IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Comparative Analysis of Convolutional Neural Network Models for Autism Spectrum Disorder Detection.

Pramod Naik [1], Deepthi S [2], Monisha S [3], Meghana G N [4] and Likitha H M [5]

Dayananda Sagar University, Bengaluru, India – 562 112

Corresponding author:
1)pramodnaik-cse@dsu.edu.in
2)eng20cs0088@dsu.edu.in
3)eng20cs0210@dsu.edu.in
4)eng20cs0195@dsu.edu.in
5)eng20cs0174@dsu.edu.in

Abstract. Autism Spectrum Disorder (ASD) diagnosis presents significant challenges, often relying on clinical observation and expert evaluation. Leveraging Convolutional Neural Networks (CNNs), holds promise for automating ASD detection. The study focuses on leveraging a pre-trained VGG-16 model, DenseNet model, NASNet, Alexnet, Inception net, Exception net, Resnet, Efficient net CNN models, initially trained on a diverse image dataset and fine-tuning it to recognize ASD-related patterns from images. Utilizing transfer learning, the network's early layers capture general features, while deeper layers adapt to learn ASD-specific patterns from the image dataset. In parallel, a feature dataset is constructed, extracting pertinent image features using techniques like convolutional neural network activations and texture analysis.

This study scrutinizes the efficacy of various CNN architectures—VGG-16, NASNet, DenseNet, InceptionNet, EfficientNet, ResNet, AlexNet, and XceptionNet—for ASD detection. The study dwelve into conducting experiments on Autism image dataset, and rigorously test the performance metrics. The proposed EfficientNet model achieved the highest accuracy, 98%, compared to others models.

Keywords: Convolutional Neural Network, VGG-16, Nas net, Dense Net, Inception Net, Efficient Net, ResNet, AlexNet, Xception Net.

INTRODUCTION

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental disorder characterized by social communication deficits and repetitive behaviors.Typically, seen in young children between the ages of 2 to 5[1].This disorder is due both neurological and genetic factors that affect the child's development.Social interaction,reasoning, visualization, repetive behaviours, and problems in expression are all indicators of ASD[4]. Early diagnosis is pivotal for timely intervention. Deep learning, specifically Convolutional Neural Networks (CNNs), offers a potential solution by analyzing biomedical data for ASD detection. This study assesses the efficacy of various CNN architectures in automating ASD diagnosis, aiming to contribute to early intervention efforts by specifically employing the Visual Geometry Group (VGG) model, to pioneer a system capable of detection and diagnosis of Autism Spectrum Disorder (ASD) through visual cues, by analysing facial expressions in children with ASD. Similarly, this study also employs the same methodology with others CNN models like the DenseNet, NASNet, EfficientNet, XceptionNet, AlexNet, ResNet, InceptionNet. By assembling a diverse and well labelled dataset encompassing both ASD-affected individuals and neurotypical counterparts, the study aims to train the various CNN models to discern intricate visual patterns associated with ASD.

STATE OF RELATED WORK

Detecting Autism Spectrum Disorder (ASD) using Convolutional Neural Network (CNN) models has been an area of active research in recent years. CNNs have shown promising results in various computer vision tasks, including medical image analysis. Thus, this has led to a shift and improvement in ASD diagnostic methods, fulfilling most clinical diagnostic requirements. However, ASD discovery remains difficult. AI technologies have the potential to revolutionise autism detection by classifying images of children and discerning subtle distinctions between autistic and non-autistic individuals. The paper explores using facial features as a potential biomarker for predicting Autism Spectrum Disorder (ASD) through a deep learning Convolutional Neural Network (CNN) algorithm. It emphasizes the importance of early detection and intervention for individuals with ASD and discusses the potential of CNNs in accurately predicting ASD based on facial images. The study uses a publicly available dataset, applies preprocessing techniques to enhance data quality, and achieves high accuracy, sensitivity, and specificity in ASD prediction.[10]

Here is a literature review summarizing the state of the art and related work in ASD detection using various CNN models:

ResNet, with its deep architecture, has been utilized in ASD detection tasks to capture intricate patterns from medical imaging data.[5] AlexNet, known for its pioneering role in deep learning, has been adapted for ASD detection tasks. [1] InceptionNet's inception modules have been leveraged for feature extraction in ASD detection tasks.[3] XceptionNet, known for its depth-wise separable convolutions, has shown potential in ASD detection tasks. [4]

DenseNet's densely connected layers have been utilized to capture rich feature representations for ASD detection. [2] VGG-16, with its simplicity and effectiveness, has been employed in ASD detection tasks.[6] NASNet, known for its neural architecture search, has been explored for ASD detection tasks to automatically design effective CNN architectures.[8] EfficientNet, designed for better efficiency-accuracy trade-offs, has shown promise in ASD detection tasks. [7]

A performance comparision of pretrained Convolutional Neural Networks by Israr Ahmad and Rashid.[24].using visual cues of the images for ResNet34, RestNet50, AlexNet, MobileNet, VGG16, VGG19 models to diagnose ASD and compare their performances showed that ResNet50 yeilded 92% accuracy compared to others models.

In summary, various CNN models, including ResNet, AlexNet, InceptionNet, XceptionNet, DenseNet, VGG-16, NASNet, and EfficientNet, have been explored for ASD detection using imaging data. These models have demonstrated effectiveness in capturing discriminative features for distinguishing between individuals with ASD and typically developing individuals, contributing to the advancement of automated ASD diagnosis and intervention. In this research, we have attempted to utilize face images for the diagnosis of ASD by employing several pre-trained CNN architectures. However, further research is needed to enhance the interpretability, generalizability, and robustness of these models in real-world clinical settings.

METHODOLOGY

Pre-trained deep learning models have the potential to democratize access to cutting-edge machine learning technology, be more efficient, and have high transferability. Selection of best model is crucial as different models perform distinctly based on the dataset and its weights. One approach to address this enigma is employing top networks as a whole to achieve good results.[24]. In this research, we employed eight widely recognized pre-trained models, ResNet, VGG16, AlexNet, DenseNet, Inception, EfficientNet, NASNet and Xception - to evaluate their performance in the detection of autistic disorders based on facial images. The block architecture of the proposed methodology is illustrated in Figure 1.

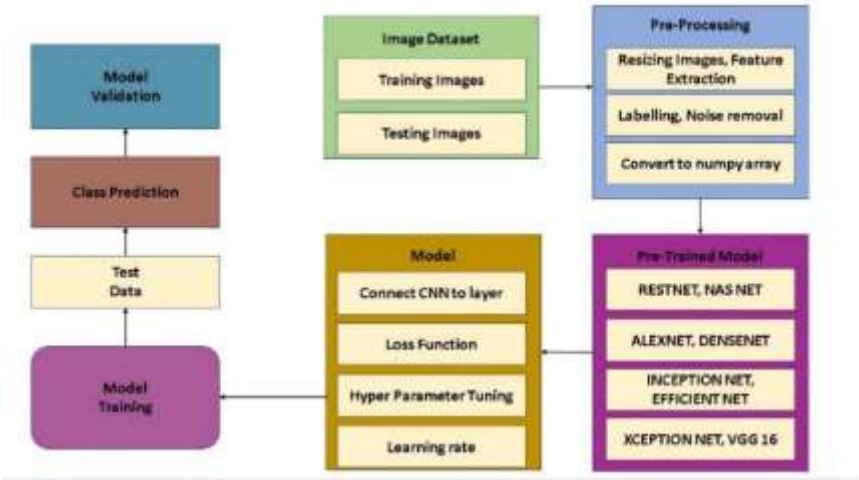

Figure 1

Image analysis through visual traits like eye gaze patterns, Body language, including gestures and posture clues enable the Machine learning and Deep learning to extract and evaluate these properties from the images. The goal of this research is to train and identify the best model based on the given images and subsequently propose the most effective model for the early detection of autism spectrum disorder (ASD).[24].

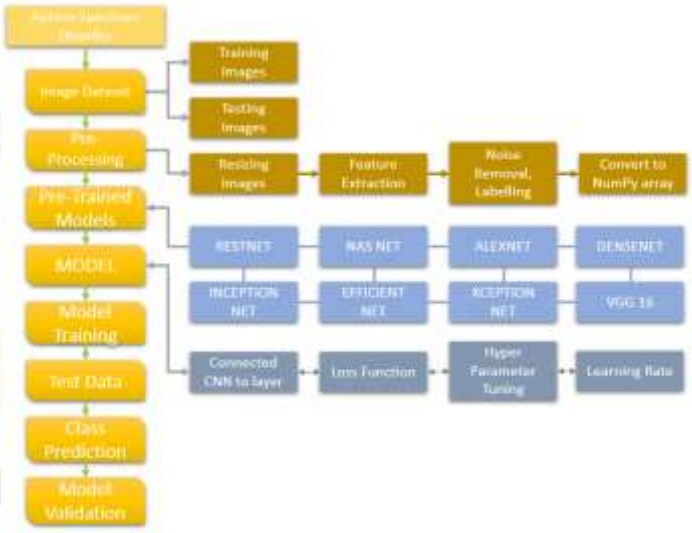

Figure 2

We considered images of the children to detect whether a child is autistic or non-autistic. For data preprocessing, converting list into NumPy arrays gives indexing to the images that makes it easier for the model to assess and train on these indexed images. Image resizing has been applied to standardize the dimensions across all images, ensuring uniformity and facilitating computational efficiency during training. Using color contouring on facial images facilitates easier training of the image model because it enhances the contrast and delineates the facial features, making them more distinguishable.

One-hot encoding has been employed to transform categorical labels to suitable machine learning models, particularly beneficial for multi-class classification tasks like autism detection. Below figure 3 is the flow chart diagram that represents our research work flow.

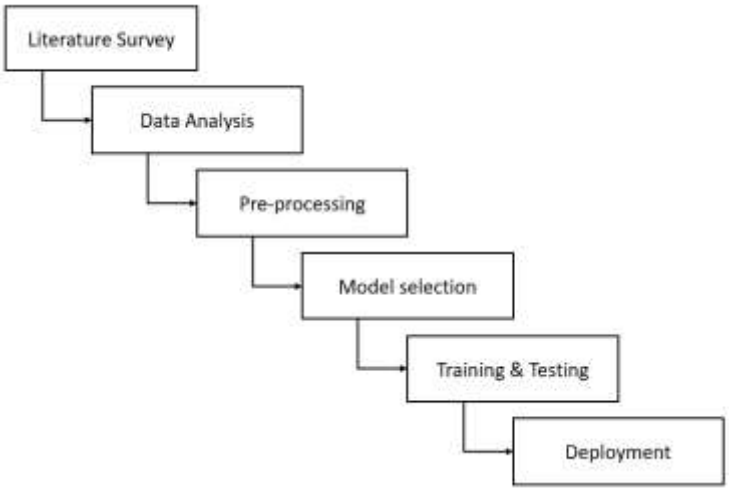

Figure 3

The implementation of 8 distinct CNN architectures for classifying autistic and non-autistic images for Auitistic Image dataset and the working modalities of each CNN architecture model is presented below:

3.1 RESNET

ResNet architecture is used to learn features that are useful for prediction.During training, the weights of the network are adjusted to minimize the prediction error. ResNet is used to address the problem of vanishing gradients indicating that hardly any change comes to weights which makes it difficult for the network to learn and optimize the model parameters. The description of ResNet is provided below.

a. Convolutional Layers:

Res-Net begins with initial convolutional layers to extract low-level features from the input data. These convolutional layers are represented by the equation:

$$H_l = \sigma(W_l * H_{l-1} + b_l). \quad (1)$$

Where H_l is the output feature map of layer l.

W_l represents the weights of the convolutional filters.

b_l represents the biases.

$*$ denotes the convolution operation.

σ represents the activation function, typically ReLU.

b. Residual Blocks:

The core of Res-Net architecture consists of residual blocks. These blocks enable the network to learn residual mappings, facilitating the training of very deep networks. The output of a residual block can be represented as:

$$H_{l+1} = F(H_l) + H_l \quad (2)$$

Where H_{l+1} is the output of the residual block.

F represents the residual function learned by the block.

H_l is the input to the residual block.

c. Fully Connected Layers:

After several layers of convolution and pooling, the feature maps are flattened and passed through fully connected layers for classification. The output of the fully connected layer can be represented by:

$$Z = W_{FC} + F_{flat} + b_{FC} \quad (3)$$

Where Z is the output of the fully connected layer.

W_{FC} represents the weights of the fully connected layer.

F_{flat} represents the flattened feature maps.

b_{FC} represents the biases.

By adding the input to the changed output, ResNet efficiently learns the residual mapping, allowing the network to focus on learning the "difficult" parts of the mapping rather than the complete mapping from scratch.

3.2 ALEXNET

AlexNet architecture is used for robust Image variation. In AlexNet, the number of filters increases as we go deeper hence extracting more features into the model.Also, the filter size decreses as we go ahead resulting in a decrease in the feature map shape. Let's discuss the math behind AlexNet's framework:

a. Convolutional Layers (Conv):

AlexNet consists of five convolutional layers. Each convolutional layer applies a set of learnable filters to the input image to produce feature maps. These feature maps capture different aspects of the input image.

Equation for a convolutional layer: $Z^l = W^l * A^{l-1} + b^l$ (4)

$$A^l = \text{ReLU}(Z^l) \quad (5)$$

where Z^l is the output of the convolutional layer, W^l is the filter weights, A^{l-1} is the input feature map from the previous layer, and b^l is the bias term.

b. Max Pooling Layers:

$$A^l = \text{MaxPooling}(A^{l-1}, \text{filtersize}, \text{stride})$$

where, MaxPooling is the max pooling operation.

c. Fully Connected Layers (FC):

$$Z^l = W^l . A^{l-1} + b^l \quad (6)$$

$$A^l = \text{ReLU}(Z^l) \quad (7)$$

$$A^l_{\text{Dropout}} = \text{Dropout}(A^l, \text{dropoutrate}) \quad (8)$$

where Z^l is the output of the convolutional layer, W^l is the filter weights, A^{l-1} is the input feature map from the previous layer, and b^l is the bias term, and dropout is dropout operation.

d. Soft-max Activation:

$$P(y = i|x) = \frac{e^{z_i}}{\sum_{j=1}^{1000} e^{z_j}} \quad (9)$$

3.3 DENSENET

DenseNet architecture densly connect the various layers and flows through one by one to reduce the size of feature maps, thus making the model parameter efficient and impicit deep supervision for improved flow of gradient.DensNet takes all previous output as an input for a future layer due to the long distnace between input and output kayers and information may vanish brfore reaching its destination. Following is the description of DenseNet.

a. Dense Block:

The key innovation in Dense Net is the dense block. In a dense block, each layer receives feature maps from all preceding layers and passes its own feature maps to all subsequent layers.

Let's denote the input to the *hlth* layer in a dense block as *xl*. The output of the *hlth* layer, denoted as *Hl*(*xl*), is computed as follows:

$$H_l\,(xl) = \sigma(W_l * [Concat\,(X_0\,, X_1, X_2\,\ldots..X_{l-1})] + b_l) \quad (10)$$

where:

$*$ represents convolution operation.
Concat denotes concatenation of feature maps.
σ is the activation function (typically ReLU).

When deconstructing the ith layer received all the feature maps from previous layer as input: $[X_0\,, X_1, X_2\,\ldots..X_{l-1}]$ Where $X_l = H_l\,([X_0\,, X_1, X_2\,\ldots..X_{l-1}])$ is considered as a single tensor.

b. Transition Layer:

Between dense blocks, transition layers are used to reduce the number of feature maps and control the model's complexity.
Transition layer operation: $Hl(xl) = \sigma(Wl * xl + bl)$ (11)

Followed by average pooling to reduce the spatial dimensions.

The model performs three different consecutive operations:

Batch- Normalization (BN), followed by a ReLU and a 3×3 convolution operation. In the transaction block, 1×1 convolutional operations are performed with BN followed by a 2×2 average pooling layer.

3.4 INCEPTION NET

Inception Net is designed for image classification, segmentation and detection. The idea behind this model is to allow the network to learn spatial and temporal features at different scales and then concatenate them together to form a more comprehensive representation of the input data. The model is used to train more efficiently and faster than other models. The discription of Inception is given below.

a. Inception Module:

The fundamental building block of Inception is the inception module, which performs parallel convolutions of different sizes on the input feature maps and concatenates the outputs.

Let's denote the input to the inception module as *x*.
The output Inception(*x*) is computed as follows:

$$Inception(x) = Concat(Conv\,1x1(x), Conv3x3(x), Conv5x5(x), MaxPool3x3(x))$$

where:

- Conv1*x*1,Conv3*x*3,Conv5*x*5 are convolutional operations with filter sizes 1×1, 3×3, and 5×5 respectively.
- MaxPool3*x*3 is max-pooling with a 3×3 filter size.

b. Factorization:

To reduce computational cost, 5×5 convolutions are factorized into two 3×3 convolutions.

$\text{Conv5}x5(x)=\text{Conv3}x3(\text{Conv3}x3(x))$

3.5 EFFICIENT NET

Efficient Net is considered as one of the powerful CNN architecture. The model scaling method employs a compound coefficient to uniformly scale the dimensions of depth, width, and resolution, unlike conventional practices that arbitrarily scale these factors. Consequently, as image resolution advances, the network's depth and width also progress in tandem.

a. **Depthwise Separable Convolution**:

Efficient Net primarily uses depthwise separable convolutions, which consist of a depthwise convolution followed by a pointwise convolution. This reduces the number of parameters and computational cost while maintaining representation power. The operation of depthwise separable convolution can be represented as follows:

$\text{ConvDepthwise}(x)=\text{DepthwiseConv}(x)$

$\text{ConvPointwise}(x)=\text{PointwiseConv}(\text{ConvDepthwise}(x))$

b. Efficient Net Scaling:

Efficient Net introduces compound scaling to uniformly scale network width, depth, and resolution. It uses coefficients to scale the width, depth, and resolution of the baseline network architecture.

$\text{Width}_{new}=\alpha\times\text{Width}_{base}$

$\text{Depth}_{new}=\beta\times\text{Depth}_{base}$

$\text{Resolution}_{new}=\gamma\times\text{Resolution}_{base}$

Where α, β, γ are scaling coefficients.

3.6 XCEPTION NET

Xception model captures features at multiplr spatial scales by using filters of different sizes within the same layer. Xception perform seperate convalution for spatial and depth information which leads to reduction in the number of parameters.

a. Depthwise Separable Convolutions: Xception primarily uses depthwise separable convolutions, which consist of a depthwise convolution followed by a pointwise convolution. This factorization reduces the number of parameters and computational cost while maintaining representation power.

The operation of depthwise separable convolution can be represented as follows:

$\text{ConvDepthwise}(x)=\text{DepthwiseConv}(x)$

$\text{ConvPointwise}(x)=\text{PointwiseConv}(\text{ConvDepthwise}(x))$

3.7 VGG-16

VGG 16 make sure that for every two or three convolutional layers, filters and max-pooling is applied to reduce the spatial dimensions. The increased depth allows the VGG16 to learn more complex features from the input images which leads to improved performance. The description of VGG16 is provided below.

a. Convolutional Layers (Conv):

The VGG-16 model consists of 13 convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function. The convolutional layers perform feature extraction by applying a set of learnable filters to the input image. Mathematically, the operation of a convolutional layer can be represented as:

$$Z^l = W^l * A^{l-1} + b^l \quad (12)$$

Where Z^l is the output of the l^{th} layer.

W^l is the set of learnable filters (weights) for the l^{th} layer.

A^{l-1} is the output of the previous layer.

b^l is the bias term.

$*$ denotes the convolution operation.

b. ReLU Activation Function:

ReLU (Rectified Linear Unit) is applied element-wise to the output of each convolutional layer to introduce non-linearity:

$$A^l = ReLU(Z^l) \quad (13)$$

$$ReLU(x) = \max(0, x)$$

c.Max Pooling Layers:

After some of the convolutional layers, max-pooling layers are applied to reduce the spatial dimensions of the feature maps while retaining the most important information. It selects the maximum value from each window. The mathematical representation of max pooling is:

$$A_{[l]} = MaxPool(A_{[l-1]}, stride = 2, filter_{size} = 2)$$

d. Fully Connected (Dense) Layers:

The last three layers of the VGG-16 model are fully connected layers. These layers take the flattened output of the preceding convolutional layers and perform classification. Mathematically, the operation of a fully connected layer can be represented as:

$$Z^l = W^l A^{l-1} * b^l \quad (14)$$

Where Z^l is the output of the l^{th} layer.

W^l is the weight matrix.

A^{l-1} is the output of the previous layer.

b^l is the bias term.

e.Softmax Activation:

The final layer of the VGG-16 model uses a softmax activation function to produce probabilities for each class. It converts the raw scores into probabilities:

$\hat{y}$ = Softmax (Z^l)

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad (15)$$

Where $\hat{y}$ is the output probability vector .

K is the number of classes.

Z_i is the raw score for class i.

This study utilizes eight different CNN architecture models with each architecture offering unique design principles and trade-offs in terms of model depth, parameter efficiency, and computational requirements. Thus, each network has a specific number of layers and parameters. The number of layers and the number of learnable parameters for each model, which provides insight into the number of filters used by the models, is presented in Table 1. By considering this information, the study aims to provide comprehensive understanding on the architectural complexities and the capacity inherit in the models.

TABLE 1. Pre-trained networks, number of convolutions layers summary.

Architecture	Layers	Trainable parameters
NAS Net	4	19.9M
Xception Net	5	4.91M
Alex Net	21	56.5M
Res Net	15	28.9M
VGG -16	16	18.9M
Inception Net	9	65.5M
Efficient Net	5	71.6M
Dense Net	17	20.2M

DATASET AND EVALUATION.

This study uses a publicly accessible dataset for ASD[46]. The dataset comprises of four components namely Train, Test, consolidation, and Valid where a random split required in data loader before feeding to CNN. The consolidation and valid sections contain two subdirectories of autistic and non-autistic for training, validation, and testing. Among the two categories, each category comprises correspondent images with the same dimensions and each image was loaded as 3 channels image.

Almost every image in the dataset is captured while posing straight face in front of camera. The dataset contained a total of 2940 images for both autistic and non-autistic, where the number of each class were equal, it means the dataset was balance. Sample images of dataset are presented in Figure 4.

Figure 4

To improve the result analysis four specific image preprocessing techniques were applied on the ASD dataset. First technique being resizing of the images or dimensionality reduction for lesser space usage and faster performance of the models. Color contrast enhancement which involves transforming an image from one color space to another, Segmentation which is used to divide an image based on its

content, and Normalization which involves adjusting pixel values to predetermined range are other pre-processing techniques employed on the ASD dataset.

For data preprocessing, several steps have been implemented. Firstly, image resizing has been applied to standardize the dimensions across all images, ensuring uniformity and facilitating computational efficiency during training. Using color contouring on facial images facilitates easier training of the image model because it enhances the contrast and delineates the facial features, making them more distinguishable.

The list into NumPy arrays representing the images have been converted into lists, likely to facilitate easier manipulation and compatibility with certain operations or libraries. Converting list NumPy arrays gives indexing to the images that makes it easier for the model to assess and train on these indexed images. One-hot encoding has been employed to transform categorical labels into a format suitable for machine learning algorithms, particularly beneficial for multi-class classification tasks like autism detection.

Furthermore, the inclusion of standardized evaluation metrics like validation accuracy, validation loss, test loss and confusion matrix are deployed for performance analysis. These features make the Autistic Image Dataset an essential component in the methodology section of a research paper, enabling a standardized evaluation of model accuracy and facilitating meaningful comparisons between different models and approaches.

RESULTS AND DISCUSSIONS.

This research utilized 80 % of data for training and validation while the rest 20% for testing to evaluate the performance of all the models for ASD detection. The experimental results focus on the comparison of results in terms of accuracy for all models.

5.1 The results comparison of Different models

Initially for the training of various models, image size of 224X224X3 were utilized with channel being 3. Image dataset with two classes and batch size of 34 were employed for model training. A constant of 20 epochs for all the models were maintained till the end of performance where each epoch represents entire dataset being passed through the model once. Thus, allowing the model to extract the features from input dataset and trained based on those patterns. All model were trained with a selective learning rate and all the frozen layers from the base models which later become unfrozen along with few custom additional layers were utilized for model training.

Comparison between different models in terms of Training accuracy, Validation accuracy, Test loss

TABLE 5.1 Models comparison on ASD

Model	Layers	Time/Epoch	Training Accuracy	Validation/Test Accuracy	Test Loss
Nas Net	4	177s	0.75	0.62	0.73
Xception Net	5	38s	0.90	0.81	0.52
Alex Net	21	6m	0.70	0.77	0.51
Res Net	15	991s	0.76	0.69	0.59
VGG -16	16	720s	0.81	0.75	0.50
Inception Net	9	30s	0.93	0.80	0.78
Dense Net	17	23m 47s	0.80	0.78	0.42
Efficient Net	5	506ms	0.89	0.82	0.44

The above table is tabulated based on comparison of eight different CNN models with parameters being number of layers, Time for each epoch, Training and validation accuracy, and Test loss functions.

On observation from the above table the study indicates that NASNet, AlexNet, ResNet, and DenseNet show accuracy score in the range of 70%-80% which is comparatively less than others models as number of layers various greatly thus models demanded increased epoch for better training. VGG16 being one of the best model in general yielded an accuracy score of 81% for the data and Xception model gave an accuracy of 90% which is considered as good performance. On contrary Inception model gave a great accuracy score of 93% however it's also the model with highest loss which of 78% that made the model unfit. Among all the models, DenseNet produced very less loss of 42% due to its reduced parameters and it's skip technique between the densely-connected layers. Finally Efficient Net provided a Training accuracy of 89 % and Validation accuracy of 82% which is closer to Xception Net with Training accuracy of 90% and Validation accuracy of 81% making both these models the best choice for ASD Detection.

5.2 Performance of Proposed models for ASD detection.

In this study, all the proposed models performance is outlined through graphic representation that indicate the Training and validation accuracy of each model. The graph represents epoch against the accuracy.

Figure 5.2.1 Training and validation accuracy of VGG-16

On interpreting the proposed VGG 16 model graph below, initially Validation accuracy was at 78% which gradually dropped to 72% at fifth epoch. But upon further training the accuracy increases to 82% at 12th epoch which suddenly hit the bottom of 70% accuracy. Finally, it gradually increased to 75% as the epoch reached 20. Similarly with the Training accuracy of the VGG 16 model. Similarly for the rest models Training and Validation accuracy is plot with epoch against accuracy that yields the results as tabulated in Table 5.1.

Training and Validation Loss are illustrated in Figure 5.2.2. Upon analyzing the graph, it is evident that the Training loss began at 57% and reached 51% at the 6th epoch. However, it increased to 52% and again suddenly dropped to 49% at 8th epoch. The Training loss was fluctuating between 50 and 48 but later it attained minimum loss of 46%. But this didn't last for long, soon the training loss reached 50% for VGG16 at 20th epoch. Similarly Training and validation Loss function graph was plot for the remaining models based on their performances.

Figure 5.2.2.Training and validation loss of VGG-16

A confusion matrix is a powerful tool for evaluating the performance of deep learning models, particularly for classification tasks. It can help to identify patterns in the model's predictions, such as which classes are being misclassified and to what degree. This can be useful in identifying areas where the model needs to be improved and can also be used to compare the performance of different models.

In the presented confusion matrix below, right predictions are exhibited diagonally in pink colour, and incorrect predictions off-diagonally in black colour. The actual(True) labels are presented on *X*-axis and predicted labels are presented on *Y*-axis.

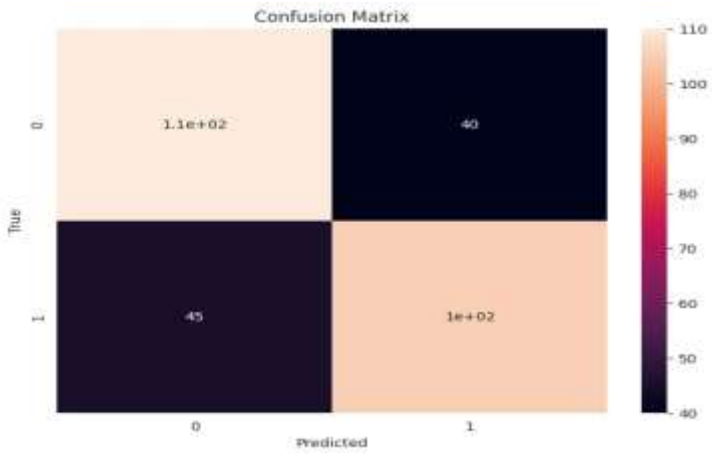

Figure 5.2.3 Confusion matrix for VGG-16

The top left of confusion matrix illustrates number of True Negatives that is model correctly predicting instances belonging to class 0 which is Autistic. Thus, 110 instances were detected as autistic. The top right shows the number of False Positives that is model incorrectly predicted instances as class 1 which is Non-Autistic when they were actually Autistic. So, 40 images were predicted as Non-Autistic while there were Autistic. The bottom right shows the number of True Positives that is model incorrectly predicted instances as Autistic when they were Non-Autistic that came up to be 45 images. Finally, the bottom right shows the number of true positives that is correctly predicted as Non-Autistic. Thus, 110 images were detected as Non-Autistic.

Overall, the model seems to perform relatively well, but it has some misclassifications when prediction Autistic. Thus, indicating further training and study is can be done to enhance the performance of the model. Similarly, confusion matrix was plotted for the other models used for this study.

CONCLUSION

The performance of well-established pre-trained models for the task of autism spectrum disorder detection was evaluated and the characteristics of these models, such as the relationship between resolution and performance, the number of CNN layers, the epochs, and the learning parameters, were investigated[24]. In conclusion, this research paper comprehensively evaluated the performance of eight Convolutional Neural Network (CNN) models, namely AlexNet, EfficientNet, ResNet, InceptionNet, XceptionNet, DenseNet, NASNet, across various metrics including Training and validation accuracy, Training and validation loss, and confusion matrix. Through rigorous experimentation and analysis, valuable insights were gained into the strengths and weaknesses of each model in handling complex image classification tasks. The results highlighted the diverse capabilities of these CNN architectures. Obtained that the EfficientNet CNN architecture model provides the best accuracy for training and validating.

Our research stands out in its approach to Autism disorder detection by leveraging the strengths of multiple CNN models and combining their performances to identify the best one. By meticulously tabulating the results of the top eight CNN models, we aim to achieve unparalleled accuracy and reliability in our detection system. This method allows us to capitalize on the unique strengths of each model, effectively mitigating biases and enhancing overall performance.

This research contributes to the understanding of CNN model performance in image classification tasks and provides a basis for informed model selection based on specific requirements and constraints. Future research could explore further optimization techniques, ensemble methods, and transfer learning strategies to enhance the performance and generalization capabilities of CNN models in diverse application domains.

REFERENCES

[1] "Early Diagnosis of Autism Disease Based on Deep Convolutional Neural Networks and fMRI Imaging" by Chen et al. 2019, IEEE

[2] "A Deep Learning Approach for Autism Spectrum Disorder Detection from fMRI Data" by Zhou et al. 2019, Frontiers in Neuroscience

[3] "Automatic Detection of Autism Spectrum Disorder in Children's MRI Data Using Deep Learning and the Importance of Normalized Input" by Kassani et al. 2020, Sensors (MDPI)

[4] "CNN-Based Framework for Autism Spectrum Disorder Diagnosis from fMRI Data: Abnormality-Aware and Transferable Feature Learning" by Sarraf and Tofighi. 2016, Scientific Reports (Nature Publishing Group).

[5] "Autism Spectrum Disorder Detection from MRI Data Using Convolutional Neural Networks" by Thabtah et al.2019, IEEE

[6] "Identification of Autism Spectrum Disorder Using Deep Learning and the ABIDE Dataset" by Heinsfeld et al.2018, NeuroImage.

[7] "Multi-Modal Deep Learning Models for Autism Spectrum Disorder Diagnosis Using Magnetic Resonance Imaging Data" by Wang et al. 2023, IEEE

[8] "A Deep Learning Approach for Predicting Autism Spectrum Disorder from Structural MRI Data" by Tan et al.2021, IEEE.

[9] "Autism Spectrum Disorder Prediction by Facial Recognition Using Deep Learning " by Kanimozhi and Dhanasri 2024, IJCRT

[10] **"**Autism spectrum disorder detection using facial images: A performance comparison of pretrained convolutional neural networks " by Israr Ahmad and Rashid 2024, IET

[11] "Classi fi cation of autism spectrum disorder using supervised learning of brain connectivity measures extracted from synchro states," J. Neural Eng., vol. 046019, 2014.

[12] "Classification of Autism Spectrum Disorder Using Random Support Vector Machine Cluster," Front. Genet., vol. 9, no. FEB, p. 18, Feb. 2018.

[13] "Automated detection of autism spectrum disorder using a convolutional neural network," Frontiers in neuroscience, vol. 13, 2020.

[14] "Classifying autism spectrum disorder using the temporal statistics of resting-state functional mri data with 3d convolutional neural networks," Frontiers in psychiatry, vol. 11, 2020.

[15] "The autism diagnostic observation schedule—generic: A standard measure of social and communication deficits associated with the spectrum of autism," Journal of autism and developmental disorders, vol. 30, no. 3, pp 2000.