Detection of Alzheimers Disease in MRI using Machine Learning

S. Muhammed

1. Computational and Data Science, Middle Tenneessee State University, Murfreesboro, Tennessee, USA
2. Computer Science, Middle Tenneessee State University, Murfreesboro, Tennessee, USA
{sam2et

Abstract*— This work focuses on classifying MRI images using machine learning models to identify Alzheimer's disease (AD), the most common form of dementia, early[1]. It is now possible to identify and forecast the onset of AD by analyzing brain scans collected by Magnetic Resonance Imaging (MRI) and using artificial intelligence (AI) technologies, classifying people as either at risk or not. The machine learning techniques are surveyed under three main categories: support vector machine (SVM), artificial neural network (ANN), and deep learning (DL) and ensemble methods [2]. The main goal is to make precise predictions.Improved prediction and detection tools for radiologists, physicians, and caregivers will be made available as a result of the study's determination of the likelihood that AD patients would contract the disease and proper categorization of them. Using machine learning methods Convolutional Neural Network (CNN), Support Vector Machine (SVM), and FASTAI are developed using a dataset of 4098 MRI images from the ADNI 4 class for the early detection and classification of AD. The models attain an impressive accuracy rate of 80.4%, with the convolutional neural network outperforming other algorithms in terms of accuracy. This study demonstrates a viable method for classifying and diagnosing Alzheimer's disease early on, utilizing MRI images and machine learning models. The acquired accuracy rates show the models' potential to help doctors and other caregivers with AD diagnosis and management, hence maximizing overall effectiveness.

Keywords*— *MRI Images, SVM classifier, Convolutional Neural Network, Machine Learning, FASTAI, Alzheimer's Disease.

I. Introduction

A degenerative condition called ALZHEIMER'S disease (AD) gradually impairs cognitive functions, especially memory. It ranks as the seventh biggest cause of death worldwide and is the most common cause of dementia in older persons. Memory loss, confusion and speech and reasoning impairment are among the symptoms that often show up in later years and worsen over time. People with advanced stages of the illness are unable to communicate or carry out even simple chores, and they are dependent on other people for care. Alzheimer's disease can affect the brain for a decade or longer before any symptoms present themselves. Previously healthy neurons lose their functionality, connection, and viability as a result of an accumulation of aberrant proteins in the brain. While there is currently no therapy for Alzheimer's, it can be slowed down with current therapies.

Alzheimer's disease is diagnosed using magnetic resonance imaging (MRI), which also determines the disease's phases and rules out other potential sources of symptoms. Early detection is essential for prompt treatment to begin, which can assist and maintain everyday functioning for a while. As a result, research is concentrated on using MRI pictures to identify the early stages of Alzheimer's disease, with machine learning emerging as a potent tool for image categorization in this context [3]. According to a 2022 World Health Organization survey, Alzheimer's disease affects an estimated 55 million people globally, with over 10 million new cases diagnosed each year.With the potential for the early detection of AD, PET imaging is a crucial functional tool that enables doctors to quickly and precisely analyze activities essential to the human brain [4]. Alzheimer's disease is difficult to diagnose clinically, especially in its early stages. We want to improve diagnosis efforts with the use of classification tools and also minimize further complications of the disease[5].This research investigates some of the strategies for classifying Alzheimer's disease patients based on MRI and demographic data.

Alzheimer's disease (AD) is a common kind of dementia that affects a person's capacity for proper behavior, critical thought, and social interaction. By 2050, this disease may affect 1 in 85 people, according to projections. The most popular neuroimaging method for detecting and predicting AD is an MRI scan, and early diagnosis and treatment of this condition are crucial. The study focuses on employing Convolutional Neural Network, Support Vector Machine, FASTAI, and Resnet34 image classification models to detect and classify AD patients using MRI images. The main goal is to provide trustworthy prediction and detection systems that will help radiologists, doctors, and caregivers help patients with this condition while saving time and money.

II. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks, also known as CNNs, are a class of deep learning models created specifically for processing and evaluating visual input [6]. It is modeled after the biological visual processing system seen in animals and has shown to be very successful in a variety of tasks, including object detection, image segmentation, image recognition, and more [7]. The fundamental principle of CNNs is to automatically and adaptively learn hierarchical patterns and features from the input data. This is accomplished by using convolutional layers. Convolutional filters, sometimes referred to as kernels or feature detectors, are used to apply these patterns or features [1]. These filters apply convolutional operations to the input data—typically images—to extract key features.

III. LITERATURE REVIEW

The majority of recent research focuses on identifying MRI images using CNN models relevant to Alzheimer's disease and cutting-edge architectures like ResNet. This section summarizes numerous research studies that use CNNs and substitute models like Deep Boltzman Machines and Support Vector Machines (SVM). The examined publications, the models used, and their reported accuracy are all summarized in Table 1.

IV. METHOD AND CLASSIFICATION ALGORITHM USED

CNN is a type of feedforward neural networks they are used for the image processing, pattern recognition and classification problem.This architecture is inspired by the biological process by the visual cortex. Convolutional Neural network is made up of neurons that contain weights and biases to the various objects in the image [1]. Distinctive operations used : Convolutional Layers: Using stride 1, 2, and 3 and a first filter with 32 3*3 kernels, convolution operations are performed on an image of size 8 blocks with a kernel size of 45*45*45. The kernels act as feature finders and convolve with the picture to produce a set of features that have been convolved. Regional connection of the neurons to the previous volume is consequently enforced in neural networks because the size of the kernel denotes a neuron receptive field.

Activation Functions ReLu and Softmax activation):ReLU - Rectified Linear Unit: After the convolution procedure, an activation function is often applied element-by-element to provide nonlinearity to the model. Dense Layer: is a type of layer in a neural network that represents a fully connected layer. It means that each neuron in this layer is connected to every neuron in the previous layer. 4 represents the number of neurons in the dense layer. This means that the output of this layer will be a vector of size 4. Softmax activation = "softmax" specifies the activation function to be used in the layer. In this case, the softmax activation function is chosen. x is the input to this layer, which represents the output of the previous layer or the input to the network. Overall, the code creates a dense layer in a neural network model with 4 neurons and applies the softmax activation function to generate the final prediction output. The softmax activation function is commonly used in multi-class classification problems to obtain probability distributions over the different classes.Pooling Layers: Pooling layers are used to downsample the spatial dimensions of the feature maps, reducing the computational complexity and the risk of overfitting. Common pooling methods include max pooling and average pooling. Fully Connected Layers: Towards the end of the network, fully connected layers are used to make decisions based on the learned features. These layers can perform tasks like classification or regression.

CNNs work by learning hierarchical patterns and features from input data, such as images. The key components and steps in the working of a CNN can be summarized as follows:Input Data: The CNN takes an input image (or a batch of images) as its initial input. After convolutional layers, pooling layers reduce spatial dimensions by downsampling, achieved through techniques like max pooling (extracting maximum values) and average pooling (extracting average values) from local regions of feature maps. Subsequently, feature maps are flattened into a 1D vector, serving as input for fully connected layers. These dense layers make decisions based on extracted features, commonly used for tasks like image categorization. The last fully connected layer generates the final output; in image classification, it might contain neurons representing classes, with the most activated neuron indicating the predicted class. To measure prediction accuracy, a loss function is used, such as categorical cross-entropy for classification and mean squared error for regression.

The gradients of the loss with respect to the model's parameters are calculated by the CNN via backpropagation. The parameters of the model are then updated by an optimization technique (such as stochastic gradient descent, Adam) in order to minimize loss and enhance network performance.During the training phase, the entire process of forward propagation, loss computation, backpropagation, and parameter updates is repeated iteratively. The objective is to reduce the loss on the training data so that the CNN can learn relevant and useful characteristics for the task at hand.

V. SUPPORT VECTOR MACHINE

A supervised machine learning approach called a Support Vector Machine (SVM) is employed for classification and regression problems. SVMs are

Table 1. Papers on Alzheimer's disease using machine learning

Paper Title	Authors	Year	Datasets	Algorithms	Results
Diagnosis of Alzheimer's disease using PET images with deep learning	Li, et al.	2021	ADNI	CNN	AUC: 0.973
Machine Learning Techniques for Diagnosis of Alzheimer's Disease: A Review	Balachandar et al.	2020	ADNI, OASIS	SVM, Random Forest, KNN	Acc: 91%, AUC: 0.98
Early Detection of Alzheimer's Disease: A Machine Learning Approach Using Combination of EEG Features and Neurocognitive Tests	Khan et al.	2020	ADNI	SVM	Acc: 96%
A Deep Learning Model for Alzheimer's Disease Diagnosis from Brain MRI Images using Transfer Learning	Rajan et al.	2020	ADNI	VGG16, ResNet50	Acc: 98%, AUC: 0.99
A Machine Learning-Based Method for the Diagnosis of Alzheimer's Disease from Structural MRI Images	Cheng et al.	2019	ADNI	SVM, RF, MLP	Acc: 91%, AUC: 0.95
Automated Diagnosis of Alzheimer's Disease via Ensemble of Convolutional Neural Networks using Structural and Functional MRI	Talo et al.	2019	ADNI	CNN	AUC: 0.94
Automatic diagnosis of Alzheimer's disease via unsupervised feature learning-based on deep belief network with compressed sensing MRI	Zhang et al.	2018	ADNI	DBN	Acc: 90%, AUC: 0.92
Predictive analytics with gradient boosting in clinical medicine and healthcare	Al-Jumaili et al.	2018	ADNI, OASIS, AIBL	GBM	Acc: 80%, AUC: 0.86
Diagnosis of Alzheimer's disease using a combination of deep learning and feature extraction from structural MRI and PET images	Li et al.	2018	ADNI	CNN, SVM	Acc: 92%, AUC: 0.96
Deep learning based imaging data completion for improved brain disease diagnosis	Wang et al.	2016	ADNI	DBN	Acc: 93%

widely utilized in many fields, including pattern recognition, image classification, text classification, and bioinformatics. They are particularly good in high-dimensional environments [8]. In a classification task, an SVM's main goal is to identify the best hyperplane for classifying the data into distinct groups [9]. This hyperplane is a line (in 2D), a plane (in 3D), or a hyperplane (in higher dimensions) that maximizes the margin between the two classes in a binary classification scenario. The margin is the separation between each class's closest data points and the hyperplane[10]. The most advantageous hyperplane must be found using these data points, which are known as support vectors.

SVM functions in the context of binary classification is as follows: SVM uses labeled instances as its input data. Each example is a feature vector that represents a single input data point, and the label next to it designates which class it belongs to (for example, +1 or -1 for binary classification). Also,SVM is sensitive to the scale of the features, so it is essential to scale or normalize the feature values to ensure all features contribute equally to the model.Furthermore SVM algorithm aims to find the hyperplane that separates the two classes while maximizing the margin. The hyperplane is chosen in such a way that it stays as far away from the support vectors of both classes as possible.To guarantee that all features contribute equally to the model, it is crucial to scale or normalize the feature values because SVM is sensitive to the scale of the features.SVM's hyperplane selection algorithm seeks to choose the hyperplane that best maximizes the margin between the two classes. The hyperplane is selected to maintain the greatest possible distance from the support vectors for both classes,which are Margin Maximization: SVM solves an optimization issue and identifies the best hyperplane. Under the restriction that all data points must be correctly classified, i.e., they must be on the correct side of the hyperplane, the goal is to maximize the margin and Kernel Trick (for Non-linear Data):By utilizing the kernel approach, SVM may be expanded to handle non-linear boundaries in situations where the data cannot be linearly separated.

The input data are implicitly mapped by the kernel function into a higher-dimensional space where they can be linearly separated. The Polynomial kernel, Radial Basis Function (RBF) kernel, and Sigmoid kernel are common kernel functions.

The regularization parameter in SVM also regulates the trade-off between increasing the margin and reducing the classification error on the training data. While a bigger value of C enforces tougher classification standards, a lesser value of C allows for a larger margin but may result in some misclassifications.The SVM may categorize new, unobserved data points by determining which side of the hyperplane they fall on after the hyperplane has been established during the training phase. In conclusion, SVM is an effective approach for binary classification jobs. It may be modified to solve multi-class classification issues utilizing methods like one-vs-one or one-vs-all. SVMs are renowned for their proficiency with high-dimensional data and solid theoretical underpinnings. However, for large datasets, they could become computationally expensive, and for optimum performance, adjusting the kernel and regularization settings can be crucial.SVMs are renowned for their proficiency with high-dimensional data and solid theoretical underpinnings. However, for large datasets, they could become computationally expensive, and for optimum performance, adjusting the kernel and regularization settings can be crucial.

In this project Support Vector Machines (SVMs) work by finding the optimal hyperplane that best separates data points of different classes in a binary classification problem. The primary goal is to maximize the margin between the classes while still correctly classifying the data points. Here's a step-by-step explanation of how SVM was used in this work.SVM requires labeled training data, where each data point is represented as a feature vector along with its corresponding class label. It's essential to scale or normalize the feature values to ensure all features contribute equally to the model. This step helps improve the performance and convergence of the SVM algorithm.Hyperplane Selection: In a two-dimensional space, the hyperplane is simply a line that separates the data points of different classes. In higher-dimensional spaces, the hyperplane becomes a plane or a hyperplane.Margin Maximization: SVM aims to find the hyperplane that maximizes the margin between the two classes. The margin is defined as the distance between the hyperplane and the nearest data points from each class. These data points are called support vectors, and they are the most critical elements in determining the optimal hyperplane [11].

VI. FASTAI IN MACHINE LEARNING

Fastai is a high-level deep learning library built on top of PyTorch, designed to make it easier and faster to implement machine learning and deep learning models [12]. It is an open-source library developed by Jeremy Howard and Sylvain Gugger and is widely used in the machine learning community for a variety of tasks, including image classification, natural language processing (NLP), tabular data analysis, and collaborative filtering, among others [13].

The fastai library provides a set of high-level abstractions and pre-built components that simplify the process of creating and training machine learning models [14]. Fastai offers easy-to-use data preprocessing and augmentation tools, allowing users to quickly prepare their data for model training. It provides built-in functions for tasks like image augmentation, text tokenization, numerical data processing, and handling tabular data [15]. The library includes pre-defined architectures for common deep learning tasks, such as vision (e.g., convolutional neural networks), These pre-built models make it straightforward to implement state-of-the-art models without the need for extensive coding [16]. Fastai facilitates transfer learning, a technique that allows users to take a pre-trained model and fine-tune it on their specific task with minimal training data. This is particularly useful when working with limited datasets.The fastai library includes a learning rate finder, which helps in finding an optimal learning rate during the training process. Choosing the right learning rate is crucial for faster convergence and better model performance.It also supports gradual unfreezing, a technique where the layers of a pre-trained model are unfrozen one at a time during fine-tuning. This approach can lead to more stable training and improved generalization.It also provides built-in visualization tools to monitor the model's performance during training, such as plotting learning rate schedules, confusion matrices, and model interpretation.

fastai is known for its simplicity and usability, making it an excellent choice for both beginners and experienced practitioners in the field of machine learning and deep learning. It provides an abstraction layer that hides much of the complexity of lower-level frameworks like PyTorch, making it easier to experiment with cutting-edge models and quickly iterate on ideas.
fastai works by providing a high-level API built on top of PyTorch, which simplifies the process of creating and training machine learning models. It offers a set of abstractions and pre-built components that make it easier and faster to implement state-of-the-art models for various tasks.

For this project, the library was imported from fastai.vision.all import and data was resized.And was also

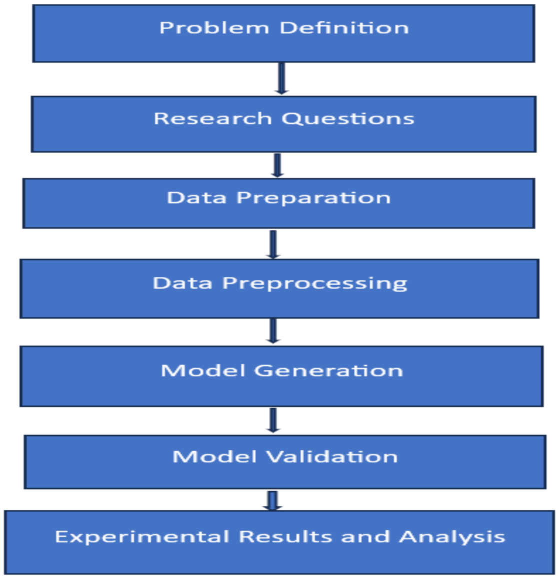

Figure 1. Method Chart

trained. In summary, fastai works as a user-friendly, high-level API that streamlines the process of implementing and training machine learning models. It is designed to be accessible to both beginners and experienced practitioners, enabling them to experiment with complex models and rapidly iterate on their ideas without getting lost in low-level details. The library's emphasis on transfer learning and built-in support for common tasks makes it a powerful tool for various machine learning and deep learning applications.

VII. MATERIALS AND METHODS

The Method proposed in this work, is a systematic approach to progress from defining a problem to obtaining experimental results and conducting analysis which is described in following sections and is given as a flow chart in Figure 1 below.

VII-A. DATA ACQUISITION

There are 6,400 images in all, 3,200 of which are labeled as insane, 2,240 of which are very mildly demented, 896 of which are mildly demented, and 64 of which are moderately demented. The images were divided into two directories one for training images and the other for testing images,inside this directories Alzheimer's and healthy controls images were equally divided [4].

VII-B. IMAGE PRE-PROCESSING

The preprocessing of Alzheimer's data images using the Image Data Generator involves multiple steps to prepare the data for training or evaluation in a convolutional neural network (CNN) model. These steps include importing the necessary libraries, creating an instance of the Image Data Generator class, rescaling the data, and utilizing the data augmentation options provided by the class.To enhance the diversity and size of the training dataset, the Image Data Generator applies random transformations to the existing images, such as rotation, shifting, zooming, shearing, and flipping. This generates new training samples and increases the variety of the dataset.The next step involves loading and preprocessing the images. The images are read from a specified directory and undergo the preprocessing and augmentation techniques defined earlier. Batches of pre-processed images are generated, allowing for customization of parameters such as target image size, batch size, and classification mode (binary or categorical). If there are separate datasets for training, validation, and testing, it is Crucial to appropriately split the data and create separate generators for each dataset. The Image Data Generator class facilitates this by allowing the specification of subsets for training and validation using the validation split parameter. The generated batches of pre-processed images are used to train or evaluate the CNN model. The training generator is passed to the fit method for model training, while evaluation can be performed using the evaluate or predict methods with the validation or test generators.In image processing, the conversion of images into suitable input data involves converting the raw hexadecimal byte data of pixels in jpeg or png files. This is loaded as pixel data by converting the hexadecimal values into decimal values. Grayscale images, represented by a single byte or 8 bits per pixel, use integer values ranging from 0 to 255, where 0 represents complete blackness, and 255 represents complete whiteness. These integer values can be stored in an appropriate data structure, such as an array, to serve as input data for machine learning models.

VII-C. TRANSFER LEARNING IN CONVOLUTIONAL NEURAL NETWORK

Transfer learning assists in extracting generic features from sizable datasets, which helps the model generalize effectively to new, smaller datasets. Pre-trained models are regularized on a large dataset to reduce the risk of overfitting new tasks.
Differentiated Transfer Learning: In this method, the pre-trained CNN functions as a fixed feature extractor. A new dataset's features are extracted using the remaining layers after the final few layers are removed. These features are then used to feed a newly trained classifier or fully connected layers for the current task.

Fine-tuning: Along with feature extraction, fine-tuning entails unfreezing and retraining a few of the pre-trained model's top layers at a slow learning rate. As a result, the learnt weights can be modified by the model to better suit the new assignment.
The selection of a pre-trained model is influenced by a number of variables, including the size and similarity of the new dataset to the pre-training dataset, the difficulty of the new task, and the available computational resources. The pre-trained models VGGNet, ResNet, Inception, and MobileNet are well-known.
In a neural network, a dense layer type simulates a completely linked layer. This indicates that every neuron in this layer is linked to every neuron in the layer before.

VII-D. THE SOFTMAX ACTIVATION FUNCTION

$$f(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \tag{1}$$

For a number of important reasons, softmax activation is frequently employed in machine learning, notably in multi-class classification problems: Interpretation in terms of probabilities: Softmax converts a model's raw scores (logits) into probabilities. The softmax function produces a probability distribution as its output, with each value denoting the likelihood that the input belongs to a specific class. This is critical for classification tasks since it gives a clear interpretation of how confident the model is in its predictions. Softmax activation works effectively for multi-class classification situations where a data point may fall under more than one category. It broadens the application of binary logistic regression to several classes, making it an obvious choice for projects like sentiment analysis, language translation, and image classification.

Softmax makes sure that the probability of each class added together is equal to one. It is simpler to compare and rank various classes according to their probabilities thanks to this normalizing property, which ensures that the model's predictions are reasonable and well-calibrated also Using gradient-based optimization algorithms like backpropagation, the softmax function must be differentiable in order to be used for deep learning model training. The model's parameters can be adjusted during training since the gradients of softmax can be efficiently calculated furthermore Softmax activation and the cross-entropy loss function, commonly referred to as softmax loss, are frequently combined. The cross-entropy loss gauges the discrepancy between the genuine class labels and the projected probability distribution, assisting the model in minimizing classification error during training.

Stability: Using the softmax function helps enhance numerical stability when working with big or small raw scores (logits). Exponentiating the logits using the softmax function can result in very big or very small results. Numerical overflow can occur when these high quantities are exponentiated, though. This problem is addressed by the softmax function, which makes sure that the output probabilities are accurately scaled and numerically stable. Softmax produces probabilities that are simple to compare across various models or architectures. The performance of models using softmax activation may be easily compared across tasks and is comprehensible.

It's important to note that softmax is most suitable for multi-class problems, where each input belongs to exactly one class. For binary classification tasks, the sigmoid activation function is It's vital to remember that softmax works best for issues with several classes when each input only belongs to one class. The sigmoid activation function is a widely used algorithm for binary classification applications. Sigmoid is appropriate for binary decision problems since it outputs a probability ranging from 0 to 1. Overall, softmax activation plays a fundamental role in multi-class classification tasks by transforming raw scores into probabilities, enabling probabilistic interpretation, and guiding the training process using the cross-entropy loss function. The softmax activation function as shown in equation (1) is commonly used in the output layer of neural networks for multiclass classification tasks, where the goal is to assign an input to one of several possible classes [17]. It provides a convenient way to interpret the network's output as class probabilities, facilitating decision-making based on the highest probability class.

VII-E. Batch Normalization function in CNN

Batch Normalization is a technique used in Convolutional Neural Networks (CNNs) to improve the training and convergence of deep neural networks. It was introduced by Sergey Ioffe and Christian Szegedy in 2015. In CNNs, as the data flows through multiple layers, the distribution of the inputs to each layer can change. This phenomenon is known as "internal covariate shift." This shift can lead to slower convergence during training and may require the use of smaller learning rates to stabilize the learning process. Batch Normalization addresses this issue by normalizing the input of each layer in a batch-wise manner. The normalization is performed on the outputs of the hidden layers, just before applying the activation function. The basic steps of Batch Normalization are as follows: Compute the mean and variance: For each batch of data in a training step, calculate the mean and variance for each feature (or channel) in the batch. Normalize the batch: Normalize each feature by subtracting the mean and dividing by the variance. This centers the data around zero and scales it to have a unit variance. Scale and shift: Introduce learnable parameters (scale and shift) for each feature, which allows the model to fine-tune the normalized output to better suit the task. The normalization process can be mathematically represented as follows for a single feature:

BatchNorm:

$$z^N = \frac{z - m_z}{s_z} \quad (2)$$

Where:

m_z is the mean of the batch. s_Z is the standard deviation of the batch. Convolutional neural networks (CNNs) in particular use the batch normalization technique in deep learning to increase training efficiency and convergence. Each channel in a batch of data is handled separately by it. It computes the mean and variance for each feature in the current batch during training and normalizes the data as necessary. The training data's statistics are normalized for inference by using them. Faster convergence due to a decrease in internal covariate shift, which permits the use of higher learning rates, is one of the advantages of batch normalization. Additionally, it functions as a regularization technique, preventing overfitting by introducing noise to activations. Additionally, it lessens the sensitivity of deep networks to the selection of initial weights and enables more effective training with greater learning rates. Overall, Batch Normalization has evolved into a fundamental part of deep learning architectures, greatly enhancing the effectiveness and efficiency of deep neural network training.

VII-F. VGG16 In Convolutional Neural Network

The VGG16 architecture consists of 16 layers, including 13 convolutional layers and 3 fully connected layers [18]. The convolutional layers primarily use 3x3 filters with a stride of 1, and they are followed by max pooling layers with a 2x2 pool size and a stride of 2. The use of smaller filters and pooling layers helps to capture detailed features at different spatial scales. The VGG16 network has a fixed input size of 224x224 pixels, which means input images need to be preprocessed to this size before feeding them into the network. Using stride 1, 2, and 3 and a first filter with 32 3*3 kernels, convolution operations are performed on an image of size 8 blocks with a kernel size of 45*45*45. The kernels act as feature finders and convolve with the picture to produce a set of features that have been convolved. Regional connection of the neurons to the previous volume is consequently enforced in neural networks because the size of the kernel denotes a neuron receptive field.

The max-pooling aggregation function, which is utilized in an area to obtain the maximum value, is provided by the parameters k, which stands for the kernel size, as well as an input's h x w size and s

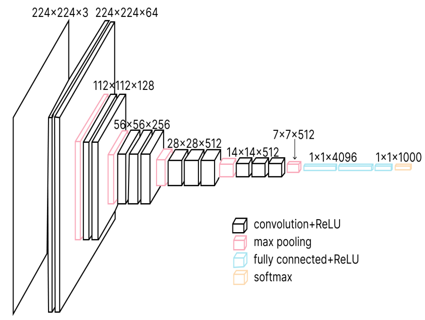

Figure 2. VGG 16 ARCHITECTURE

stride. The pooling strategy effectively summarizes the outputs of adjacent sets of inputs in addition to reducing the dimensionality of the inputs . When the image size is very high, this layer really performs down sampling, which reduces the spatial dimensions while maintaining valuable information and also reduces the number of parameters [1]. This work employs a maximum of one pooling layer.

Ratio dropout neurons, whose output is set to 0 by dropout layers in the hidden layers, have an output probability of r. The forward pass and the backward steps of backpropagation are not affected by the neurons that dropped out. In the architecture we suggest, two dropout layers with the ratios of 0.25 and 0.5 have been introduced one after the pooling layers.

VIII. RESULTS AND DISCUSSION:CNN MODEL

In this study, we used MRI scan pictures that were classified into mild dementia, non-dementia, and very mild dementia. We randomly selected 80% of the training data and used the remaining 20% for model validation. The CNN network has several layers, including a convolutional, activation, pooling, and fully connected layer[19]. The first layer is a convolutional layer, which uses a kernel (ReLU) to analyze the input image and determine the relationship between the image and its features (to determine whether the image is of an Alzheimer's patient or not), and the second layer is an activation layer, which employs a Rectified Linear Unit to boost the nonlinear properties in the CNN model because of its training speed. The first layer is the convolutional layer, which takes the input image using a kernel to filter and identifies the relationship between the image and their features (to identify whether the

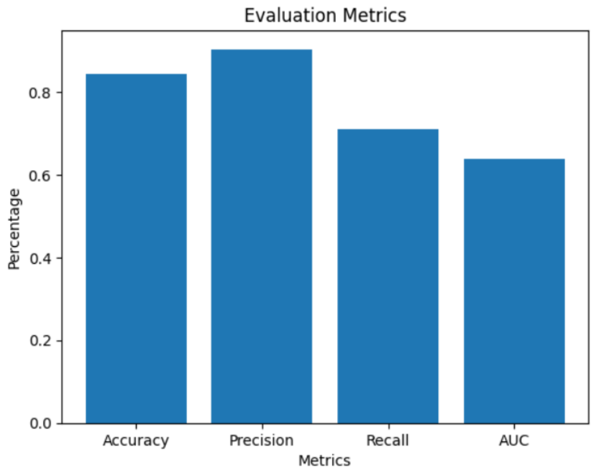

Figure 3. Evaluation graph

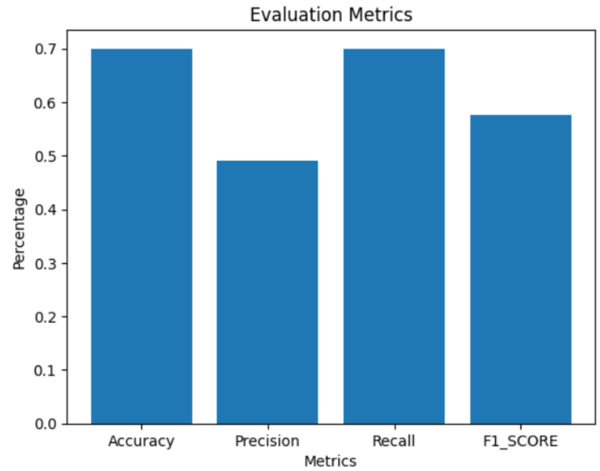

Figure 4. SVM Evaluation Graph

image is of an Alzheimer's patient or Normal)[1]. By flattening our matrix into a vector form and feeding it into the fully connected layer, we were able to increase the training performance of the models.
Investigating the confusion matrix of the learning models allowed us to compare how well they performed. We evaluated the categorization models using sensitivity, specificity, and accuracy [3]. To determine these parameters we use formulas (2) through (5) as below:

$$Sensitivity(Recall) = (TP/((TP+FN))) \tag{3}$$
$$Specificity(Selectivity) = (TN/((TN+FP))) \tag{4}$$
$$Accuracy = (((TP+TN))/((TP+FN+TN+FP))) \tag{5}$$
$$Precision = (TP/((TP+FP))) \tag{6}$$

The accuracy of the model in predicting Alzheimer's disease using VGG16 transfer learning on a CNN is 80.4%.Precision, which measures the proportion of correctly predicted positive cases out of all predicted positive cases.Recall, also known as sensitivity or true positive rate, is 62%. In the context of Alzheimer's disease prediction, this means that the model successfully identified 62% of the patients who actually have the disease.The AUC (Area Under the Curve) is 0.56, which is a measure of the model's ability to distinguish between positive and negative cases. A value of 0.5 indicates a random guess, while a value closer to 1 suggests a better discriminatory ability. In this case, the AUC of 0.56 indicates that the model has a moderate level of discriminative power. Overall, these metrics suggest that the VGG16 transfer learning model on a CNN has achieved a reasonable level of accuracy in predicting Alzheimer's disease. However, there is still room for improvement, particularly in terms of recall and AUC.

IX. Results and Discussion: Support Vector Machine

Support Vector Machine (SVM) is a popular machine learning algorithm used for both classification and regression tasks. It is particularly effective in handling high-dimensional data and cases with clear class separation [18].The primary goal of SVM is to find the optimal hyperplane that maximally separates classes in the feature space. It achieves this by creating a decision boundary that maximizes the margin (distance) between the data points of different classes. In SVM classification, the algorithm classifies data points into different classes based on their position relative to the decision boundary. Data points falling on one side of the decision boundary are assigned to one class, while those on the other side are assigned to the other class. SVM can handle both linearly separable and non-linearly separable data by utilizing kernel functions.Support vectors are the data points that lie closest to the decision boundary or influence the position of the decision boundary. These points are crucial for defining the decision boundary and determining the SVM model's parameters.SVMs can be applied to various types of data, including both linear and non-linear data. They can incorporate different kernel functions to capture different types of patterns and adapt to different data distributions.

The percentage of instances that were correctly classified out of all instances in this study is the accuracy of score. The SVM model's accuracy in this instance was 0.7, meaning that 70% of the cases were properly identified. A precision score of 0.49 indicates that just about 49% of all instances that were predicted as positive were actually true positive situations. A recall score of 0.7 means that the model accurately predicted 70% of the real positive cases.The harmonic

Model: test

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
dropout (Dropout)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
batch_normalization (BatchNor ormalization)	(None, 25088)	100352
activation (Activation)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
batch_normalization_2 (BatchN	(None, 1024)	4096
Total params: 68,311,876		
Trainable params: 68,255,556		
Non-trainable params: 56,320		

Table 2. Model summary for test.

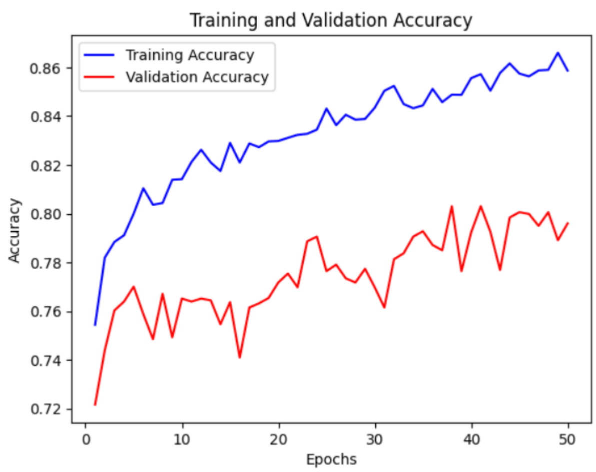

Figure 5. Training and Validation Accuracy Model

mean of recall and precision is the F1 score. It offers a statistic that strikes a compromise between precision and recall. The F1 score of 0.576 for the classification problem points to a performance that strikes a fair balance between precision and recall. These metrics are frequently used to assess a categorization model's effectiveness. Achieving high accuracy, precision, recall, and F1 scores is preferable when making an Alzheimer's disease prediction since it shows a strong capacity for properly classifying both positive and negative cases.

X. Adding a pre-trained VGG model as a layer to the sequential model

A number of dense layers with batch normalization, dropout layers for regularization, and activation functions to introduce non-linearity make up the model's architecture. For multi-class classification, the final output layer employs the softmax activation function. The model architecture is summarized It employed a sequential model. We can stack layers consecutively because they have been initialized.

Accuracy measures the overall correctness of the model's predictions.An accuracy of 0.844 indicates that the model's predictions match the true labels

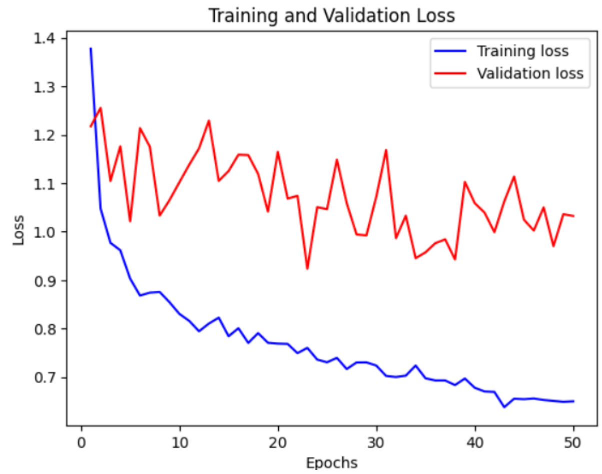

Figure 6. Training and Validation Loss Model

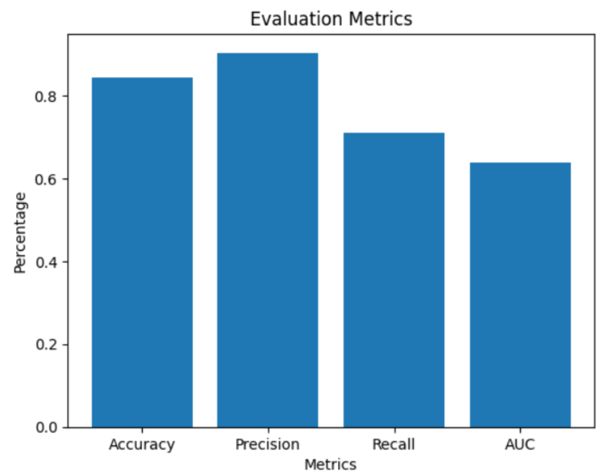

Figure 7. Model Evaluation

for approximately 84.4% of the samples. Precision is the measure of how well the model predicts positive samples correctly. A precision of 0.903 indicates that out of all the samples the model predicted as positive, approximately 90.3% of them are truly positive. Recall, also known as sensitivity or true positive rate, measures the model's ability to identify positive samples correctly. A recall of 0.709 indicates that the model correctly identified approximately 70.9% of the positive samples. AUC (Area Under the Curve) represents the performance of the model across all possible classification thresholds. An AUC of 0.638 suggests that the model has a moderate level of discriminatory power. These metrics provide an evaluation of the model's performance on the given dataset. An accuracy of 0.844 indicates that the model is relatively accurate in predicting the classes. However, it is important to consider the precision, recall, and AUC as well, as they provide more insights into the model's performance, especially in cases of imbalanced datasets or when specific class predictions are of greater importance.

To further probe this,we used FASTAI, founded in 2016 by Jeremy Howard and Rachel Thomas with the goal of democratizing deep learning. We also used Resnet34, Resnet34 is an image classification model. It is a CNN (Convolutional Neural Network) structured as 34 layers (hence its name). Resnet34 is pre-trained on the ImageNet dataset, which contains more than 100,000 images across 200 classes. A CNN is usually used for image classification purposes.

XI. CONCLUSIONS

The main goal of this research is to develop algorithms to accurately detect Alzheimers disease.The models attain an impressive accuracy rate of 80.4%, with the convolutional neural network outperforming other algorithms in terms of accuracy. We have combined the application of machine learning Algorithms to determine which of the Algorithms detect the disease accurately and timely. These models also overcome the overfitting problem of machine learning techniques to classify the disease accurately with less computational complexity. Based on this result, it becomes realistic that CNN model detects the disease early. Later, other advanced machine learning models may be implemented separately to further increase the classification accuracy.

XII. ACKNOWLEDGEMENTS

I would like to thank Dr Khem Poudel, the Computer Science and Computational and Data Science departments for providing the necessary resources, facilities, and research environment that facilitated the smooth progress of this study.

REFERENCES

[1] A. W. Salehi, P. Baglat, B. B. Sharma, G. Gupta, and A. Upadhya, "A cnn model: earlier diagnosis and classification of alzheimer disease using mri," *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2020, pp. 156–161.

[2] M. Tanveer, B. Richhariya, R. U. Khan, A. H. Rashid, P. Khanna, M. Prasad, and C. Lin, "Machine learning techniques for the diagnosis of alzheimer's disease: A review," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 1s, pp. 1–35, 2020.

[3] M. Chowdhury, C. Li, and K. Poudel, "Combining deep learning with traditional machine learning to improve phonocardiography classification accuracy," *2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE, 2021, pp. 1–5.

[4] M. Odusami, R. Maskeliūnas, R. Damaševičius, and S. Misra, "Explainable deep-learning-based diagnosis of alzheimer's disease using multimodal input fusion of pet and mri images," *Journal of Medical and Biological Engineering*, pp. 1–12, 2023.

[5] D. C. Rao, G. H. Charan, G. Mounika, K. K. Sri, P. N. Reddy, and S. L. Krishna, "Diagnosis of alzheimer's disease using machine learning algorithms," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 7, p. 1581–1584, 2022.

[6] R. Janghel and Y. Rathore, "Deep convolution neural network based system for early diagnosis of alzheimer's disease," *Irbm*, vol. 42, no. 4, pp. 258–267, 2021.

[7] A. Farooq, S. Anwar, M. Awais, and S. Rehman, "A deep cnn based multi-class classification of alzheimer's disease using mri," *2017 IEEE International Conference on Imaging systems and techniques (IST)*. IEEE, 2017, pp. 1–6.

[8] L. Auria and R. A. Moro, "Support vector machines (svm) as a technique for solvency analysis," 2008.

[9] Y.-D. Zhang, S. Wang, and Z. Dong, "Classification of alzheimer disease based on structural magnetic resonance imaging by kernel support vector machine decision tree," *Progress In Electromagnetics Research*, vol. 144, pp. 171–184, 2014.

[10] J. Neelaveni and M. G. Devasana, "Alzheimer disease prediction using machine learning algorithms," *2020 6th international conference on advanced computing and communication systems (ICACCS)*. IEEE, 2020, pp. 101–104.

[11] G. Mirzaei and H. Adeli, "Machine learning techniques for diagnosis of alzheimer disease, mild cognitive disorder, and other types of dementia," *Biomedical Signal Processing and Control*, vol. 72, p. 103293, 2022.

[12] R. Kadri, M. Tmar, and B. Bouaziz, "Alzheimer's disease prediction using efficientnet and fastai," *Knowledge Science, Engineering and Management: 14th International Conference, KSEM 2021, Tokyo, Japan, August 14–16, 2021, Proceedings, Part II 14*. Springer, 2021, pp. 452–463.

[13] C. R. Francis, U. Lokhande, P. K. Bamrah, and A. D'costa, "Alzheimer's disease prediction using fastai," *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 1*. Springer, 2021, pp. 765–775.

[14] S. P. Praveen, P. N. Srinivasu, J. Shafi, M. Wozniak, and M. F. Ijaz, "Resnet-32 and fastai for diagnoses of ductal carcinoma from 2d tissue slides," *Scientific Reports*, vol. 12, no. 1, p. 20804, 2022.

[15] A. Dixit, "Hybrid architecture for brain abnormality detection on brain mr scans via fastai," 2021.

[16] C. Neuner, R. Coras, I. Blümcke, A. Popp, S. M. Schlaffer, A. Wirries, M. Buchfelder, and S. Jabari, "A whole-slide image managing library based on fastai for deep learning in the context of histopathology: two use-cases explained," *Applied Sciences*, vol. 12, no. 1, p. 13, 2021.

[17] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[18] M. Shahin, F. F. Chen, A. Hosseinzadeh, and N. Zand, "Using machine learning and deep learning algorithms for downtime minimization in manufacturing systems: An early failure detection diagnostic service," 2023.

[19] S. López, M. Golmohammadi, I. Obeid, and J. Picone, "An Analysis of Two Common Reference Points for EEGs," *Proceedings of the IEEE Signal Processing in Medicine and Biology Symposium*, 2016, pp. 1–4.