet-18. Transfer learning from ImageNet is used to overcome the

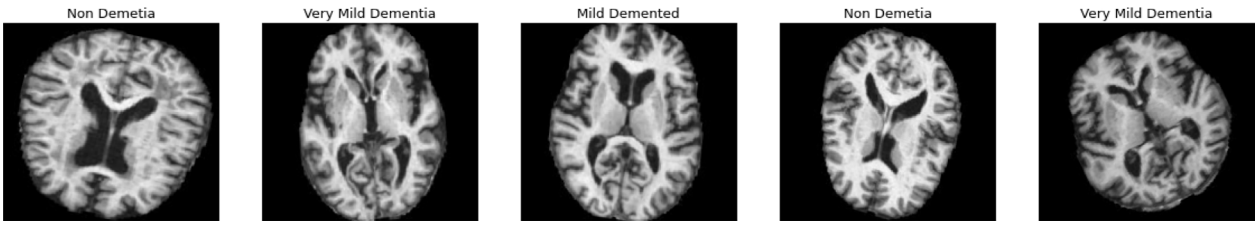

Fig. 4. Random sample images of three different categories.

drawbacks of having a big and balanced dataset, and the loss function is weighted to ensure equal importance for each class. Additionally, an experiment is conducted by replacing the network activation function with a mish activation function to improve accuracy. The test results reveal that the model achieves an accuracy of 88.3% when utilizing transfer learning, weighted loss, and the mish activation function. This accuracy value is significantly higher compared to the baseline model, which only achieves an accuracy of 69.1%.

Hamed Ghaffari et al. [60], take advantage of CNN-based transfer learning methods for the classification of AD using MRI. The study utilizes three pre-trained models such as ResNet101, Xception, and InceptionV. For binary (AD vs. non-AD) and 4-class (AD/pMCI/sMCI/NC) classification, the proposed networks are trained and tested using pre-processed and augmented segmented and full pictures. External test sets from the OASIS and AIBL datasets are used for performance evaluation. The results demonstrate that transfer learning-based CNN models outperform models trained from scratch. InceptionV3-TL achieves the highest accuracy and AUC in both binary and multiclass classification tasks on the ADNI test set, OASIS test set, and AIBL test set. For binary classification on the OASIS test set they have achieved an accuracy of 93.33% with 93.0% AUC.

In this paper [61] the authors focus T1-weighted MRI scans to do classification based on deep learning approach. The researchers utilized transfer learning with an EfficientNet model trained on ImageNet to classify subjects as AD vs CN. The study used data from the ADNI and achieved an accuracy of 91.36% and AUC of 83%.

According Hridhee et al. [62] to The VGG16 and Xception models, along with a custom CNN model, were evaluated with 2D MR images. The custom model achieved the highest performance with an accuracy of 0.9477 and F1-score of 0.9481. It outperformed traditional methods, offering reduced complexity, shorter processing time, and improved efficiency compared to 3D MRI-based CNN techniques. The author [63] author evaluate the performance of the MobileNet approach using two publicly available datasets: OASIS and ADNI. Results indicate that the approach outperforms conventional methods in terms of accuracy and sensitivity across various evaluation cases. Particularly, this implementation of MobileNet achieves high accuracy in the OASIS, ADNI, and merged test sets (OASIS + ADNI) of 95.24%, 81.94%, and 83.97%, respectively, surpassing the performance of conventional approaches.

According Rabeh et al. [64] the authors combined approach of CNN and SVM for early-stage prediction of AD, specifically MCI. The proposed method is tested on a dataset from the OASIS database, which comprises 420 subjects, including 210 normal individuals and 210 with MCI. The results demonstrate an accuracy of 94.44% for AD prediction using this approach. This study [65] proposes a multimodal fusion approach that Discrete Wavelet Transform (DWT) analysis with VGG16. The fused images are reconstructed using inverse DWT and then classified using a pre-trained vision transformer. Accuracies of 81.25% for AD/EMCI and AD/LMCI in MRI test data, and 93.75% for AD/EMCI and AD/LMCI in PET test data. The proposed model outperforms existing studies, achieving an accuracy of 93.75% for PET data. Table 2. Present a summary of all recent studies with their performance.

6. Materials and methods

While numerous studies in recent years have focused on datasets such as ADNI and OASIS, which use several imaging modalities incorporating many features of pictures from multiple modalities, such as sMRI, CT, fMRI, and MRI scans, our study focuses on pre-processed MRI images from a Kaggle source. We depart from current patterns and investigate the uncharted territory of DenseNet architecture, with the goal of diversifying its configurations for enhanced performance. We simplify preprocessing to meet the model's criteria while avoiding the unnecessary complexity seen in other approaches. We choose a three-class categorization over existing approaches that handle several output classes like 2, 3 or 4.

Furthermore, we emphasize the importance of on-the-fly data augmentation, a strategy that has received little attention in recent research. Furthermore, we address time complexity, stressing the applicability of our model in resource-constrained contexts. Finally, the relevance of our work rests in its simplicity, which was attained by careful architecture selection and efficient preprocessing. DenseNet201 performs well, demonstrating the potential significance of models in classification of AD. The subsection we have talks in detail about the data description and the model description.

6.1. Data description

Numerous datasets are available online for the classification of Alzheimer's disease. Many of these datasets, though, are in the form of Comma-separated Values which do not meet the criteria for our study. Notably, organizations like Open Access Series of Imaging Studies (OASIS) and Alzheimer's Disease Neuroimaging Initiative (ADNI) make their datasets accessible for the public since the research is concern. Despite this, these datasets are incredibly big and mainly comprise 3D image types. For instance, the size of dataset from ADNI is staggeringly large at 450 GB, while the OASIS dataset is around 18 GB. For this study. Our dataset was sourced from an online Kaggle challenge, specifically focusing on MRI brain images. The dataset provided for this challenge includes a total number 6400 including training and testing and categorized in to four classes as ND, MID, MOD and VMD and contain 200 subjects with 32 slices of the image for each subject the samples from anonymous patients. The data source can be found in Kaggle (Alzheimers-dataset-4-class-of-images). We have considered three stages of AD for this study such as ND, MID and VMD since the MOD contain very less amount of image slices, we did not consider that class. As seen in Fig. 4, each image represents a 2 Dimentional MRI cross-section of the brain, capturing a specific plane at varying heights within the brain. These images are grayscale with 176 × 208 pixels dimension. Additionally, all the MRI brain is cross-section and centered and the background of the images are clipped. The dataset underwent various preprocessing steps to ensure the data was appropriately prepared for subsequent stages.

Data preprocessing is a method of putting together input data and turning it into something DL models can use. It is the cornerstone of the suggested research methodology [66]. To remove the noise, find the missing values, and fix them, as well as to prepare the data for proper use, data preprocessing is necessary [67]. This also improves the effectiveness of the entire research model that has been proposed. In this

Table 2
Present each recent paper's technique, modality use and the achieved results.

Author & Date	Technique	Data modality	No-of-Classes	Obtained accuracies
Suk et al. [54]	VGG-16 & 19	PET and MRI scans	3 Classes: MCI - CN and AD - CN	Accuracies: 85.67% and 95.35%
Yang et al. [55]	3D CNN 3D-VGGNet & 3D-ResNet	ADNI	2 AD vs CN	3D-ResNet accuracy of 79.4%.
	CNN	ADNI sMRI, DTI	2 AD VS NC	Accuracy of 96.7%
Kazemi and Houghten [57]	AlexNet	fMRI	5 Classes: AD, NC, EMCI, SMC and LMCI	Average accuracy 97.63% Accuracy of all classes 94.97%.
Yudin et al. [53]	ShuffleNet, DenseNet and NASNet-mobile	CT	3 MID, MOD and VMD	Best accuracy of DenseNet: 87.36%
Odusami et al. [1]	ResNet18, AlexNet, SqueezeNet, VGG16, InceptionV3 & DenseNet	OASIS	4 NC, MCI, AD	SqueezeNet 82.53% DenseNet 67.31% and 67.72%,
Ajagbe et al. [58]	DCNN-VGG-16 & VGG-19	MRI	4 Classes: MOD, VMD and ND	Accuracy of all Approaches: 0.710%, 0.770%, 0.776%
Oktavian et al. [59]	CNN with the ResNet-18	ADNI	3 AD, CN, MCI	69.1%
Ghaffari et al. [60]	ResNet101, Xception, and InceptionV	ADNI, OASIS	2 AD vs non-AD 4 AD/pMCI/sMCI/NC	accuracy of 93.33% with 93.0% AUC. Binary
Sethi et al. [61]	EfficientNet Model	ADNI	2 AD vs CN	Accuracy: 91.36% with AUC of 83%
Hridhee et al. [62]	VGG16 and Xception and custom CNN	2D MR images	Full paper not available	Best accuracy custom model: 0.9477
Ghosh et al. [63]	MobileNet	ADNI, OASIS	4 OASIS: MID, MOD, VMD and ND 2 ADNI: ND and VMD	Accuracy: 95.24% ADNI, OASIS 81.94%, Merged 83.97%
Rabeh et al. [64]	CNN and SVM	OASIS	2 MCI vs CN	Accuracy: 94.44%
Odusami et al. [65]	Pixel-Level Fusion Approach DWT with VGG16	ADNI: sMRI and FDG-PET	3 AD vs EMCI AD vs LMCI	Accuracy: 93.75%

model, data normalization and data augmentation are the two types of data pre-processing that are used [68]. The modified DenseNet model's numerical stability is preserved by data normalization. One crucial preprocessing step involved normalizing the image values. Initially, the images were represented in grayscale, with pixel values ranging from 0 to 255, and then rescaled the pixel values within a normalized range of 0 to 1. The images are resized to 224×224 due to the requirement of the model.

A large dataset is required to increase the model's usefulness. However, obtaining these datasets presents challenges due to numerous sites, data limitations and privacy concerns. The Keras ImageDataGenerator is used for data augmentation purposes. This module augments data to increase model generalization. In data augmentation, a random image data generator does translations, rotations, scale changes, and vertical flips. In real-time data augmentation, Keras ImageDataGenerator generates batches of tensor picture data (Tiara [69]). Using the correct parameters and input, we can use the ImageDataGenerator resize class. Batch size and the number of inputs impact the number of images created. We have used on-the-fly data augmentation, where modification methods such as rescale, rotation range, zoom range, horizontal and vertical flip were used in order to generate new images from the original image. Training and test sets are originally created and split from the dataset. Table 3 demonstrates the distribution of dataset across three classes and splitting into training and testing set.

Fig. 4 displays a random selection of images from three different categories, namely ND, MOD, and VMD. Each category is represented by four sample images. The images show different brain scans that are used in the analysis of dementia. These images are an essential component of the research, as they help in performance testing and training the model in differentiating between healthy individuals and those with varying degrees of dementia.

Table 3
Dataset distribution across three different classes.

Class	Training	Testing
Non-demented	2560	640
Mild-demented	717	179
Very-mild-demented	1792	448
Total	5069	1267

6.2. Model description

The experimentation was conducted using Google Colab, a cloud-based platform that provides a Python development environment. Google Colab offers the advantage of readily available computing resources, including GPUs, which are essential for training deep learning models. This allowed us to leverage the power of accelerated hardware for faster model training and evaluation. Since we are faced with limitation in computational speed therefore, we try to attempt the small changes in hyperparameter tunning. Rather than using advanced techniques which require more computing power and consume more time to train. We utilized the TensorFlow framework for our experimentation. By leveraging TensorFlow's high-level APIs and efficient computational backend, we were able to build and train our models effectively. TensorFlow's extensive ecosystem also allowed us to access

pre-trained models, perform data preprocessing, and evaluate model performance. Keras library, which is now integrated with TensorFlow has been used for our data augmentation and preprocessing in the training, validation, and testing stages. Several data augmentation techniques to the training images as mentioned earlier in dataset description. Rest of the process and steps are as follows:

- Resizing the images to a consistent size is necessary to ensure compatibility with the DenseNet model's input layer, which expects images of a specific size. Therefore, In this case, the images are being resized to a target size of 224 × 224 pixels.
- Batch size parameter is used to determine the number of samples in each batch of data during training, in this case, 128.
- The DenseNet201 model was used as the base model with the dimensions of the input images. By setting attribute of the architecture include_top to False, the fully connected layers at the top of the DenseNet201 model, which were responsible for the final classification, were excluded. The general structure will remain the same, with freezing the layers, for extracting the general features of the data while another stack of hidden layers is created to perform the domain specific feature extraction. Therefore, the feature extractor strategy of TL has been used in this study.
- The weights parameter was set to "ImageNet", indicating that we used the pre-trained weights from training on the ImageNet dataset. This initialization helped the model leverage the knowledge learned from millions of images in a wide range of categories.

  Customizing the model's last few layers are as follows:

  - **A sequential model** is first created. The base model is added as the first layer.
  - **Dropout** layers are added after each fully connected layer with a dropout rate of 0.5 to reduce overfitting.
  - **Flatten layer** is used to flatten the output from the previous layer into a 1D vector.
  - **Batch Normalization** layers are added after each fully connected (Dense) layer. Batch normalization helps stabilize the training process and improves model performance.
  - **Activation('relu')** is applied after each batch normalization layer to introduce non-linearity.
  - **SoftMax**, the last Dense layer with 3 units and activation is added for multi-class classification, indicating the outputs classes.

Different data preprocessing methods were used to improve model training. To diversify the training data, image augmentation techniques such as rotation up to 30 degrees, zooming within a range of 0.8x to 1.2x, and horizontal and vertical flipping were used. To assess model performance, the dataset was further divided into training and validation sets with validation splits of 0.2, 0.4, and 0.6.

In order to avoid overfitting, a batch size of 128 was used during training, and a dropout rate of 0.5 was implemented at multiple layers. The difference between the predicted and actual labels was calculated using the categorical cross entropy loss function. The Adam optimizer was selected to reduce this loss, and a learning rate of 0.003 was set for efficient convergence. We carefully experimented with various learning rates during the training process to find the ideal balance.

A learning rate that is too high or too low can compromise training. After experimenting with different values, we chose 0.003 because it allowed for a quick convergence without compromising stability. This decision was essential for effective training, ensuring the model could efficiently learn from the data and generalize. Early stopping was used with a 15-epoch patience period, monitored by the validation AUC score, to ensure the model's robustness and prevent overfitting. Accordingly, model checkpoints were saved. The model was able to effectively learn and optimize its parameters because the training process was over 170 epochs.

Additionally, the model was evaluated using the METRICS defined previously, including accuracy, precision, recall, AUC, and F1 score. The complete proposed workflow diagram is given in the following Fig. 5. The architecture selected for classifying the stages of AD is DenseNet201. It has a total of 214,146,627 parameters, out of which 195,628,803 parameters are trainable. The remaining 18,517,824 parameters are non-trainable. Table 4 presents the summary of significant parameters used in this experiment.

Using a variety of techniques, we thoroughly evaluated and compared the performance of our model. We rigorously tested four different approaches such as (1) training each model with the original unaltered dataset, (2) applied data augmentation to enrich the training dataset, (3) a modified augmentation technique without specific transformations and (4) varying the training and test data split ratios. These methods were painstakingly created to provide a profound understanding of the impact of data augmentation, the requirement for particular transformations, and the impact of dataset size on model performance. Our goal was to compare these approaches methodically in order to determine which one would be the most useful for our classification task. We also contrasted how long each model took using various methods (see Fig. 6).

7. Experiment results

A variety of metrics are used to assess a classification model's performance, such as accuracy which is one of the simplest classification metrics that is calculated as the proportion of accurate predictions to all other predictions. Precision is determined by the proportion of positive predictions that were accurate. To calculate the True Positive (TP) predictions which are exactly true divide by all positive predictions such as True Positive and False Positive (FP). Another metric is Recall; the main goal is to determine the percentage of actual positives predictions that were incorrectly predicted. We can calculate this as taking all the TP that are actually true dividing it with total number of TP and False Negative (FN) either positive correctly predicted or negative incorrectly predicted. f1_score used to tell your FP and FN rates are low. By giving each variable equal weight, The harmonic mean of both precision and accuracy and recall parameters can be used to compute it. AUC, which displays the model's capability in distinguishing between classes which is used as a summary of the ROC curve [70]. Lastly, ROC curve(Receiver Operating Curve) is a graph that displays how well the model performs at various threshold values. Below are the mathematical equations for calculation of the metrics (Frankie [71]).

$$Accuracy = \frac{\text{TruePositives} + \text{TrueNegatives}}{\text{TruePositive} + \text{TrueNegatives} + \text{FalsePositive} + \text{FalseNegative}} \tag{3}$$

$$Precision = \frac{\text{TruePositives}}{\text{TruePositive} + \text{FalsePositive}} \tag{4}$$

$$Recall = \frac{\text{TruePositives}}{\text{TruePositive} + \text{FalseNegative}} \tag{5}$$

$$f1 - Score = 2^{*}\frac{\text{Precision } ^{*}\text{Recall}}{\text{Precisionp} + \text{Recall}} \tag{6}$$

We present a thorough assessment of our model using various approaches in our results and analysis. As a first approach, we establish a performance baseline without data augmentation. We then examine the improved performance brought about by data augmentation in the second approach, demonstrating its efficiency in allowing the model to generalize from sparse data. The effects of skipping particular transformations also done in the third approach, Additionally, by adjusting the training and test split ratios, we examine the impact of dataset size in fourth approach. The accompanying figures and tables give our findings a clear visual representation while succinctly presenting the insights gained from these methods. These visual aids contribute to making well-informed decisions about model design and improve understanding of our model's performance under various conditions.

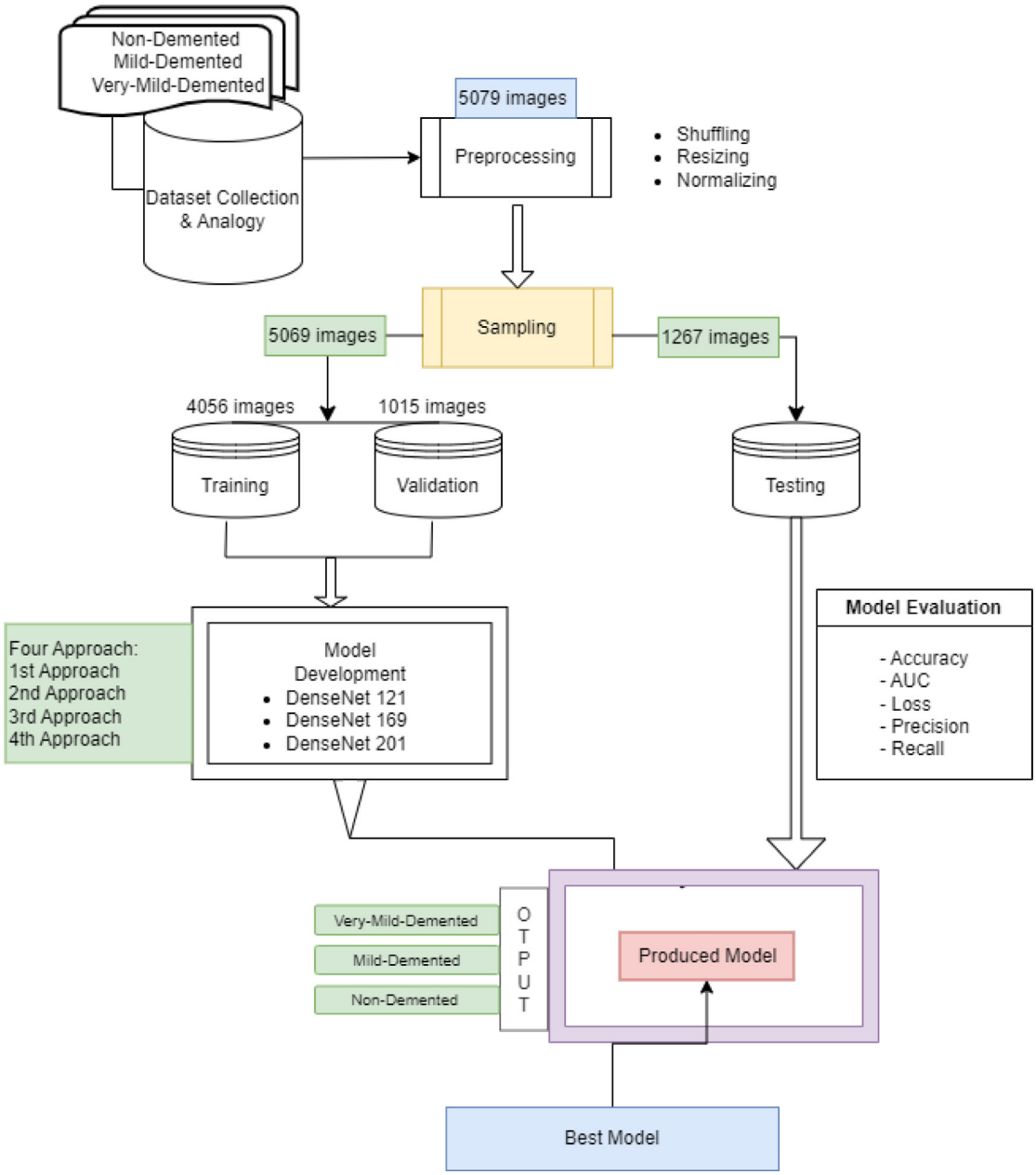

Fig. 5. Proposed workflow diagram.

Table 4
Hyperparameters.

Parameters	Values	Parameters	Values
Weights	ImageNet	Dropout Rate	0.5 (used multiple times)
Rotation Range	30	Dense Layer Units	2048, 1024, 512, 256, 3
Zoom Range	0.2	DenseNet Architecture	DenseNet201
Horizontal Flip	True	Loss Function	Categorical Cross entropy
Vertical Flip	True	Optimizer	Adam
Validation Split	0.2, 0.4, 0.6	Learning Rate	0.003
Target Image Size	(224, 224)	Early Stopping Patience	15
Batch Size (Training and Validation)	128	Callback Checkpoint Monitor	'val_auc'
Activation Function	ReLU and SoftMax	Epochs	170

1. **Baseline Performance without Data Augmentation:**

This strategy acts as a benchmark for contrast. In this instance, no data augmentation was used when training the models. This method was used to assess the model's performance when it only uses the original, undisturbed dataset. The training and validation metrics for DenseNet models 201, 169, and 121, which include loss, AUC, and accuracy, are shown in Figs. 7, 8, and 9 respectively. Additionally, Tables 5 and 6 offer a thorough summary of the training and testing results for each model, assisting in a thorough evaluation of the models.

2. **Enhanced Performance with Data Augmentation:**

In this method, the dataset was enhanced with data before the models were trained. By applying transformations like rotations, zoom

```
1. Data Preparation:
   a  Define image data generators for training, validation, and test datasets.
   b. Set data preprocessing parameters.
   c. Load and preprocess the training dataset using the training data generator.
   d. Load and preprocess the validation dataset using the validation data generator.
   e. Load and preprocess the test dataset using the test data generator.

2. Model Initialization:
   a. Initialize the DenseNet201 base model.
   b. Freeze the layers of the base model.

3. Building Model:
   a. Initialize a sequential model.
   b. Add the base model as the first layer.
   c. Add a dropout layer.
   d. Add a flatten layer.
   e. Add batch normalization layer(s).
   f. Add dense layer(s) with specified initialization, activation, and dropout.
   g. Add the output layer with SoftMax activation.

4. Compile Model:
   a. Define the optimizer, loss function, and metrics.
   b. Compile the model with the specified settings.

5. Defining Callbacks:
   a. Define early stopping callback with specified monitor, mode, and patience.
   b. Define checkpoint callback to save the best model based on a specified monitor.

6. Model Training:
   a. Train the model on the preprocessed training and validation datasets.
   b. Use the defined callbacks during training.

7. Model Evaluation:
   a. Evaluate the trained model on the preprocessed test dataset.

8. Prediction:
   a. Load and preprocess a sample image.
   b. Make predictions using the trained model on the sample image.
   c. Calculate the probability of the predicted class.
   d. Display the predicted class and probability.
```

Fig. 6. Pseudocode of the proposed approach.

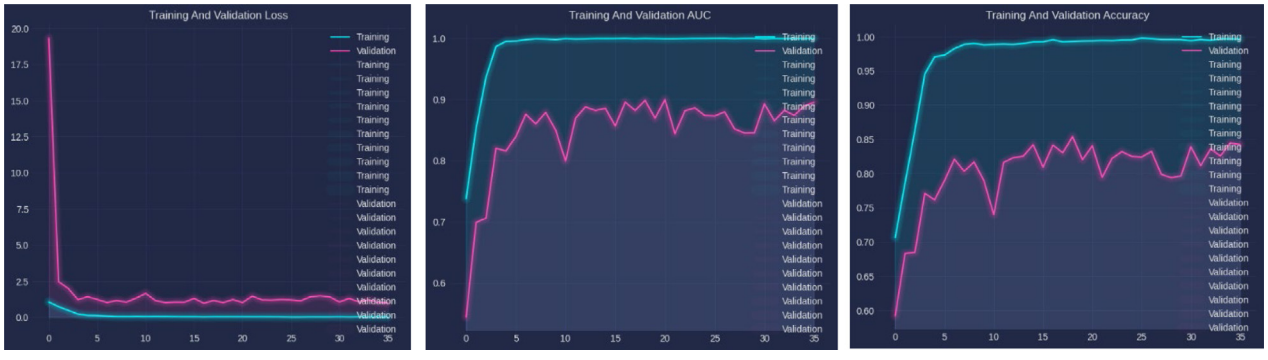

Fig. 7. Training and validation AUC and Loss and Accuracy of DenseNet201 model.

Table 5
Summary of training performance.

Model	Optimizer	Loss	Accuracy	Precision	Recall	AUC	F1 score	Training process
DenseNet121	Adam	0.0162	0.9961	0.9941	0.9941	0.9997	0.9941	39: early stopping
DenseNet169	Adam	0.0189	0.9962	0.9943	0.9943	0.9999	0.9942	41: early stopping
DenseNet201	Adam	0.0189	0.9962	0.9943	0.9943	0.9999	0.9227	35: early stopping

Fig. 8. Training and validation AUC and Loss and Accuracy of DenseNet169.

Fig. 9. Training and validation AUC and Loss and Accuracy of DenseNet121.

Fig. 10. Training and validation AUC and Loss of DenseNet121 model.

Table 6
Summary of testing performance.

Model	Loss	Accuracy	Precision	Recall	AUC	f1_score
DenseNet121	1.8494	0.7914	0.6876	0.6860	0.8274	0.6869
DenseNet169	1.7561	0.8207	0.7316	0.7298	0.8458	0.7307
DenseNet201	1.4991	0.8300	0.7456	0.7435	0.8700	0.7446

range, horizontal and vertical flips to the original images, data augmentation is a common technique for artificially expanding the training dataset. This was carried out to improve the model's performance and increase its capacity to generalize from sparse data. This method allowed us to evaluate how well the models functioned after being trained on augmented data. The training and validation metrics, including loss, AUC, and accuracy, are shown in Figs. 10, 11, and 12 for DenseNet models 121, 169, and 201, respectively. Tables 8 and 9 similarly give an overview of each model's training and testing performance. Due to the unpredictability inherent in various augmentation techniques (such as zoom, rotation, and so on), calculating the actual data size after applying data augmentation can be difficult. However, depending on the following augmentation parameters we have set, we can approximate the size by leveraging the capabilities of the Keras ImageDataGenerator. Therefore, we have taken an example as 30 degrees of rotation range, Up to 20% of zoom range, horizontal and vertical flip enabled. We first counted the number of batches in the training set and validation set to assess the growth in data size. We computed class-wise sample counts for each set of data using a custom function called count_samples. To estimate the total samples for both sets of data, we then consider our best model DenseNet201 which ended up with 120 early stopping therefore, batch size and step count per epoch settings are 128 and 32, respectively. This implies that you will generate and use 32 batches of data, each of which has 128 samples, for each epoch. When paired with these criteria, the count_samples function

Fig. 11. Training and validation AUC and Loss of DenseNet169 model.

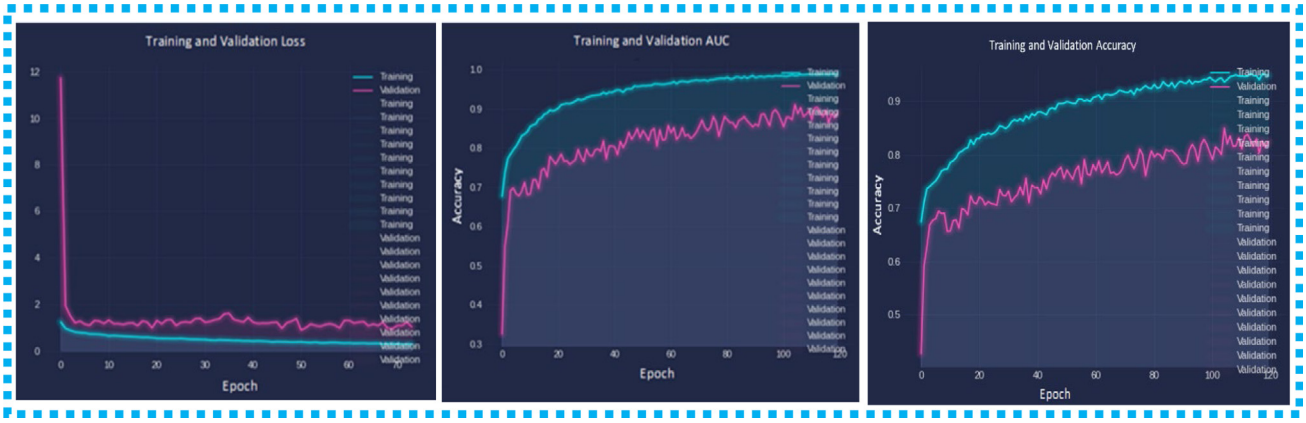

Fig. 12. Training and validation AUC and Loss of DenseNet201 model.

Fig. 13. Training and validation AUCs and Loss and Accuracy of DenseNet121.

Table 7
Training and validation dataset size before and after augmentation.

Before and after	Training	Validation
Original Dataset- Total of training and validation set	4,054	1,015
Augmented - approximate total of training and validation set	393,216	98,304

offered an approximation of the dataset size after augmentation as the following shown in Table 7. So total augmented samples for training and validation for the entire training process are calculated as follows total number of samples in a single epoch and total number of epochs ($4054 \times 120 = 491520$).

3. Selective Augmentation Analysis

A data augmentation technique was used in this case, but it did not include changes to attributes' horizontal and vertical dimensions. This particular decision was made in order to examine how these particular transformations would affect the model's performance. We attempted to determine whether particular data augmentations contribute more or less to the model's capacity to make accurate predictions by excluding these transformations. The training and validation metrics for the corresponding DenseNet models under this particular augmentation strategy are shown in Figs. 13, 14 and 15. Tables 10 and 11 also provide

Fig. 14. Training and validation AUCs and Loss and Accuracy of DenseNet169.

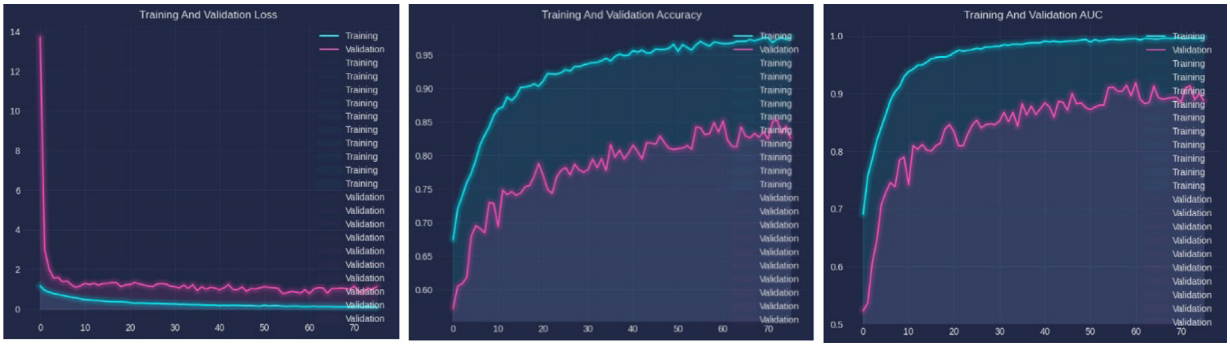

Fig. 15. Training and validation AUCs and Loss and Accuracy of DenseNet201.

Fig. 16. Training and validation AUC and Loss and Accuracy of DenseNet201 (40%).

Table 8
Summary of training performance.

Model	Parameters	Loss	Accuracy	Precision	Recall	AUC	f1_score	Training process
DenseNet121	Adam	0.3607	0.9248	0.8687	0.8415	0.9637	0.8547	135: early stopping
DenseNet169	Adam	0.2671	0.9300	0.8999	0.8890	0.9797	0.8948	145: early stopping
DenseNet201	**Adam**	**0.2013**	**0.9605**	**0.9264**	**0.9193**	**0.9901**	**0.9227**	**120: early stopping**

Table 9
Summary of testing performance.

Model	Loss	Accuracy	Precision	Recall	AUC	f1_score
DenseNet121	1.5907	0.8248	0.7687	0.6915	0.7637	0.7547
DenseNet169	1.5671	0.8300	0.7999	0.7790	0.7797	0.8948
DenseNet201	1.3304	0.9001	0.8014	0.7069	0.8607	0.8022

a thorough breakdown of the training and testing results, providing important insights into model behavior.

4. Impact of the Training/Test Split Ratio:

We applied data augmentation and the training and test data split percentages (40%, 60%, and 80%) were changed in this method. We aimed to explore the impact of the size of the training dataset on the model performance by varying the split ratios. Overfitting can occur with smaller training datasets, while generalization may be enhanced with larger ones. We can choose the ideal ratio for our particular dataset thanks to this variation. The training and validation metrics for DenseNet 201 are shown in Figs. 16 and 17, which shed light on the influence of split ratios (see Tables 12 and 13).

Fig. 17. Training and validation AUC and Loss and Accuracy of DenseNet201 (60%).

Table 10
Summary of training performance.

Model	Parameters	Loss	Accuracy	Precision	Recall	AUC	f1_score	Training process
DenseNet121	Adam	0.2185	0.9450	0.9215	0.9129	0.9859	0.9171	48: early stopping
DenseNet169	Adam	0.0992	0.9772	0.9669	0.9646	0.9966	0.9657	90: early stopping
DenseNet201	Adam	0.0985	0.9762	0.9664	0.9621	0.9971	0.9643	76: early stopping

Table 11
Summary of testing performance.

Model	Loss	Accuracy	Precision	Recall	AUC	f1_score
DenseNet121	1.7676	0.7643	0.6476	0.6425	0.8135	0.6442
DenseNet169	1.1743	0.8275	0.7425	0.7384	0.8770	0.7403
DenseNet201	2.0271	0.8011	0.7023	0.7001	0.8275	0.7012

The training and testing outcomes for different models under various approaches are comprehensively summarized in Table 14 that is being presented. It is clear that data augmentation significantly improves accuracy and AUC scores across the models, having a major impact on model performance. Notably, DenseNet201 demonstrated strong performance in the majority of scenarios, highlighting its potential for accurate forecasting. Initial observations might show that alternative approaches have higher training accuracy and lower loss, but a more thorough examination of validation metrics using related graphs and testing tables reveals a more interesting picture. When testing the model with unseen set of data, DenseNet201 in the "Enhanced Performance with Data Augmentation" approach consistently displayed superior performance. This demonstrated the model's potent generalization capabilities. Additionally, the experiments with different training/test split ratios and the exclusion of particular augmentations shed light on the nuanced influence of these factors during model training. These revelations emphasize the crucial role that careful model configuration and thoughtful data preprocessing play in achieving the best performance for realistic deployment.

Fig. 18 depicts the variation in accuracy of the top model, DenseNet201, in relation to the quantity of training epochs. The graph demonstrates that the model's accuracy rises as the number of epochs increases until it hits a high at about 70 epochs, at which point it stabilizes. This suggests that the model has learned the patterns and features of the data well enough, and that further training does not result in substantial improvement in accuracy. The graph also shows that the accuracy of the model on the validation set is consistently higher than that on the training set, which indicates that the model is not overfitting. Overall, this graph provides a visual representation of the learning curve of the DenseNet201 model and helps in determining the optimal number of epochs for training the model. After training and evaluation of each model we also choose the best model which is DenseNet201 to test calculate the percentage of predicted values for each image category such as ND, MOD and VMD the output of each predicted classes is shown in the Fig. 19.

We compared the recent studies in the field to our proposed approach for the classification of. Table 15 provides a summary of the comparison's outcomes. The table provides a comprehensive overview of the key findings from the selected studies and highlights how our proposed approach differs in terms of methodology, dataset, and performance metrics. By examining this table, it becomes evident that our approach achieves competitive results in terms of accuracy and classification performance. Our proposed approach outperforms several previous studies in terms of accuracy, achieving a remarkable 96.05% accuracy rate with DenseNet201. Moreover, it demonstrates improved precision and recall values, indicating a higher level of accuracy in correctly classifying ND, MID, VMD cases. In our comparison with various recent studies, it is noteworthy that our approach also places a strong emphasis on an important metric that is frequently overlooked in recent works the AUC. AUC is a crucial metric, particularly in applications involving medical imaging, as it sheds light on how well the model can distinguish between classes. The lack of reported AUC values in several recent studies can be a drawback when thoroughly assessing the model's performance. A high AUC indicates a strong capacity to distinguish between classes, boosting trust in the model's predictions. With an impressive AUC of 99%, our DenseNet201 model demonstrates its exceptional discriminative power and diagnostic precision.

Time taken for training the model:

The time complexity of an algorithm refers to how long it takes to train and predict. Parameters, types of the model, hardware, and dataset size, all have an impact. Several significant factors and strategies were used in the training of our model. To begin, as previously indicated, on-the-fly data augmentation was used, greatly increasing the sample count and its diversity. Considering our best performed model DenseNet201 which has 214,146,627 parameters. With a dropout rate

Fig. 18. The performance of training accuracy of the best model.

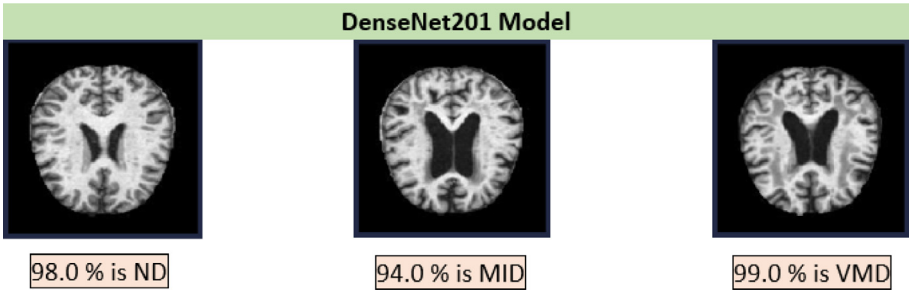

Fig. 19. Predicted output as percentage on three different classes.

Table 12
Summary of training performance.

Model	Ratio	Loss	Accuracy	Precision	Recall	AUC	f1_score	Training process
DenseNet201	0.6	0.3083	0.9117	0.8754	0.8573	0.9726	0.8665	38: early stopping
DenseNet201	0.4	0.2495	0.9330	0.9038	0.8941	0.9819	0.8990	81: early stopping
DenseNet201	0.2	0.2013	0.9605	0.9264	0.9193	0.9901	0.9227	120: early stopping

Table 13
Summary of testing performance.

Model	Ratio	Loss	Accuracy	Precision	Recall	AUC	f1_score
DenseNet201	0.6	4.2957	0.6817	0.5225	0.5225	0.6647	0.5222
DenseNet201	0.4	3.2227	0.7989	0.6555	0.6533	0.7747	0.6544
DenseNet201	0.2	1.3304	0.9001	0.8014	0.7069	0.8607	0.8022

of 0.5, 128 batch size and 0.003 learning rate, the training of the model consisted of 170 epochs, with 32 steps per epoch. The training included early stops and ended around the 120th epoch as shown in Fig. 20. The training time of our model DenseNet201 took around 1.97 h. Some of the elements as per our observation that may have contributed to the 1.97 h training duration for DenseNet201 include:

- Dataset size: the dataset is relatively large, with 5069 and 1267 for training and testing set respectively. And after applying the data augmentation which will significantly increase the amount of the existing data. This implies that the model has a large amount of data to learn from, which may lengthen the training period.
- Batch size: another key aspect which contribute to this complexity might be due to the value of batch which in our case it is 128. This means that the model is trained on 128 photos simultaneously. The advantage could be greater batch size can improve training efficiency, but drawback can lead to lengthen training duration.
- Model's complexity: DenseNet architectures are sophisticated in terms of their structure with many parameters. This can also lengthen the training period.
- Learning rate: a learning rate of 0.003. which is relatively low learning rate, which can benefit us to keep the model from overfitting, but it can also lead to lengthen training time.

Table 14
Overall summary of training, testing of various models with respect to their accuracy, AUC and time.

Approach	Model	Training		Time	Testing	
		Accuracy	AUC	Hour	Accuracy	AUC
Without Data Augmentation	DenseNet121	0.9961	0.9997	0.28	0.7914	0.8274
	DenseNet169	0.9962	0.9999	0.31	0.8207	0.8458
	DenseNet201	0.9962	0.9999	0.83	0.8300	0.8700
With Data Augmentation	DenseNet121	0.9248	0.9637	2.19	0.8248	0.7637
	DenseNet169	0.9300	0.9797	2.39	0.8300	0.7797
	DenseNet201	0.9605	0.9901	1.97	0.9001	0.8607
With Data Augmentation Excluding Horizontal and Vertical Attributes	DenseNet121	0.9450	0.9859	1.35	0.7643	0.8135
	DenseNet169	0.9772	0.9966	2.07	0.8275	0.8770
	DenseNet201	0.9762	0.9971	2.02	0.8011	0.8275
With Data Augmentation, Training Ratio (0.6, 0.4, 0.2) DenseNet201	DenseNet201	0.9117	0.9726	0.49	0.6817	0.6647
	DenseNet201	0.9330	0.9819	1.44	0.7989	0.7747
	DenseNet201	0.9605	0.9901	1.97	0.9001	0.8607

Table 15
Comparison of the recent studies with proposed approach.

TL-Approaches	Images	Classifier	Accuracy	AUC
VGG-19 no	PET+MRI (ADNI)	Multi-class (3)	95.35%	None
3D-ResNet	MRI (ADNI)	Binary	79.40%.	86.3%
CNN	sMRI, DTI (ADNI)	Binary	96.70%	None
AlexNet	fMRI (OASIS)	Multi-class (5)	94.97%.	95% 94%
DenseNet	CT	Multi-class (3)	87.36%	None
SqueezeNet	MRI (OASIS)	Multi-class (4)	82.53%	None
VGG-19	MRI (Kaggle)	Multi-class (4)	77.60%	81%
CNN+ResNet-18	MRI (ADNI)	Multi-class (3)	69.10%	None
ResNet101	MRI (ADNI, OASIS)	Multi-class (4)	93.33%	93%
EfficientNet Model	MRI ADNI	Binary	91.36%	83%
Custom-CNN	MRI	Binary	94.7%	None
MobileNet	MRI (ADNI+OASIS)	Multi-class (4)	83.97%	None
CNN+SVM	MRI (OASIS)	Binary	94.44%	None
VGG16	sMRI+FDG-PET	Multi-class (3)	93.75%	None
Proposed				
DenseNet121	MRI	Multi-class (3)	92.48%	96%
DenseNet169			93.00%	97%
DenseNet201			**96.05%**	**99%**

For various models using various the three approaches, the figure shows training hours and early stopping counts. The three methods without augmentation, with augmentation and selective augmentation are contrasted. Each model's training hours are displayed, and the early stopping percentages are written above the bars. This visual help us evaluate training effectiveness and early stopping trends across models and augmentation approaches (see Fig. 21).

8. Discussion

Deep learning approaches have the ability to effectively diagnose and categorize AD utilizing a variety of neuroimaging modalities, according to recent studies. In these works, many methodologies such as multimodality, TL-based techniques, and CNN technique have been investigated. Considering the contrast shown in Table 7, in our suggested method, data augmentation and feature extractor of TL strategy are used with a DenseNet model. We tested our method using a dataset of MR pictures that was made accessible to the public, and we classified the three stages of AD with an astounding 96.05 percent accuracy rate ND, MID, VMD. Our method also performed better in terms of precision, recall, and AUC than several earlier trials. Several studies have achieved high accuracy rates in the range of 77.60% to 95.35% [58,63]. Notably, the CNN+ResNet-18 model [59] achieved an accuracy of 69.10%, which is comparatively lower than other approaches. These results highlight the effectiveness of DenseNet models in capturing relevant features from MRI images and accurately classifying ND, MID and VMD cases. The DenseNet models outperform several previous approaches, including VGG-19, SqueezeNet [65], MobileNet [63], which achieved accuracy rates ranging from 77.60% to 83.97%. In this context, our research which we used DenseNet, a powerful CNN architecture known for its dense connectivity and feature reuse capabilities, to classify the three stages of AD. The implemented transfer learning technique can take advantage from generic features captured by DenseNet architecture which has already undergone training using the ImageNet dataset, and the top layers of deep CNNs then used for capturing domain-specific features.

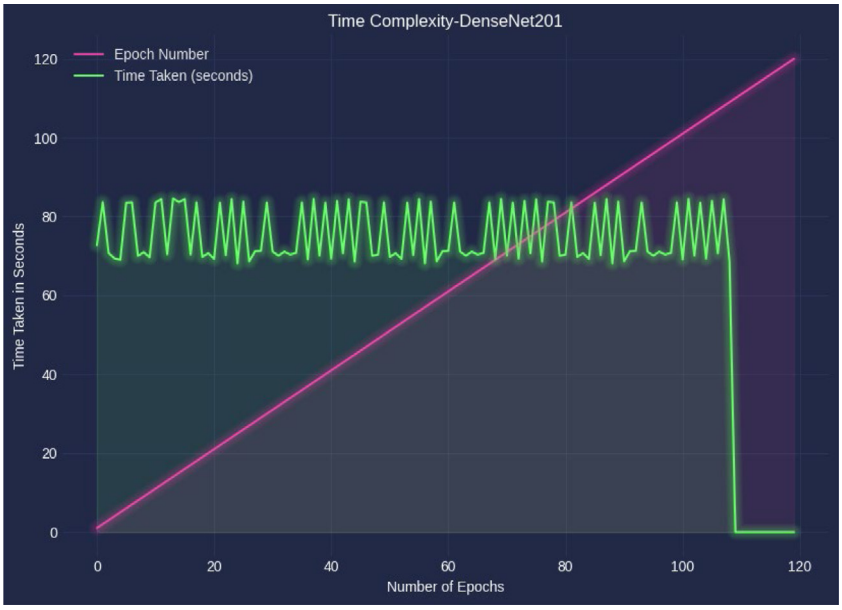

Fig. 20. Time taken for training DenseNet201 model.

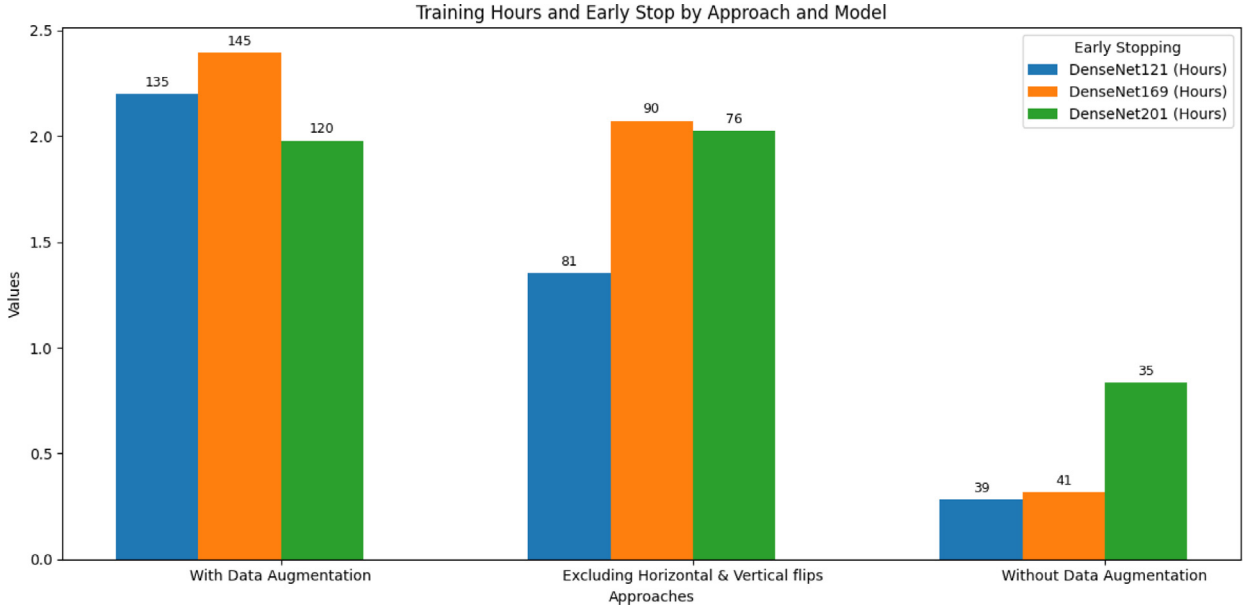

Fig. 21. Training hours and early stop by various models.

Deep learning approaches have the ability to effectively diagnose and categorize AD utilizing a variety of neuroimaging modalities, according to recent studies. In these works, many methodologies such as multimodality, TL-based techniques, and CNN technique have been investigated. Considering the contrast shown in Table 15, We thoroughly compared our technique to recent categorization studies. Our model, notably DenseNet201, performed exceptionally well. It outperformed previous trials, with a stunning 96.05% accuracy. Precision, recall, and AUC were all exceptionally high, demonstrating great diagnostic accuracy. AUC, an important parameter that is frequently overlooked, was an amazing 99%, highlighting the model's great class distinction. This emphasis on AUC distinguishes and strengthens our findings, filling a significant gap in many of recent works. These encouraging findings highlight the potential of our method in enhancing precise dementia categorization. Our approach is also able to effectively handle the challenges of limited training data. Additionally, our work's outstanding efficiency can be due to several important elements, including:

- DenseNet Connectivity: We take advantage of the DenseNet models' dense connectedness to effectively reuse features across layers. With this approach, the model performs more accurately in terms of categorization since it is better able to identify complex correlations and patterns in the dataset like neuroimaging. Additionally, it enables the extract pertinent characteristics from overlapping regions, enhancing its efficiency even in the face of such challenges.

- Unlike the datasets used in most earlier investigations, we used a publicly accessible collection of MR scans from online repository Kaggle. As a result, we can assess our method's effectiveness on a more diverse dataset.
- We made use of Keras' capability the technique called on-the-fly data augmentation. As a result, we were able to expand and diversify our training dataset, which enhanced the effectiveness of our method.
- Given the constraints on computational resources, we took a careful approach to adjusting hyperparameter We carefully set important hyperparameters like learning rates and batch sizes to enhance the model's convergence and overall performance even though we were unable to investigate advanced methodologies.
- DenseNet121, 169, and 201 are the three different DenseNet models that we compared. As a result, we were able to determine which design would accomplish the task the best.
- We present a thorough assessment of our model using four various approaches. How those factors in as we mentioned in four approaches affect the model's performance.
- Considering the computational and time complexity we also examined the performance of each model with respect to various approaches.

Limitations:

Understanding and acknowledging the following limitations is crucial for interpreting the findings and assessing the generalizability of our results.

- One limitation is the prolonged computational time, which restricted the exploration of larger datasets and more complex architectures, potentially impacting the overall results. The machine used for implementation faced challenges in terms of processing power and speed, affecting the efficiency and scalability of the experiments.
- Due to the significant processing needs of Grid Search CV, we were unable to use it. Instead, we manually fine-tuned our model's hyperparameters. While this method produced promising results, using Grid Search CV could have resulted in more complete optimization. With additional computer capacity, future study could explore this technique to further improve the performance of our model [58].
- Reliance on a single modality dataset (MRI), which may restrict the comprehensive understanding of the disease and overlook valuable information from other modalities.
- Since the optimal neural network structure is concern [54]. Manual determination of the number of hidden units in each layer. This approach may have resulted in suboptimal network structure for discovering more high-level and domain-specific features. Further research is needed to explore automated methods for learning the optimal network structure from large-scale data, especially for the practical implementation of deep learning in clinical settings.

9. Conclusion and recommendation

Healthcare decision support systems play a crucial role in clinical settings, assisting clinicians in accurate diagnosis and decision-making. In this study, a healthcare decision support system was developed using a transfer learning approach called DenseNet was used for multi-class classification of AD stages (ND, MID, VMD). MRI images are gone through an adequate preprocessing step. Our suggested model was tested on various test data to ensure its generalizability. with 92.48%, 93.00%, and 96.05%, DenseNet121, DenseNet169 and DenseNet201, in that order. This indicates that the transfer learning architecture that has been trained with MRI data, can be tested on test data, and attain a high level of accuracy. This suggests that the model has learned how to efficiently extract key elements from the MRI dataset and is able to generalize effectively to new, untested data and also showing the power of data augmentation on how it can be useful. Additionally, DenseNet201 has a remarkable AUC of 0.9901, proving its efficiency in recognizing many classes with excellent discriminative power.

In summary, these findings demonstrate the potential of the proposed model, trained on MRI data, as a promising approach for classifying different stages of AD. The proposed DenseNet model performed significantly better than two other variants of DenseNet, however, additional study is needed to increase its accuracy in MRI data and to generalsize even better.

The experimental findings on the presented dataset demonstrated the superiority of the proposed approach over previous methods, as evidenced by improved performance across a range of quantitative metrics. This work contributes to provide the use of DenseNet in prediction of AD, enhancing our comprehension of transfer learning's potential in achieving accurate classification models even with limited training data.

Moving forward, there is a plan to extensively investigate the capabilities of this framework for tackling other complex tasks, such as the identification of tumors.

To enhance the model performance, additional data preprocessing methods can be applied. Moreover, bringing more diversity in dataset by integrating data from multiple sources. In addition, we want to consider two image modalities in which the training in one set of data set and validation of the model in complete another dataset. This exploration provided insights into the strengths and weaknesses of DenseNet models in relation to other transfer learning approaches, highlighting their potential for other researchers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M. Odusami, R. Maskeliunas, R. Damaševičius, S. Misra, Comparable study of pre-trained model on Alzheimer disease classification, 2021, pp. 63–74, http://dx.doi.org/10.1007/978-3-030-86976-2_5.

[2] P. Vemuri, H.J. Wiste, B.S.D. Weigand, L.M. Shaw, J.Q. Trojanowski, M.W. Weiner, D.S. Knopman, R.C. Petersen, C.R. Jack, MRI and CSF biomarkers in normal, MCI, and AD subjects predicting future clinical change on behalf of the alzheimer's disease neuroimaging initiative*, 2009, www.loni.ucla.edu/ADNI/Collaboration/ADNI_Manuscript_Citations.pdf.

[3] G. Umbach, P. Kantak, J. Jacobs, M. Kahana, B.E. Pfeiffer, M. Sperling, B. Lega, Time cells in the human hippocampus and entorhinal cortex support episodic memory, Proc. Natl. Acad. Sci. 117 (45) (2020) 28463–28474, http://dx.doi.org/10.1073/pnas.2013250117.

[4] C. Salvatore, P. Battista, I. Castiglioni, Frontiers for the early diagnosis of AD by means of MRI brain imaging and support vector machines, Curr. Alzheimer Res. 13 (5) (2016) 509–533, http://dx.doi.org/10.2174/1567205013666151116141705.

[5] J. Islam, Y. Zhang, An ensemble of deep convolutional neural networks for Alzheimer's disease detection and classification, 2017, http://arxiv.org/abs/1712.01675.

[6] Y. Seo, H. Jang, H. Lee, Potential applications of artificial intelligence in clinical trials for Alzheimer's disease, Life 12 (2) (2022) 275, http://dx.doi.org/10.3390/life12020275.

[7] C.-H. Chang, C.-H. Lin, H.-Y. Lane, Machine learning and novel biomarkers for the diagnosis of Alzheimer's disease, Int. J. Mol. Sci. 22 (5) (2021) 2761, http://dx.doi.org/10.3390/ijms22052761.

[8] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, J. Tohka, Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects, NeuroImage 104 (2015) 398–412, http://dx.doi.org/10.1016/j.neuroimage.2014.10.002.

[9] W.Y. Ng, C.Y. Cheung, D. Milea, D.S.W. Ting, Artificial intelligence and machine learning for Alzheimer's disease: Let's not forget about the retina, Bri. J. Ophthalmol. 105 (5) (2021) 593–594, http://dx.doi.org/10.1136/bjophthalmol-2020-318407.

[10] M. Gharaibeh, M. Elhies, M. Almahmoud, S. Abualigah, O. Elayan, Machine learning for Alzheimer's disease detection based on neuroimaging techniques: A review, in: 2022 13th International Conference on Information and Communication Systems, ICICS, 2022, pp. 426–431, http://dx.doi.org/10.1109/ICICS55353.2022.9811143.

[11] A.P. Brady, R.G. Beets-Tan, B. Brkljačić, C. Catalano, A. Rockall, M. Fuchsjäger, The role of radiologist in the changing world of healthcare: A white paper of the European society of radiology (ESR), Insights Imaging 13 (1) (2022) 100, http://dx.doi.org/10.1186/s13244-022-01241-4.

[12] B. Malmir, M. Amini, S.I. Chang, A medical decision support system for disease diagnosis under uncertainty, Expert Syst. Appl. 88 (2017) 95–108, http://dx.doi.org/10.1016/j.eswa.2017.06.031.

[13] S. P.C., V. Sherimon, P. S.P., R. Nair, R. Mathew, A systematic review of clinical decision support systems in Alzheimer's disease domain, Int. J. Online Biomed. Eng. (IJOE) 17 (08) (2021) 74, http://dx.doi.org/10.3991/ijoe.v17i08.23643.

[14] S. Ghwanmeh, A. Mohammad, A. Al-Ibrahim, Innovative artificial neural networks-based decision support system for heart diseases diagnosis, J. Intell. Learn. Syst. Appl. 05 (03) (2013) 176–183, http://dx.doi.org/10.4236/jilsa.2013.53019.

[15] M.-A. Moreno-Ibarra, Y. Villuendas-Rey, M.D. Lytras, C. Yáñez Márquez, J.-C. Salgado-Ramírez, Classification of diseases using machine learning algorithms: A comparative study, Mathematics 9 (15) (2021) 1817, http://dx.doi.org/10.3390/math9151817.

[16] I.H. Sarker, Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions, SN Comput. Sci. 2 (6) (2021) 420, http://dx.doi.org/10.1007/s42979-021-00815-1.

[17] H.-P. Chan, R.K. Samala, L.M. Hadjiiski, C. Zhou, Deep learning in medical image analysis, 2020, pp. 3–21, http://dx.doi.org/10.1007/978-3-030-33128-3_1.

[18] H.L. Dawson, O. Dubrule, C.M. John, Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification, Comput. Geosci. 171 (2023) 105284, http://dx.doi.org/10.1016/j.cageo.2022.105284.

[19] A.S. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on MRI, Zeitschrift Für Medizinische Physik 29 (2) (2019) 102–127, http://dx.doi.org/10.1016/j.zemedi.2018.11.002.

[20] Y. Li, J. Zhao, Z. Lv, J. Li, Medical image fusion method by deep learning, Int. J. Cogn. Comput. Eng. 2 (2021) 21–29, http://dx.doi.org/10.1016/j.ijcce.2020.12.004.

[21] M. Bucholc, X. Ding, H. Wang, D.H. Glass, H. Wang, G. Prasad, L.P. Maguire, A.J. Bjourson, P.L. McClean, S. Todd, D.P. Finn, K. Wong-Lin, A practical computerized decision support system for predicting the severity of Alzheimer's disease of an individual, Expert Syst. Appl. 130 (2019) 157–171, http://dx.doi.org/10.1016/j.eswa.2019.04.022.

[22] M. Iman, H.R. Arabnia, K. Rasheed, A review of deep transfer learning and recent advancements, Technologies 11 (2) (2023) 40, http://dx.doi.org/10.3390/technologies11020040.

[23] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, 2018, pp. 270–279, http://dx.doi.org/10.1007/978-3-030-01424-7_27.

[24] A.W. Reza, M.M. Hasan, N. Nowrin, M.M. Ahmed Shibly, Pre-trained deep learning models in automatic COVID-19 diagnosis, Indones. J. Electr. Eng. Comput. Sci. 22 (3) (2021) 1540, http://dx.doi.org/10.11591/ijeecs.v22.i3.pp1540-1547.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, http://dx.doi.org/10.1109/CVPR.2009.5206848.

[26] H. Ravishankar, P. Sudhakar, R. Venkataramani, S. Thiruvenkadam, P. Annangi, N. Babu, V. Vaidya, Understanding the mechanisms of deep transfer learning for medical images, 2016, pp. 188–196, http://dx.doi.org/10.1007/978-3-319-46976-8_20.

[27] H.-C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285–1298, http://dx.doi.org/10.1109/TMI.2016.2528162.

[28] P.P. Dalvi, D.R. Edla, B.R. Purushothama, Diagnosis of coronavirus disease from chest X-ray images using DenseNet-169 architecture, SN Comput. Sci. 4 (3) (2023) 214, http://dx.doi.org/10.1007/s42979-022-01627-7.

[29] Y. Xu, Y. Wang, N. Razmjooy, Lung cancer diagnosis in CT images based on alexnet optimized by modified bowerbird optimization algorithm, Biomed. Signal Process. Control 77 (2022) 103791, http://dx.doi.org/10.1016/j.bspc.2022.103791.

[30] L. Dong, Y. Zhao, C. Dai, Detection of inception cavitation in centrifugal pump by fluid-Borne noise diagnostic, Shock Vib. 2019 (2019) 1–15, http://dx.doi.org/10.1155/2019/9641478.

[31] C.M. Vasile, A.L. Udriştoiu, A.E. Ghenea, M. Popescu, C. Gheonea, C.E. Niculescu, A.M. Ungureanu, Udriştoiu, Ştefan, A.I. Drocaş, L.G. Gruionu, G. Gruionu, A.V. Iacob, D.O. Alexandru, Intelligent diagnosis of thyroid ultrasound imaging using an ensemble of deep learning methods, Medicina 57 (4) (2021) 395, http://dx.doi.org/10.3390/medicina57040395.

[32] Y. Yu, H. Lin, J. Meng, X. Wei, H. Guo, Z. Zhao, Deep transfer learning for modality classification of medical images, Information 8 (3) (2017) 91, http://dx.doi.org/10.3390/info8030091.

[33] C. Kavitha, V. Mani, S.R. Srividhya, O.I. Khalaf, C.A. Tavera Romero, Early-stage Alzheimer's disease prediction using machine learning models, Front. Public Health 10 (2022) http://dx.doi.org/10.3389/fpubh.2022.853294.

[34] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551, http://dx.doi.org/10.1162/neco.1989.1.4.541.

[35] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.

[36] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, 2016, http://arxiv.org/abs/1608.06993.

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-ResNet and the impact of residual connections on learning, in: 31st AAAI Conference on Artificial Intelligence, in: AAAI 2017, 2017, pp. 4278–4284, http://dx.doi.org/10.1609/aaai.v31i1.11231.

[38] U.K. Lopes, J.F. Valiati, Pre-trained convolutional neural networks as feature extractors for tuberculosis detection, Comput. Biol. Med. 89 (2017) 135–143, http://dx.doi.org/10.1016/J.COMPBIOMED.2017.08.001.

[39] World Alzheimer Report 2010, www.deutsche-alzheimer.de.

[40] M.A. Morid, A. Borjali, G. del Fiol, A scoping review of transfer learning research on medical image analysis using ImageNet, Comput. Biol. Med. 128 (2021) 104115, http://dx.doi.org/10.1016/j.compbiomed.2020.104115.

[41] A. Bokade, A. Shah, Breast cancer diagnosis in mammography images using deep convolutional neural network-based transfer and scratch learning approach, Indian J. Sci. Technol. 16 (18) (2023) 1385–1394, http://dx.doi.org/10.17485/IJST/v16i18.39.

[42] A. Varghese, M. Jawahar, A.A. Prince, Fine-tuning ConvNets with novel leather image data for species identification, in: J. (Jessica) Zhou, W. Osten, D.P. Nikolaev (Eds.), Fifteenth International Conference on Machine Vision, ICMV 2022, SPIE, 2023, p. 17, http://dx.doi.org/10.1117/12.2679363.

[43] A.A. Mukhlif, B. Al-Khateeb, M.A. Mohammed, An extensive review of state-of-the-art transfer learning techniques used in medical imaging: Open issues and challenges, J. Intell. Syst. 31 (1) (2022) 1085–1111, http://dx.doi.org/10.1515/jisys-2022-0198.

[44] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, 2016, http://arxiv.org/abs/1608.06993.

[45] Densely Connected Convolutional Networks | Arthur Douillard, Retrieved January 6, 2023, from https://arthurdouillard.com/post/densenet/.

[46] Gaurav Singhal, Introduction to DenseNet with TensorFlow | pluralsight, 2020, https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow.

[47] W. Lin, T. Tong, Q. Gao, D. Guo, X. Du, Y. Yang, G. Guo, M. Xiao, M. Du, X. Qu, Convolutional neural networks-based MRI image analysis for the Alzheimer's disease prediction from mild cognitive impairment, Front. Neurosci. 12 (2018) http://dx.doi.org/10.3389/fnins.2018.00777.

[48] Pablo Ruiz, Understanding and visualizing DenseNets, in: Towardsdatascience.Com/Understanding-and-Visualizing-Densenets, 2018.

[49] R.K. Srivastava, K. Greff, J. Schmidhuber, Highway networks, 2015, http://arxiv.org/abs/1505.00387.

[50] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015, http://arxiv.org/abs/1512.03385.

[51] G. Larsson, M. Maire, G. Shakhnarovich, FractalNet: Ultra-deep neural networks without residuals, 2016, http://arxiv.org/abs/1605.07648.

[52] C. Shan, S. Gong, P.W. McOwan, Facial expression recognition based on local binary patterns: A comprehensive study, Image Vis. Comput. 27 (6) (2009) 803–816, http://dx.doi.org/10.1016/j.imavis.2008.08.005.

[53] D.A. Yudin, A.V. Dolzhenko, E.O. Kapustina, The usage of grayscale or color images for facial expression recognition with deep neural networks, 2020, pp. 271–281, http://dx.doi.org/10.1007/978-3-030-30425-6_32.

[54] H. Suk, il, S.W. Lee, D. Shen, Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis, NeuroImage 101 (569) (2014) http://dx.doi.org/10.1016/J.NEUROIMAGE.2014.06.077.

[55] C. Yang, A. Rangarajan, S. Ranka, Visual Explanations From Deep 3D Convolutional Neural Networks for Alzheimer's Disease Classification.

[56] A. Khvostikov, K. Aderghal, J. Benois-Pineau, A. Krylov, G. Catheline, 3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies, 2018.

[57] Y. Kazemi, S. Houghten, A deep learning pipeline to classify different stages of Alzheimer's disease from fMRI data, in: 2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB, 2018, pp. 1–8, http://dx.doi.org/10.1109/CIBCB.2018.8404980.

[58] S.A. Ajagbe, K.A. Amuda, M.A. Oladipupo, O.F. AFE, K.I. Okesola, Multi-classification of alzheimer disease on magnetic resonance images (MRI) using deep convolutional neural network (DCNN) approaches, Int. J. Adv. Comput. Res. 11 (53) (2021) 51–60, http://dx.doi.org/10.19101/IJACR.2021.1152001.

[59] M.W. Oktavian, N. Yudistira, A. Ridok, Classification of Alzheimer's disease using the convolutional neural network (CNN) with transfer learning and weighted loss, 2022.

[60] H. Ghaffari, H. Tavakoli, G. Pirzad Jahromi, Deep transfer learning–based fully automated detection and classification of Alzheimer's disease on brain MRI, Bri. J. Radiol. 95 (1136) (2022) http://dx.doi.org/10.1259/bjr.20211253.

[61] M. Sethi, S. Ahuja, S. Singh, J. Snehi, M. Chawla, An intelligent framework for Alzheimer's disease classification using EfficientNet transfer learning model, in: 2022 International Conference on Emerging Smart Computing and Informatics, ESCI, 2022, pp. 1–4, http://dx.doi.org/10.1109/ESCI53509.2022.9758195.

[62] R.A. Hridhee, B. Bhowmik, Q.D. Hossain, Alzheimer's disease classification from 2D MRI brain scans using convolutional neural networks, in: 2023 International Conference on Electrical, Computer and Communication Engineering, ECCE, 2023, pp. 1–6, http://dx.doi.org/10.1109/ECCE57851.2023.10101539.

[63] T. Ghosh, M.I.A. Palash, M.A. Yousuf, Md. A. Hamid, M.M. Monowar, M.O. Alassafi, A robust distributed deep learning approach to detect Alzheimer's disease from MRI images, Mathematics 11 (12) (2023) 2633, http://dx.doi.org/10.3390/math11122633.

[64] A. Rabeh, Ben, F. Benzarti, H. Amiri, CNN-SVM for prediction Alzheimer disease in early step, in: 2023 International Conference on Control, Automation and Diagnosis, ICCAD, 2023, pp. 1–6, http://dx.doi.org/10.1109/ICCAD57653.2023.10152440.

[65] M. Odusami, R. Maskeliūnas, R. Damaševičius, Pixel-level fusion approach with vision transformer for early detection of Alzheimer's disease, Electronics 12 (5) (2023) 1218, http://dx.doi.org/10.3390/electronics12051218.

[66] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, J. Tohka, Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects, NeuroImage 104 (2015) 398–412, http://dx.doi.org/10.1016/j.neuroimage.2014.10.002.

[67] C. Misra, Y. Fan, C. Davatzikos, Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: Results from ADNI, NeuroImage 44 (4) (2009) 1415–1422, http://dx.doi.org/10.1016/j.neuroimage.2008.10.031.

[68] E.F. Fang, Y. Hou, K. Palikaras, B.A. Adriaanse, J.S. Kerr, B. Yang, S. Lautrup, M.M. Hasan-Olive, D. Caponio, X. Dan, P. Rocktäschel, D.L. Croteau, M. Akbari, N.H. Greig, T. Fladby, H. Nilsen, M.Z. Cader, M.P. Mattson, N. Tavernarakis, V.A. Bohr, Mitophagy inhibits amyloid-β and tau pathology and reverses cognitive deficits in models of Alzheimer's disease, Nature Neurosci. 22 (3) (2019) 401–412, http://dx.doi.org/10.1038/s41593-018-0332-9.

[69] Tiara Williamson, What is the output of ImageDataGenerator? | deepchecks, 2021, https://deepchecks.com/question/what-is-the-output-of-imagedatagenerator/.

[70] B. Haq Alshara, Performance metrics in machine learning, 2020, https://www.javatpoint.com/performance-metrics-in-machine-learning.

[71] Frankie Cancino, Evaluation metrics for machine learning, 2021.