A Comparative Analysis of Deep Learning Models for Alzheimer's Disease Diagnosis

Nipun Chandra, Ayush Kumar, Swasti Gautam, Dr. Ankush Jain, Rohit Kumar
Computer Science and Engineering
Netaji Subhas University of Technology
Delhi, India

***Abstract*—Alzheimer's is the most common cause of dementia, a general term for memory loss and other cognitive abilities serious enough to interfere with daily life. Alzheimer's disease accounts for 60-80% of dementia cases. It is not a typical aspect of growing older. Growing older is the biggest known risk factor, and most Alzheimer's patients are 65 years of age or older. If an individual under 65 has Alzheimer's disease(AD), it is referred to as younger-onset AD. Early-onset AD is another name for younger-onset Alzheimer's. AD can be in its early, medium, or late stages in people with younger onset. Deep learning, a subfield of Machine Learning (ML), has shown promise for non-invasive AD detection using neuroimaging data. This research compares the accuracy of multiple pre-trained deep learning models for AD diagnosis.**

Keywords—Alzheimer's Disease (AD), Deep Learning, Diagnosis, Machine Learning.

I. INTRODUCTION

Alzheimer’s disease (pronounced “alz-HAI-mirs”) is a neurological disorder that gradually impairs one's capacity for memory, thought, learning, and organization. It eventually affects a person’s ability to carry out basic daily activities. Alzheimer’s disease (AD) is the most common cause of dementia. The symptoms of Alzheimer’s worsen over time. Researchers believe the disease process may start 10 years or more before the first symptoms appear. AD most commonly affects people over the age of 65. Early diagnosis of AD is essential for the progress of more prevailing treatments. Machine learning (ML), a branch of artificial intelligence, employs a variety of probabilistic and optimization techniques that permits PCs to gain from vast and complex datasets.In recent years, researchers provided analysis on the works done using machine learning for Alzheimers[2]. In 2017, Litjens et al. [1] presented a review on deep learning methods for medical image analysis. It is mentioned that although deep learning models are considered as ‘black boxes’, some statistical techniques can be used to estimate uncertainty of the network. Shen et al. [3] performed a survey on deep learning for Alzheimers. It also supported this fact of uncertainty in prediction by deep learning models. In 2018, Jose et al. provided a review on neuroimaging techniques for brain disorders. It is stated that machine learning techniques can be useful for finding the underlying neurological causes of brain disorders [4]. Pellegrini et al. [5] discussed the machine learning techniques used for dementia and cognitive impairment from 2006-2016 in 111 papers. It stressed on the development of novel machine learning models from an interdisciplinary approach. Rathore et al. [6] also provided a review on feature extraction and classification of Alzheimers and its prodromal stages. Use the enter key to start a new paragraph. The appropriate spacing and indent are automatically applied.

II. DATASET

The dataset for this research paper is one of the publicly available Alzheimer's MRI Preprocessed Dataset on Kaggle [7].

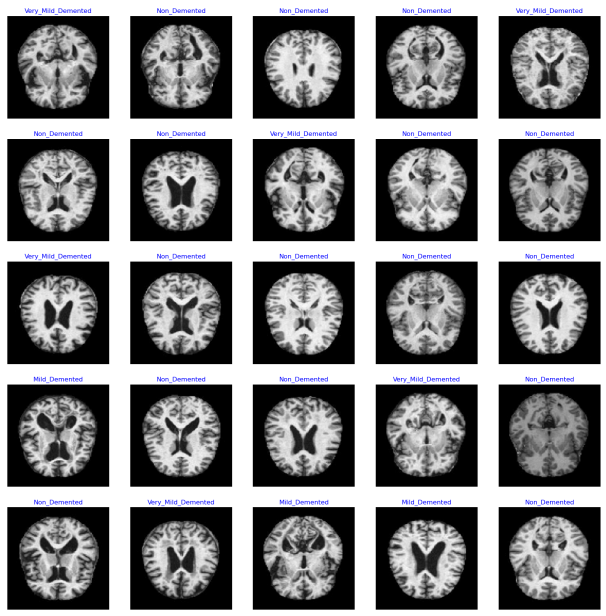


Figure 1: Glimpse of the MRI images from the dataset.

This dataset consists of 6400 pre-processed Magnetic Resonance Imaging (MRI) scans categorized into four classes: Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. Each MRI image is cropped to 128x128 pixels.

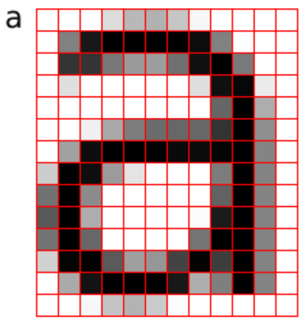
III. ANALOGY

A) Neural Networks:

Heart of deep learning algorithms, Neural networks, which are a subset of machine learning. They are made up of node layers, which have an output layer, an input layer, and one or more hidden levels. Every node has a threshold and weight that are connected to one another. A node is activated and sends data to the following layer of the network if its output exceeds a given threshold value. If not, no data is transferred to the network's subsequent tier. Neural networks come in many varieties and are applied to diverse use cases and kinds of data. For example, recurrent neural networks are commonly used for natural language processing and speech recognition whereas convolutional neural networks (ConvNets or CNNs) are more often utilized for classification and computer vision tasks. Identifying objects in images required laborious, manual feature extraction techniques before CNNs were developed. Convolutional neural networks, on the other hand, now offer a more scalable method for classifying images and recognizing objects in images by using the concepts of matrix multiplication and linear algebra to find patterns in images. That said, they can be computationally demanding, requiring graphical processing units (GPUs) to train models. [8]

B) Convolutional Neural Networks:

Convolutional neural networks, or CNNs or ConvNets, are a subclass of neural networks that are very adept at processing data with a topology resembling a grid, like images.. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be. [9]



1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Figure 2: Representation of image as a grid of pixels

The human brain processes a huge amount of information the second we see an image. Every neuron functions within its own receptive area and is interconnected with other neurons to include the whole visual field. Each neuron in a CNN processes data only in its receptive field, just as each neuron in the biological vision system responds to stimuli only in the limited area of the visual field known as the receptive field. The layers are set up to identify more complicated patterns (faces, objects, etc.) later on and simpler patterns (lines, curves, etc.) earlier.

1) Convolutional Neural Network Architecture:

Convolutional neural networks function better with picture, speech, or audio signal inputs than other types of neural networks. There are three primary categories of layers in them::

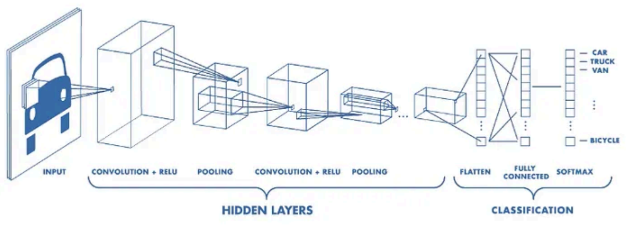


Figure 3: Architecture of a CNN

2) Convolution layer:

A convolutional layer is the main building block of a CNN. It has a number of filters, also known as kernels, the parameters of which must be learned during the training process. Generally speaking, the filters' size is less than that of the original image. If we have an input of size W x W x D and D*out* number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula[9]:

$$W_{out} = \frac{W-F+2P}{S} + 1 \qquad (1)$$

This will yield an output volume of size W*out* x W*out* x D*out*.

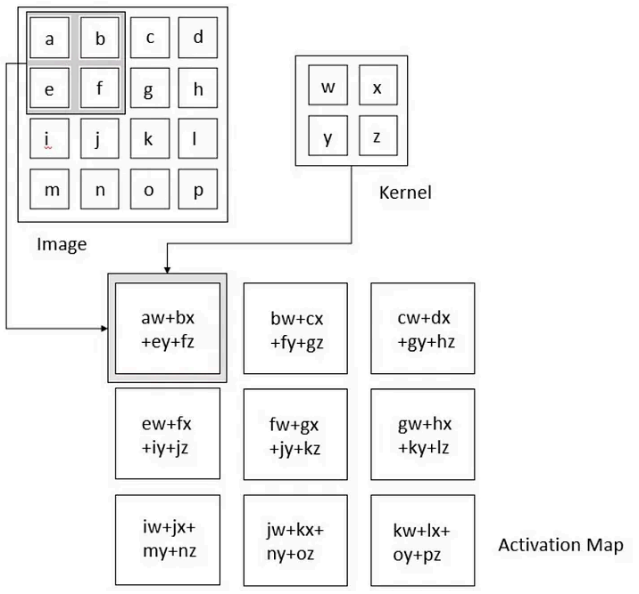


Figure 4: Convolution Operation (Source: Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville)

3) Pooling Layer:

The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This aids in shrinking the representation's spatial size, which lowers the quantity of computation and weights needed. Each slice of the representation is processed independently for the pooling operation. The L2 norm of the rectangular neighborhood, the mean of the rectangular neighborhood, and the weighted average based on the distance from the central pixel are some examples of pooling functions. Nonetheless, the most widely used method is max pooling, which provides the neighborhood's maximum output.[9]

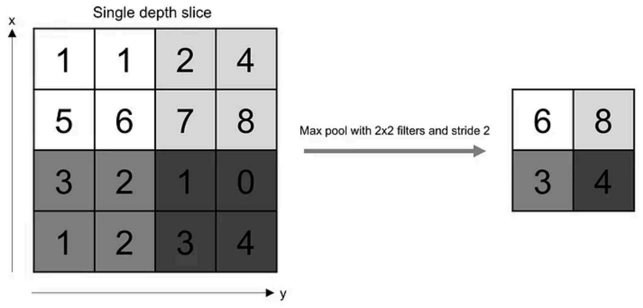


Figure 5: Pooling Operation (Source: O'Reilly Media)

If we have an activation map of size W x W x D, a pooling kernel of spatial size F and stride S, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W-F}{S} + 1 \tag{2}$$

The result will be an output volume that is W*out* x W*out* x D*in* size.

4) *Fully-connected layer:*

As previously stated, with partially linked layers, the pixel values of the input image are not directly connected to the output layer. Every output layer node in the fully-connected layer, on the other hand, is directly connected to every other layer node. Using the features that were retrieved from the earlier layers and their various filters, this layer classifies the data. FC layers typically utilize a softmax activation function to categorize inputs adequately, producing a probability from *0 to 1*, whereas convolutional and pooling layers typically use ReLu functions.

5) *Types of Convolutional Neural Networks:*

i) VGG-16:

The University of Oxford's Karen Simonyan and Andrew Zisserman proposed the VGGNet convolutional neural network design in 2014. The primary topic of this research is how the accuracy of a convolutional neural network is affected by its depth. The ***224 x 224*** RGB image is the input for the VGG based convNet. The preprocessing layer subtracts the mean image values, which are determined over the whole ImageNet training set, from an RGB image having pixel values between 0 and 255.

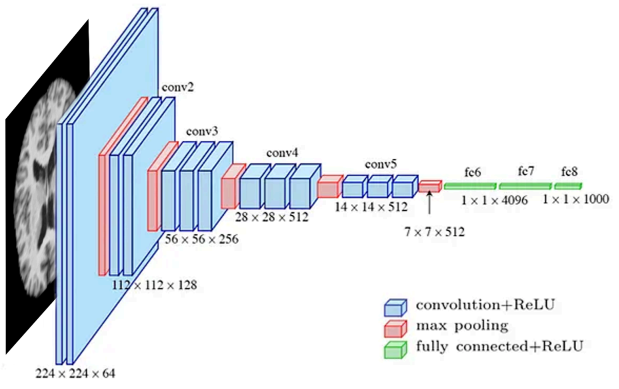


Fig 6: A visualization of the VGG architecture[12]

A convolutional neural network (CNN) architecture called the VGG-16 model was put forth by the University of Oxford's Visual Geometry Group (VGG). With 16 layers total—13 convolutional layers and 3 fully linked layers—it is distinguished by its depth. VGG-16 is well known for its efficiency and simplicity, as well as for its ability to perform well on a variety of computer vision tasks, such as object recognition and image categorization. The design of the model consists of a stack of progressively deeper max-pooling layers after a series of convolutional layers. What would happen if we wanted to feed the model a batch of m training examples?

$$X = [x^{(1)}, x^{(2)}, \ldots x^{(m)}] \tag{3}$$

Hence, we can consider X to be a tensor of dimensions (m,224,224,3), where m is the batch's total number of examples. The training example within the batch is indicated by the superscript. Thus, the first training example in the batch would be x(1). Likewise, regarding output:

$$Y = [y^{(1)}, y^{(2)}, \ldots y^{(m)}] \tag{4}$$

ii) Inception v3:

Inception V3 is a deep convolutional neural network model that has been widely recognized for its performance in various computer vision tasks such as image classification, object detection, and semantic segmentation. Originally intended for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), it was a version of the Inception architecture expanded. Subsequently, it has been employed in other medical image analysis uses, including Alzheimer's disease categorization.It uses Factorized 7 x 7 convolutions, Label Smoothing, and an auxiliary classifier to transport label information lower down the network (along with batch normalization for layers in the side head) among other enhancements.Inception v3 network's architecture is constructed step-by-step, as follows:

1) *Factorized convolutions:* As *Gilbert et al.[15]* stated by lowering the number of network parameters, this technique lowers computing efficiency. It also monitors the effectiveness of the network.
2) *Smaller convolutions:* Training proceeds more quickly when smaller convolutions are used in place of larger convolutions. If two 3 × 3 filters are used in place of a 5 × 5 convolution, the number of parameters reduced to 18 (3*3 + 3*3) from the 25 in the case of a 5 x 5 filter..
3) *Asymmetric convolutions:* One possible substitution for a 3 × 3 convolution would be a 1 × 3 convolution, then a 3 × 1 convolution. A 2 x 2 convolution would require a few more parameters than the suggested asymmetric convolution in place of a 3 × 3 convolution.
4) *Auxiliary classifier:* During training, an auxiliary classifier is a tiny CNN that is injected between layers; the loss it incurs is contributed to the main network loss. Auxiliary classifiers were utilized for a deeper network in GoogLeNet, however in Inception v3, they serve as a regularizer.
5) *Grid size reduction*: Pooling techniques are typically used to reduce grid size. However, a more effective method is suggested to overcome the computational cost bottlenecks.[15]

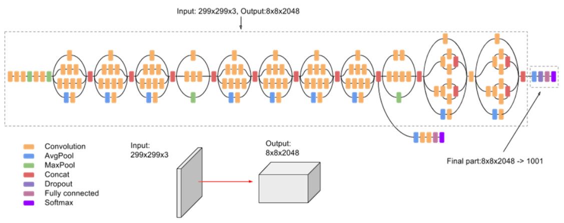


Fig 7: Main architecture of Inception v3

iii) Efficientnet B3:

Google Research has developed a series of convolutional neural network topologies called EfficientNet. One of the models in the EfficientNet family, which is thought to be efficient and well-balanced, is EfficientNet-B3. Using a compound scaling strategy to automatically scale up the model's architecture, in terms of depth (number of layers) and width (number of filters per layer), dependent on the input image resolution, is one of the key contributions of EfficientNet-B3. With this paradigm, we can process both large and tiny photos with greater efficiency and better outcomes by using more resources. The third model in the EfficientNet family, the EfficientNet-B3, has 1.2 times the resolution and 1.3 times the width of the EfficientNet-B0. Its depth is the same. In [16], Efficient-Net was initially presented. EfficientNet makes use of a state-of-the-art CNN model scaling method. To obtain efficient results, simple compound coefficients are employed. Unlike traditional techniques, which expand a network's depth, width, and resolution, EfficientNet uniformly grows in every dimension. Model performance improves with scaling of each size, but performance is much improved when network characteristics are balanced with respect to resource availability. The authors [16] systematically raised the image's resolution, width, and coefficient of depth as a result of the equations.

$$\text{Depth: } d = \alpha\varphi \tag{5}$$

$$\text{Width: } w = \beta\varphi \tag{6}$$

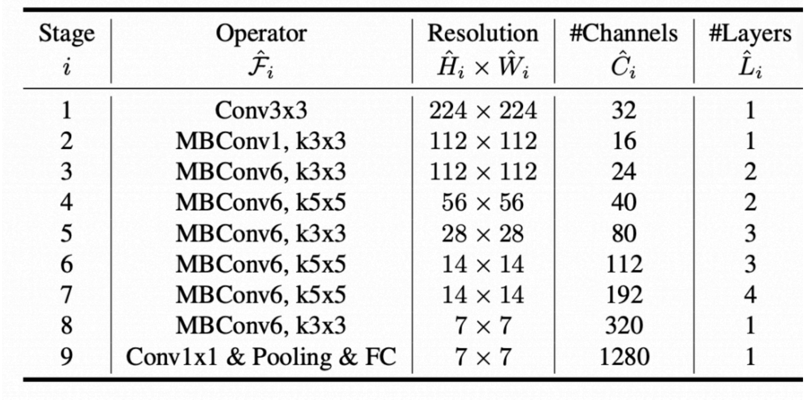
$$\text{Resolution: } r = \gamma\varphi \tag{7}$$

$$\text{s.t.}\alpha \cdot \beta 2 \cdot \gamma 2 \approx 2 \qquad \alpha \geq 1,\ \beta \geq 1,\ \gamma \geq 1 \tag{8}$$

The availability to new resources needed to scale the model is controlled by the coefficient ϕ. These constants, which specify how the additional resources are distributed to the depth, width, and resolution, can be found with an initial grid α, β, and γ search.
There are 11184179 parameters in the model overall. The other parameters, 90375, are not trained; only 11093804 is capable of training.

- The convolutional neural network EfficientNet-B3 (Functional) has pre-trained weights and an output shape of (None, 1536) with 10783535 parameters.
- The global average pooling 2D 5 pooling layer averages each feature map over the input's spatial dimensions to produce the result (None, 1536).
- Dropout randomly drops neurons at a rate of 0.45 during training to minimize overfitting.
- Dense is a final output layer with 1028 parameters that is a fully connected layer made up of four units.

In order to decrease over-fitting, the EfficientNet-B3 design was expanded by including a Global Average Pooling 2D Flatten and drop rate of.45, as well as a Soft-max layer with two classes, 0 and 1. The architecture uses the Adamax optimizer for quicker network optimization, the learning rate is.001, and it was optimized for 6400 sample images in 40 epochs.
The loss function of the model in question is the Mean Squared Error. In regression situations, where the objective is to minimize the difference between the expected and actual outputs, mean square error (MSE) is a frequently used loss function. Adam serves as the model's optimizer. Adam is an adaptive learning rate optimization approach that updates the model's parameters based on the gradient of the loss function. Adaptive learning rates are calculated for every parameter. Generally speaking, when the data is noisy, Adam can be a decent option. With a batch size of 40, the gradients will be computed for 40 training examples prior to the model's parameters being updated. This significant hyper-parameter has the potential to impact the model's performance and speed. Greater batch sizes may result in quicker training, but they may also require more memory. The model will undergo 40 iterations of training on the complete dataset due to the 40 epochs. The training will terminate after twenty epochs, or iterations. It's important to remember that experimenting with different values for these hyper-parameters may be necessary to get the best results, depending on the complexity of the problem and the model's performance.



Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Table 1: Kernel Size, resolution, channels, and no. of layers information.

iv) Xception: Xception (Extreme Inception) is another convolutional neural network (CNN) architecture developed by Google Research. It was suggested that Xception be used instead of more conventional Inception-style architectures since it would be more accurate and efficient. Xception also employs skip connections to allow for better gradient flow and improved accuracy. The architecture has achieved state-of-the-art results on various image classification benchmarks such as ImageNet, and it has been widely used in computer vision applications [17]. The following are the main features and benefits of the Xception model [18]:

1. Parameter Efficiency: Depthwise separable convolutions have a much lower parameter count than conventional convolutions, which lightens the model and improves parameter usage efficiency.
2. Performance: In a range of visual recognition tasks, the Xception model beats other sophisticated models of the era, including the original Inception model.
3. Adaptability: The Xception model is better suited for deployment on devices with constrained processing resources because it has fewer parameters.

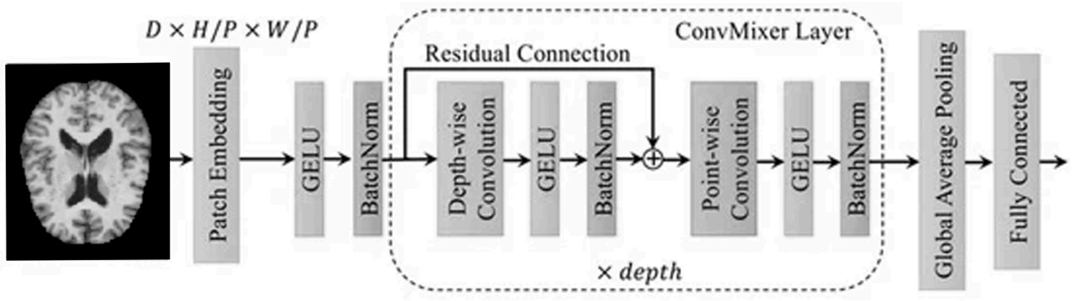


Fig: 9 Xception Network

1) *Standard Convolution:*

$$y_{i,j,k} = \sum_{l,m,n} x_{i+l,j+m,n} \cdot w_{l,m,n.k} + b_k \tag{9}$$

where:

- $y_{i,j,k}$ is the output value at a specific location (i, j) and filter k.
- $x_{i+l,j+m,n}$ is the input value at a specific location (i + l, j + m) and channel n.
- $w_{l,m,n.k}$ is the weight of the convolution filter.
- b_k is the bias for filter k.

2) *Depth-wise Separable Convolution:*

Xception uses depth-wise separable convolutions instead of standard convolutions. This convolution can be decomposed into two steps:

$$y_{i,j,c} = \sum_{d,e} x_{i+d,j+e,c} \cdot w_{d,e,c} \tag{10}$$

This operation applies a single filter per input channel, unlike the standard convolution, which uses a filter across all channels.

3) *Point-wise Separable Convolution:*

$$y_{i,j,k} = \sum_{c} y_{i,j,c} + w_{c,k} \quad (11)$$

The output from the depthwise convolution is combined across all channels with a 1x1 convolution, reducing the number of channels and combining the features.

IV) THE PROPOSED METHODOLOGY

Convolutional neural networks (ConvNets) play a pivotal role in analyzing medical imaging data. Inspired by seminal work such as Krizhevsky et al. (2012), ConvNet training adheres to a regimen of mini-batch gradient descent with momentum, utilizing multinomial logistic regression objectives[24]. Regularization techniques like weight decay and dropout are employed, with specific configurations to counteract overfitting. The training process involves careful initialization of network weights, leveraging pre-training for deeper architectures to mitigate the challenges of gradient instability. To enhance model robustness, multi-scale training strategies are adopted, allowing networks to learn from images rescaled within a range, thereby accommodating variations in object sizes. Additionally, attention mechanisms are integrated into the ConvNet architecture to emphasize salient features in the data, potentially improving diagnostic accuracy.

Efforts to optimize ConvNet performance for Alzheimer's detection include selecting appropriate model variants like EfficientNet-B3, balancing computational constraints with accuracy requirements. Training protocols incorporate distributed machine learning systems and techniques like RMSProp optimization and gradient clipping to stabilize training and accelerate convergence. These methodologies, combined with meticulous hyperparameter tuning, contribute to the development of robust ConvNet models capable of detecting Alzheimer's-related patterns in medical imaging data. Moreover, considerations such as weight decay rates and the absence of dropout in large datasets are tailored to suit the complexities of Alzheimer's diagnosis tasks. By integrating these advancements into ConvNet architectures, researchers aim to enhance the accuracy and efficiency of Alzheimer's detection methods, potentially aiding in early diagnosis and intervention strategies.

A CNN model typically consists of multiple convolutional layers that work sequentially on an image. In order to assist the model approximate nonlinear functions, the convolutional layers are entwined with other layer types such as pooling layers, normalization layers, and activation function layers. However, as these model specifics are unrelated to the topic at hand, we will disregard them as we have discussed them in the above sections. They utilize complex patterns and features retrieved from brain pictures, which are essential for the detection and classification of Alzheimer's disease. Observe how a tiny portion of the image's attributes are captured by the neurons in the first convolutional layer. This area will be 3 × 3 in size if the filters used are of that size.
Look at Fig 10. This region of the image is referred to as the neuron's receptive field. Each neuron in the CNN's subsequent layer is convolved with the same 3x3 region as in the layer before it, but this results in a larger receptive field in the input image. Each neuron represents an ever-larger receptive field as we move deeper and deeper into the network.
The receptive field in the scan is represented by the learned feature vector that runs along the third dimension of the neurons. As a result, the dashed green square in Figure 9's receptive field may be represented by the cyan feature vector of the last convolutional layer. Similarly, the additional feature vectors in Fig. 10 (purple, magenta, and green) correspond to distinct receptive fields within the image. Various areas of the image are represented by these various receptive fields. As a result, we may use a weighted mixture of the feature vectors to pay attention at this level. By teaching the model the relative weights we apply to the appropriate feature vectors, we may train it to focus on significant areas of the image. The final characteristics generated from the CNN model are combined into a single vector using two methods. The feature vectors can be pooled, or they can be added, multiplied, averaged, or subjected to other operations. Alternatively, the feature vectors can be flattened, or layered one on top of the other (see Fig. 10a). For instance, Fig. 10b employs Global Average Pooling (GAP), which is discovered to be more reliable in real-world scenarios.
Nevertheless, GAP processes the input image by calculating a basic average with equal weights, without paying any particular attention to any specific receptive fields or regions.Thus, the GAP operation performs the operation shown in the equation below:

$$L_j = GAP(L_{j-1}) = \frac{1}{MXM}\sum_{i=0}^{MXM} F_i \quad (12)$$

where L_j and L_{j-1} are the input and output layers of the GAP function. F_i is the feature vectors contained in layer L_j, and MXM is the number of feature vectors (2x2 in Fig.10). In order to implement attention, we need to compute a weighted average as follows:

$$L_j = GAPATN(L_{j-1}) = \sum_{i=1}^{MXM} w_i F_i \quad (13)$$

where w_i are weights to be learned by the model automatically.

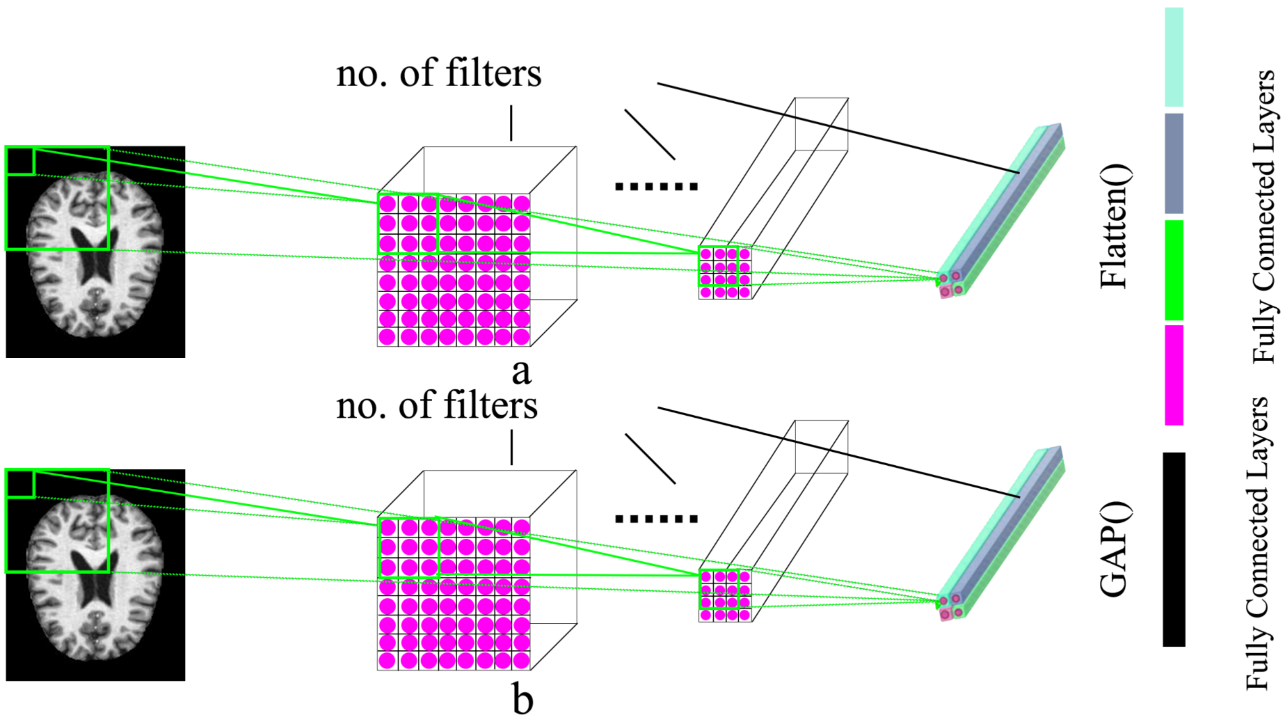
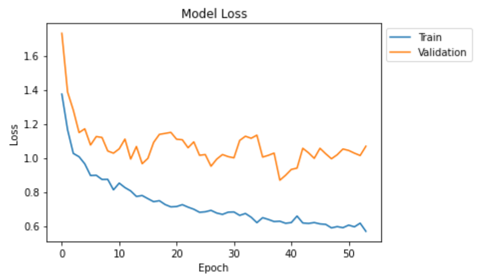
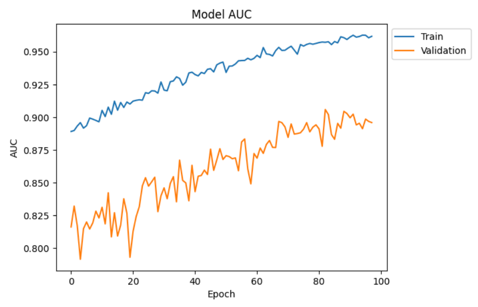


Fig:10 Typical CNN architecture (a) Last feature vectors are flattened into one large vector, (b) Last feature vectors are averaged.

V) RESULT

In our work, the models were trained with a graphics processing unit and Google Colab, and the produced system processed and classified data using the Python programming language. We carried out three trials with three configurations in order to explicitly determine whether the first or the second model has a significant improvement. The first two are used to determine the best way to divide data into different ratios and classify AD. 20% of the data was used for testing, and 80% for training.





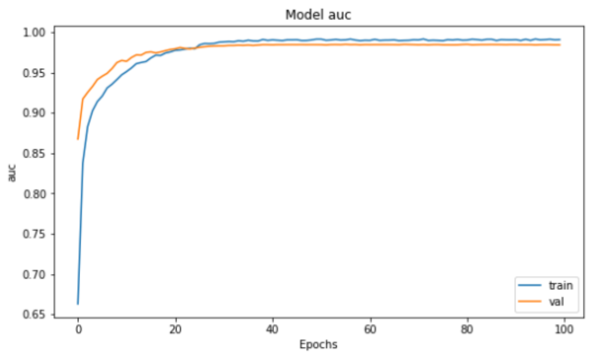
The above mentioned curves are Model Loss andModel AUC curves for VGG-16 . Specifically, the Area Under the Curve (AUC) curve offers valuable insights into the model's ability to discern between different stages of Alzheimer's disease or distinguish individuals with Alzheimer's from those without the condition. This metric serves as a vital indicator of the model's discriminatory power and its overall classification accuracy. Meanwhile, the model loss curve tracks the progression of the model's training process, providing a visual representation of how effectively the model minimizes errors and adjusts its parameters to optimize performance in Alzheimer's classification tasks. and the performance metrics for VGG-16 are:

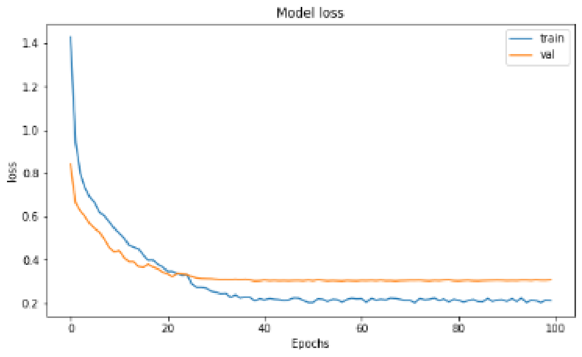
Accuracy	0.838
Precision	0.681
Recall	0.664
AUC	0.898

Table 2: Performance matrix for VGG-16

Precision quantifies the proportion of true positive predictions among all positive predictions made by the model, thus indicating its ability to avoid false positives. Recall, on the other hand, measures the proportion of true positive predictions identified by the model among all actual positive instances in the dataset, illustrating its capacity to detect all relevant cases. Additionally, the Area Under the Curve (AUC) metric provides a comprehensive measure of the model's performance across various threshold settings, offering a holistic view

of its ability to discriminate between Alzheimer's patients and non-patients.
The curves below represent the performance evaluation of the Inception Net V3 model..



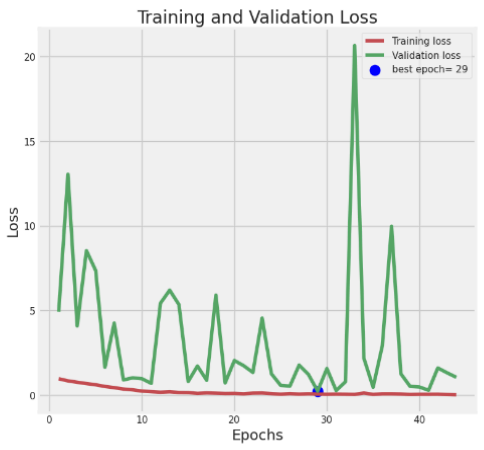


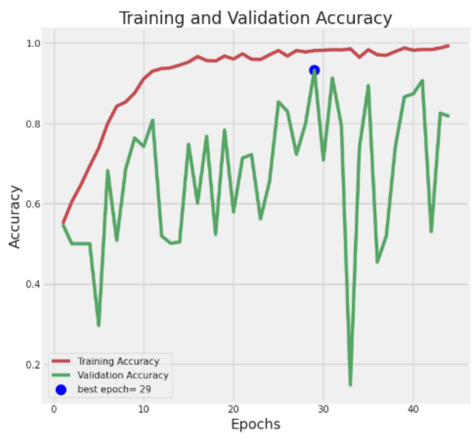
The performance metrics for Inception Net V3 are:

Accuracy	0.896
Precision	0.890
Recall	0.890
AUC	0.985

Table 3: Performance matrix for Inception Net V3

In the case of Xception, the curves for loss and accuracy are:



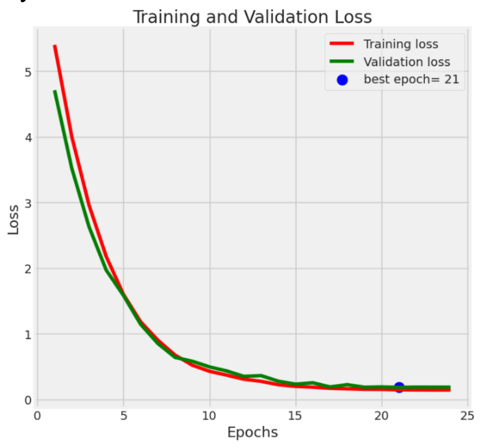


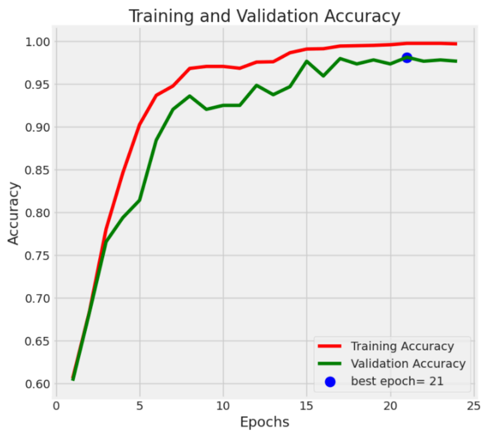
The performance metrics for Xception are:

Accuracy	0.925
Precision	0.93
Recall	0.93
f1-score	0.93

Table 4: Performance matrix for Xception

Lastly, Efficient Net B3, the curves for model's loss and accuracy are:



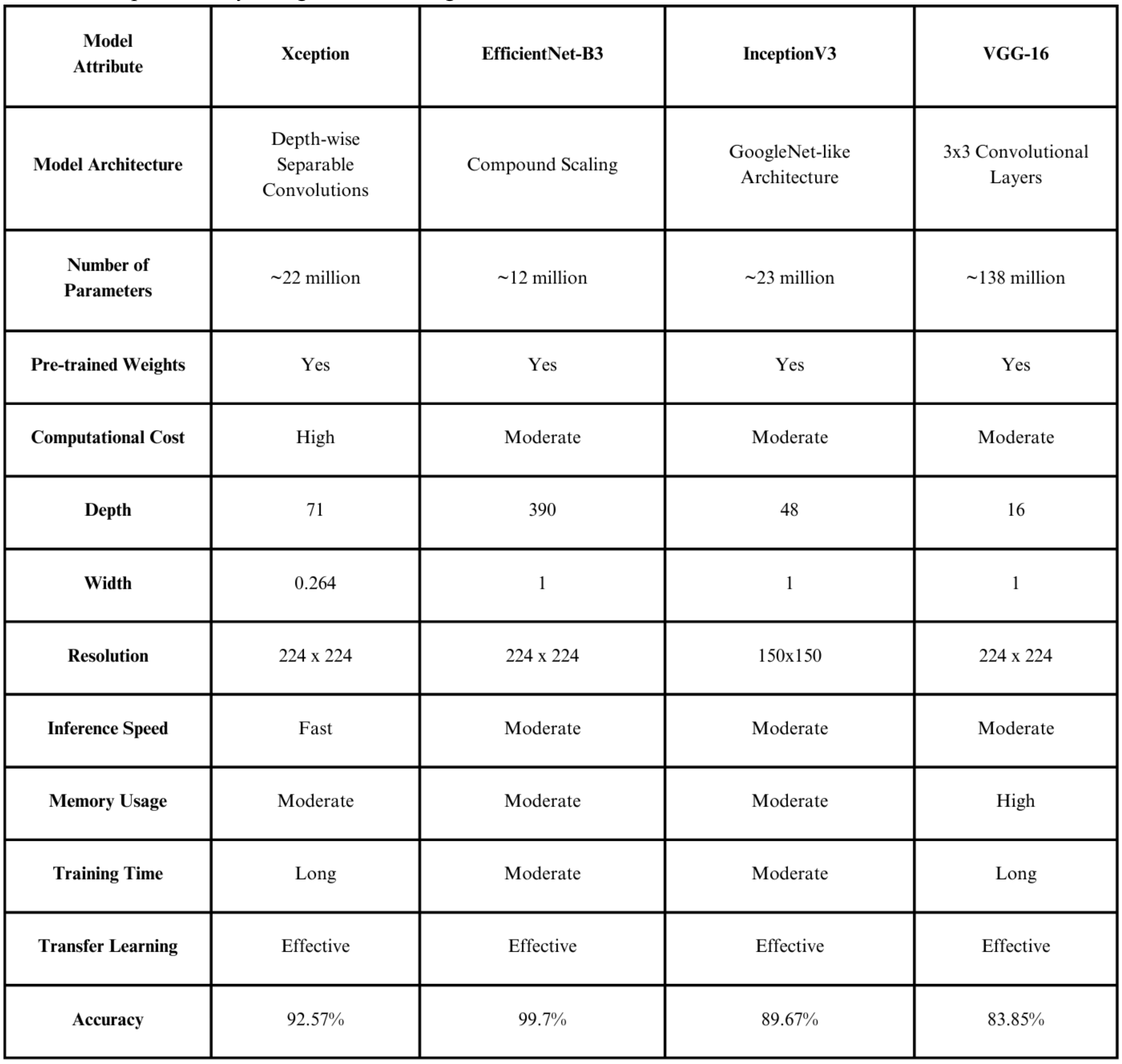


The performance metrics for Efficient Net B3 are as follows:

Accuracy	0.997
Precision	0.99
Recall	0.99
f1-Score	0.99

Table 5: Performance Matrix for Efficient Net B3

Below is the comparison of these models, implemented using the T4-GPU provided by Google Colab. Google Colab, a hosted Jupyter Notebook service, offers a user-friendly environment for running code and accessing computing resources. The T4-GPU, renowned for its high performance, enables efficient execution of deep learning tasks, including those involving complex models like Inception Net V3 for Alzheimer's classification. Leveraging this powerful combination of hardware and platform, researchers and developers can conduct experiments and analyses with ease, benefiting from the convenience and accessibility afforded by Google Colab's free-tier resources.



Model Attribute	Xception	EfficientNet-B3	InceptionV3	VGG-16
Model Architecture	Depth-wise Separable Convolutions	Compound Scaling	GoogleNet-like Architecture	3x3 Convolutional Layers
Number of Parameters	~22 million	~12 million	~23 million	~138 million
Pre-trained Weights	Yes	Yes	Yes	Yes
Computational Cost	High	Moderate	Moderate	Moderate
Depth	71	390	48	16
Width	0.264	1	1	1
Resolution	224 x 224	224 x 224	150x150	224 x 224
Inference Speed	Fast	Moderate	Moderate	Moderate
Memory Usage	Moderate	Moderate	Moderate	High
Training Time	Long	Moderate	Moderate	Long
Transfer Learning	Effective	Effective	Effective	Effective
Accuracy	92.57%	99.7%	89.67%	83.85%

Fig 11 Comparison of all Models

VI) CONCLUSION

Our paper focused on Alzheimer's disease (AD) detection using deep learning techniques on neuroimaging data. We explored various convolutional neural networks (CNNs) such as VGG-16, InceptionNet, EfficientNet B3 and Xception to accurately classify brain MRI images across different stages of AD progression.

Among these models, Xception stood out for its high accuracy and fast inference speed, with moderate computational cost and memory usage due to its depth-wise separable convolutions. EfficientNet B3 also showed exceptional accuracy through transfer learning with reasonable computational overhead, making it promising for real-world applications. Inception V3 and VGG-16 demonstrated competitive performance with varying computational requirements.

Our study contributes to advancing early detection and treatment of AD using AI and machine learning. By leveraging deep learning and neuroimaging data, we aim to improve the quality of life for AD patients and caregivers.

VII) CHALLENGES AND FUTURE SCOPE

Challenges in AD detection using deep learning include limited availability of high-quality neuroimaging datasets, data imbalance, lack of model interpretability, generalizability to clinical settings, ethical and legal considerations, computing resource requirements, and validation for clinical use.

Future directions in Alzheimer's detection and treatment include exploring advanced model architectures tailored for this purpose and leveraging deep learning for drug discovery. This involves virtual screening, drug repurposing, molecular docking, and prediction of binding affinity. Additionally, integrating multimodal neuroimaging data alongside clinical and genetic information holds promise for enhancing diagnostic accuracy and predictive modeling.

VIII) REFERENCES

[1] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. 2017. A survey on deep learning in medical image analysis. Medical image analysis 42 (2017), 60–88.

[2] Cleveland Clinic. (2023, December). Alzheimer's Disease: Causes, Symptoms, Treatment & Stages. https://my.clevelandclinic.org/services/alzheimers-disease-treatmen t

[3] Dinggang Shen, Guorong Wu, and Heung-Il Suk. 2017. Deep learning in medical image analysis. Annual review of biomedical engineering 19 (2017), 221–248.

[4] José María Mateos-Pérez, Mahsa Dadar, María Lacalle-Aurioles, Yasser Iturria-Medina, Yashar Zeighami, and Alan C Evans. 2018. Structural neuroimaging as clinical predictor: A review of machine learning applications. NeuroImage: Clinical (2018).

[5] Enrico Pellegrini, Lucia Ballerini, Maria Del C Valdes Hernandez, Francesca M Chappell, Victor González-Castro, Devasuda Anblagan, Samuel Danso, Susana Muñoz-Maniega, Dominic Job, Cyril Pernet, et al. 2018. Machine learning of neuroimaging for assisted diagnosis of cognitive impairment and dementia: A systematic review. Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring 10 (2018), 519–535.

[6] Saima Rathore, Mohamad Habes, Muhammad Aksam Iftikhar, Amanda Shacklett, and Christos Davatzikos. 2017. A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer's disease and its prodromal stages. NeuroImage 155 (2017), 530–548.

[7] SachinKumar413. (2022, August 25). Alzheimer's MRI Preprocessed Dataset. Kaggle. https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-d ataset

[8] International Business Machines Corporation (IBM). (n.d.). Convolutional neural networks. [IBM]. Retrieved April 11, 2024, from https://www.ibm.com/topics/convolutional-neural-networks

[9] Mayank Mishra. (n.d.). Convolutional neural networks explained [Towards Data Science]. Retrieved April 11, 2024, from https://towardsdatascience.com/understanding-cnn-convolutional-n eural-network-69fd626ee7d4

[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[11] Li, F.-F., Johnson, J., & Yeung, S. (n.d.). CS231n: Convolutional neural network for visual recognition [Course]. Stanford University. https://cs231n.stanford.edu/

[12] Davi Frossard, VGG in TensorFlow, 2016

[13] Frossard, D. (2016, June 17). VGG in TensorFlow. Retrieved from https://davisr.github.io/vgg16/

[14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

[15] Gilbert, A. (2019, March 15). Popular deep learning architectures: ResNet, InceptionV3, SqueezeNet. Paperspace Blog. https://blog.paperspace.com/popular-deep-learning-architectures-re snet-inceptionv3-squeezenet/

[16] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Mach. Learn., 2019, pp. 6105–6114

[17] Muhammad Mujahid, 'An Efficient Ensemble Approach for Alzheimer's Disease Detection Using an Adaptive Synthetic Technique and Deep Learning'

[18] Shaojie Li, Haichen Qu, Xinqi Dong, 'Leveraging Deep Learning and Xception Architecture for High-Accuracy MRI Classification in Alzheimer's Diagnosis', 2024

[19] Cummings, J. L., Leisgang Osse, A. M., & Kinney, J. W. (2023). Alzheimer's Disease: Novel Targets and Investigational Drugs for Disease Modification.

[20] Alhazmi, H. A., & Albratty, M. (2022). An update on the novel and approved drugs for Alzheimer disease. https://doi.org/10.1016/j.jsps.2022.10.004

[21] Pan, X., Yun, J., Coban Akdemir, Z. H., Jiang, X., Wu, E., Huang, J. H., Sahni, N., & Yi, S. S. (2023). AI-DrugNet: A network-based deep learning model for drug repurposing and combination therapy in neurological disorders. https://doi.org/10.1016/j.csbj.2023.02.004

[22] Arrué, L., Cigna-Méndez, A., Barbosa, T., Borrego-Muñoz, P., Struve-Villalobos, S., Oviedo, V., Martínez-García, C., Sepúlveda-Lara, A., Millán, N., Márquez Montesinos, J. C. E., Muñoz, J., Santana, P. A., Peña-Varas, C., Barreto, G. E., González, J., & Ramírez, D. (2023). New Drug Design Avenues Targeting Alzheimer's Disease by Pharmacoinformatics-Aided Tools. doi: 10.3390/pharmaceutics14091914

[23] A. Batool and Y. -C. Byun, "Lightweight EfficientNetB3 Model Based on Depthwise Separable Convolutions for Enhancing Classification of Leukemia White Blood Cell Images," in IEEE Access, vol. 11, pp. 37203-37215, 2023, doi: 10.1109/ACCESS.2023.3266511. keywords: {Feature extraction;Computer architecture;Microprocessors;White blood cells;Cancer;Bones;Image segmentation;Acute lymphoblastic leukemia (ALL) ;efficient net-B3;CNN;white blood cell image classification;deep learning}

[24] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.