Detection of Autism Spectrum Disorder

A PROJECT REPORT

Submitted by

BL.EN.U4CSE20081 Kotha Chakradhar

BL.EN.U4CSE20082 Kotte Thulasi Tharun

BL.EN.U4CSE20124 Periyavaram Sandesh Kumar Reddy

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING, BENGALURU

AMRITA VISHWA VIDYAPEETHAM

BENGALURU 560 035

MAY 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, BENGALURU, 560035

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Detection of Autism Spectrum Disorder"** submitted by

BL.EN.U4CSE20081	Kotha Chakradhar
BL.EN.U4CSE20082	Kotte Thulasi Tharun
BL.EN.U4CSE20124	Periyavaram Sandesh Kumar Reddy

in partial fulfillment of the requirements as part of **Bachelor of Technology** in "**COMPUTER SCIENCE AND ENGINEERING"** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Bengaluru.

Dr. Thangam S.	
Assistant Professor Sr. Gr.	Chair
Dept. of CSE, School of Computing	Dept. of CSE, School of Computing

This project report was evaluated by us on 8th May 2024.

Internal Examiner 1	Internal Examiner 2	External Examiner

ACKNOWLEDGEMENTS

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of **“AMMA”, MATA AMRITANANDAMAYI DEVI** who showered her blessing upon us throughout the course of this project work.

We owe our gratitude to **Prof. Manoj P.**, Director, Amrita Vishwa Vidyapeetham Bengaluru Campus. We would like to place our heartfelt gratitude to **Dr. Gopalakrishnan E.A.,** Principal, Amrita School of Computing, Bengaluru for his valuable support and inspiration.

It is a great pleasure to express our gratitude and indebtedness to our project guide **Dr Thangam S., Assistant Professor (Sr. Gr.),** Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru for her/his valuable guidance, encouragement, moral support, and affection throughout the project work.

We would like to thank express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported and encouraged us in all our endeavors.

ABSTRACT

Autism Spectrum Disorder (ASD) is a complex condition that affects social interactions, communication, and behavior. It's important to identify ASD early so that people with the condition can get the help they need. This research project uses advanced computer technology to develop a system that can automatically detect ASD by analyzing data about a person's behavior. If successful, this system could help doctors diagnose ASD more quickly and accurately.

The study employs a comprehensive dataset comprising diverse features such as facial expressions, speech patterns, eye movements, and various behavioral responses. These multi-modal data are preprocessed and integrated to create a unified representation suitable for deep learning models.

Deep learning models, including convolutional and recurrent neural networks, are used to find important patterns in data. Transfer learning techniques are used to modify pre-trained models to work better for autism spectrum disorder detection, which is a difficult problem because there isn't much data. The proposed method is evaluated using strict cross-validation techniques and compared to traditional machine learning techniques and clinical assessments.

By providing clinicians and researchers with a deeper understanding of the salient features driving the diagnostic predictions, our framework not only enhances diagnostic accuracy but also offers valuable insights into the underlying mechanisms of ASD pathology. This research extensive approach has the ability to provide valuable insights into the causes of ASD and enhance diagnostic methods. This, in turn, could lead to personalized treatments and support techniques that are specifically designed to meet the unique requirements of individuals with ASD.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER - 1
INTRODUCTION

1.1 Overview of Autism Spectrum Disorder

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental condition characterized by challenges in social interaction, communication, and repetitive behaviors. The spectrum encompasses a wide range of symptoms and severity levels, leading to individualized experiences for each person with ASD. Early signs of ASD often manifest in infancy or early childhood, with some children displaying delays in language development, lack of eye contact, and difficulty in understanding social cues. As individuals with ASD grow older, they may exhibit specific interests or repetitive behaviors, struggle with changes in routine or sensory sensitivities, and face challenges in forming and maintaining relationships with others. Diagnosing ASD involves a comprehensive evaluation that considers the individual's behavior, communication abilities, and developmental history. Early detection and intervention are crucial in helping individuals with ASD reach their full potential, as specialized therapies and support services can enhance their communication skills, social interactions, and overall quality of life. While there is currently no known cure for ASD, ongoing research and advancements in understanding the condition have led to more effective strategies for managing symptoms and improving outcomes for individuals with ASD. By increasing awareness about the signs and characteristics of ASD, promoting early screening and diagnosis, and providing tailored support and resources, we can create a more inclusive and supportive environment for individuals on the autism spectrum to thrive and engage meaningfully in society [17].

1.2 Importance of Early Detection

Early detection of Autism Spectrum Disorder (ASD) is crucial for a variety of reasons. First and foremost, early detection allows for early intervention, which has been shown to significantly improve outcomes for individuals with ASD. By identifying the disorder at a younger age, children can begin receiving tailored treatments and therapies that can help address their unique needs and challenges. Early intervention can lead to improved social skills, communication abilities, and overall quality of life for individuals with ASD. Another important reason for early detection of ASD is the ability to start building a strong support network for both the individual with ASD and their families. With an early diagnosis, families can access resources, assistance, and guidance from professionals and support groups that specialize in ASD. This can help families better understand the disorder, navigate the various challenges that may arise, and access the appropriate services and interventions for their loved one with ASD. Ultimately, early detection of ASD enables individuals to receive the support and care they need as early as possible, setting them on a path towards better outcomes and a brighter future.

1.3 Common Signs and Symptoms

Three common signs and symptoms for detecting Autism Spectrum Disorder (ASD) include challenges in social communication, repetitive behaviors, and sensory sensitivities. In terms of social communication, individuals with ASD may have difficulty with verbal and nonverbal communication cues such as maintaining eye contact, understanding gestures, and expressing emotions. They may also struggle with recognizing social cues or engaging in reciprocal conversations. Repetitive behaviors are another hallmark sign of ASD, which can include repetitive movements like hand flapping or rocking, rigid adherence to routines, and intense fixations on specific topics. Sensory sensitivities are also frequently seen in individuals with ASD, as they may be hypersensitive or hyposensitive to sensory stimuli such as sound, light, touch, or textures. These sensitivities can lead to strong reactions or aversions to certain sensory inputs, impacting their daily functioning [14]. By recognizing these common signs and symptoms of ASD, individuals and professionals can facilitate early detection and intervention to provide appropriate support and resources for individuals with autism.

1.4 Challenges in Diagnosis

Diagnosing Autism Spectrum Disorder (ASD) presents several challenges due to its complex and heterogeneous nature. First, the wide variability and overlap in symptoms among different neurodevelopmental disorders can make it difficult to distinguish ASD from other conditions such as intellectual disability or language disorders. This diagnostic overshadowing can lead to misdiagnosis or underdiagnosis of ASD, delaying appropriate interventions. Second, the reliance on behavioral observation and subjective assessments by clinicians can introduce variability and subjectivity into the diagnostic process. Objective measures such as biomarkers or imaging techniques are still under development and not widely available for routine clinical use. Third, the presentation of ASD symptoms can change over time, leading to diagnostic uncertainty, especially in young children who may exhibit a delay in symptom manifestation. This developmental trajectory of symptoms necessitates careful monitoring and reassessment to ensure an accurate diagnosis. Lastly, cultural and socioeconomic factors can impact access to diagnostic services and influence the way ASD symptoms are perceived and reported. Disparities in healthcare resources and cultural beliefs about neurodevelopmental disorders can contribute to disparities in diagnosis and treatment for individuals with ASD. Addressing these challenges requires a multidisciplinary and culturally sensitive approach that incorporates standardized assessment tools, collaboration across healthcare disciplines, and awareness of the diverse expressions of ASD across different populations.

1.5 Statistics

According to statistics, more children in India are being diagnosed with autism; between ages two and nine, it is estimated that ASD affects 1 to 1.5% of kids in the country. Studies have indicated lower autism rates in developing countries; 1 out of 45 children (ages 3-17) in the US versus only 23 out of 10,000 children in India are diagnosed with autism. About 10 percent of all students at schools have various levels of learning difficulties ranging from mild ones which might look like simple laziness or lack of motivation up to severe cases when such a person cannot speak properly while handling simple tasks such as reading and writing without assistance has become a challenge for them. Regrettably, due to minimum resources accessible, general ignorance, and wrong diagnosis, there has been an overwhelming increase in the number of this disorder among people residing in the nation [15]. One more contributing factor [16] on why this diagnosis often comes late is because it may be confused with mental retardation or schizophrenia in adults who exhibit such symptoms but do not always develop autism. Drafting effective policies and intervening early can go a long way in solving the problems faced by autistic children and integrating them into society

CHAPTER - 2

LITERATURE REVIEW

The identification of ASD is crucial for early intervention and support, leading to improved outcomes for individuals and better management of healthcare resources. Early detection [12] allows for timely access to interventions and therapies tailored to the individual's needs, promoting optimal development and functioning. Moreover, [13] accurate diagnosis aids in the effective allocation of healthcare resources and the implementation of appropriate support services, benefiting both individuals with ASD and the healthcare system as a whole. A graph kernel-based clustering method is proposed in the study [1] to explore transdiagnostic biotypes for schizophrenia (SZ) and ASD using functional connectivity (FC) estimated from fMRI data. From among all subjects' whole-brain FCs, the proposed method detects common subnetworks and then calculates the graph kernel similarity to determine the clustering relationship between subjects within the constructed Graphs. The results show significant differences in FC between the identified biotypes for SZ and ASD. The article [2] presents a deep convolutional neural network-based detection system for ASD in facial images. The system aims to identify ASD at earlier stages to provide therapy for skill improvement. The researchers used a dataset from Kaggle, split it into training and testing sets, and got an accuracy of 88% and loss of 0.53. The article [3] discusses the use of machine learning approaches to screen and predict ASD in individuals. This paper presents a wide range of characteristics involved in diagnosing ASD from ASD datasets that can be obtained from open resources and makes an application of ML techniques including Logistic regression, XGboost, SVC, Naive Bayes. According to the results of the research, XGboost performs well giving more accurate predictions. According to the article machine learning aided screening methods could serve as an approach to improve on those aspects in terms of diagnosis accuracy, speed as well as access permissions in blocking out persons diagnosed with autism spectrum disorder. The article [4] proposes a deep neural network model with multilabel classification for emotion analysis in children with ASD. The model uses face landmarks to predict ASD/non-ASD and detect emotions. The results of the model is being tested on different datasets and provides promising results.

The article [5] discusses the prediction of ASD using Efficient Net, a convolutional neural network. The model utilizes facial images to detect whether a person is autistic or typically developing. The study used a dataset of 2530 facial images for training and achieved an accuracy level of 88%. The article also provides an overview of related work in the field of ASD detection using machine learning techniques. The article [6] provides a review of fuzzy logic-based approaches for the detection of ASD. ASD can be defined as a condition that originates in the brain and has an impact on how a person perceives others around him/her hence affecting learning capabilities, emotional responses as well as behavioral patterns. Early detection of ASD is necessary for effective intervention. The article discusses various fuzzy logic methodologies and concepts used for ASD detection, including fuzzy expert systems, artificial neural networks, and fuzzy neural networks. It also presents an overview of studies that have utilized fuzzy logic for ASD detection, highlighting the methods, goals, and accuracy achieved in each study. Applying eye-trackers of machine learning to identify autism in high-functioning adults is argued for in this paper.

Authors [7] propose that combining eye-tracking data from web-related tasks with transfer learning can yield results that are really hopeful with regard to automatic autism detection while the use of decision trees, logistic regression and transfer learning enabled reaching high classification accuracy by researchers. The study highlights the importance of objective measures in autism assessment and the potential of eye-tracking technology in early detection. The article [8] presents a pilot study on using advanced computer audition technology to detect ASD in mice.

The study focuses on analyzing the ultrasonic vocalizations of mice, which have been shown to be efficient markers for distinguishing wildtype mice from those with ASD. In order to extract high-level representations from the USVs, the authors propose use of a large-scale audio neural network which has been pre-trained. The results show promising accuracy in detecting ASD in mice. The article [9] presents a deep learning approach called DL-ASD for identifying ASD in children aged 1-10. The proposed model uses a convolutional neural network (CNN) trained on image datasets to classify individuals as having ASD or not. The model achieves a classification accuracy of up to 87%. The article [10] discusses the use of eye tracking (ET) and electroencephalography (EEG) data in ASD research. The authors explain how correlating ET and EEG data can help identify diagnostic biomarkers and shed light on the inconsistent findings in ASD research. They review eight articles that have used ET-EEG correlative analytics and summarize the reported correlations between ET and EEG patterns. The Autism-Image-Data [11] dataset contains a diverse range of images, featuring children aged 3 to 8 with various characteristics. This dataset can be utilized in our survey to provide a comprehensive representation. The study [12] investigates the potential of distinguishing autistic children from typically developing children using facial biomarkers and Convolutional Neural Network models. Densenet121 achieved the highest performance with an accuracy of 96.66%, outperforming other models in the analysis.

The paper [13] comes up with a method of diagnosing ASD using structural Magnetic Resonance Imaging (sMRI) through detection of outlying values. The distinction between healthy subjects and those who have Anorexia Nervosa (AN) in terms of the volume and density within Disinhibited Eating Behavior (DEB) related brain regions is tested by employing Spatio-Temporal Generative Adversarial Networks (ST-GAN). The mechanism introduced here consists of a parallel structure, where Generator generates 3 quality images, while Discriminator is responsible for determining if the images are the ones created by Generator or generated in some other way.

The study [14] suggested an unsupervised method to diagnose ASD using deep learning models such as UNet, GAN, and SAGAN. The study [15] is to develop a UKF-based framework which will allow for tracking of ECG signals in continuous real time in order to estimate anxiety in patients with developmental disabilities. The main idea behind this method is that it helps people understand each other better, by being able to capture symptoms quickly enough so that interventions can be made promptly which ultimately will reduce death rates. The exceptional capturing and analyzing visual attention patterns, eye tracking has a potential to detect autism [16] while also giving useful information on cognitive processes and uncommon behaviors by autistic children. The main objective of the paper [17] is to develop a system of emotion recognition that can be used by special needs people mainly those with autism to assist carers at schools. It is proposed that through accurate identification of emotions the system will reduce misinterpretations as well as enhancing the emotional wellness of such persons.

The paper [18] proposes a multi-object detection system employing YOLO (You Only Look Once) algorithm to efficiently detect and localize objects in images. The paper [19] conducts a comparative analysis of various image processing methods for detecting human presence in images. The paper [20] demonstrates real-time object detection capabilities using YOLOv8, showcasing its efficiency in quickly identifying objects in live video streams. A real-time [21] pedestrian detection system was achieved by adapting YOLO v2, demonstrating its effectiveness in swiftly identifying pedestrians in video streams. The paper [22] proposes a deep learning approach utilizing YOLO (You Only Look Once) for detecting potholes in Indian roads, aiming to enhance road maintenance and safety measures. A comparative analysis [23] between image processing and deep learning techniques for image dehazing, aiming to evaluate their effectiveness in enhancing visibility in hazy images. The study [24] looks at object detection and recognition in real-time applications with YOLO, showing how well it works. The efficiency of YOLO is shown here with its ability to find objects quickly on screen.

They also demonstrated that YOLO can detect many different items at once from pictures or video clips where people might not see everything that's going on as well as how these are handled from the point of view of such media they're embedded into by discussing the effectiveness for image categorization as well as location marking including detection which can track various things at once in photo or video shots something human perception goes beyond for instance. The paper [25] focuses on developing a neural network architecture capable of accurately categorizing images into predefined classes, thereby advancing automated image recognition tasks.

Researchers have explored various methods to enhance the detection of ASD from facial images, aiming to leverage facial features as potential biomarkers for improved diagnostic accuracy. Several studies have looked at different aspects of autism spectrum disorder (ASD). However, it is urgent to have combined methods, which look at many aspects at the same time like genetics, neurobiology, behavior patterns and the environment around us. In order to help us diagnose infants who have ASD before it becomes severe we may need to use biomarkers which are stable enough for example neuroimaging tests, genetic markers or other objective measures. Moreover, tracking people with autism from their childhood years to adulthood would be a helpful move forward for the discipline. Such studies should help people understand how autism takes shape over time as well as examine what can lead to this disease. Tracking these people could also help determine whether treating them early enough can completely cure or not. By addressing these research gaps mainly by combining the behavioral patterns and the environment, we can advance our understanding of ASD and improve outcomes for affected individuals and their families.

CHAPTER – 3

SYSTEM SPECIFICATIONS

3.1 Software requirements

1. **Google Jupyter / Colab:**

Jupyter and Google Colab are interactive computing environments that help with collaborative code generation, sharing, and execution. They are useful resources for creating and recording machine learning models for the identification of lung diseases. These platforms provide smooth interaction with different libraries and frameworks and support Python programming.

2. **Python environment:**

The main programming language used to create machine learning apps and algorithms for autism detection is Python. The healthcare domain is a popular choice for academics and developers because to its huge libraries, ease of comprehension, and simplicity.

3. **TensorFlow:**

TensorFlow, which is widely used in lung disease detection, offers deep learning model development and training tools. It may be used for a variety of tasks, from picture categorization to more intricate medical diagnostics, because to its flexibility and scalability.

4. **Scikit-Learn:**

Scikit-Learn provides rapid and simple tools for data analysis and modelling. Its many features and methods are essential for pre-processing data, creating models, and evaluating how well such models perform in terms of identifying facial features.

5. **OpenCV:**

OpenCV may be used to image processing, feature extraction, and analysis of medical pictures for the purpose of detection. This will help identify irregularities and patterns.

6. Pandas:

Pandas is a Python toolkit for data analysis and manipulation. It is especially helpful for analyzing and arranging data in organized way. Pandas makes it easier to do activities like cleaning, exploring, and getting datasets ready for model training by offering data structures like DataFrames.

7. NumPy:

A Python library for numerical computing is called NumPy. It is crucial for the numerical operations and calculations required in processing pictures and data for the identification since it supports huge, multi-dimensional arrays and matrices.

8. Keras:

It functions as a training and modeling interface for deep learning, frequently intandem with TensorFlow. Keras makes building neural networks and experimenting with various architectures easier when it comes to autism detection.

9. Matplotlib:

Matplotlib is an extensive Python data visualization toolkit that is essential. Researchers and healthcare practitioners may display medical data, diagnostic results, and model performance e using its extensive toolkit for producing static, animated, and interactive charts.

10. OS Python Module:

An interface for interacting with the operating system that is agnostic of platform is offered by the standard Python library known as the "os module." The OS library is essential for controlling file paths, directories, and system-level instructions.

11. Streamlit:

Streamlit has an intuitive Python framework which enables data-centric web applications to be quickly developed and deployed, thus making it simple to deploy webpages. Streamlit uses simple syntax and interactive elements allowing data-driven insights to be created and shared seamlessly through web browsers thus making it valuable for data scientists and developers too.

12. TensorBoard:

TensorBoard works by being a visualization tool for TensorFlow, thus easing the comprehension and elimination of bugs in machine learning models, with interactive visualization in training metrics including embeddings and model graphs, thereby improving the monitoring of deep learning experiments for their optimization. With TensorBoard, it allows you to monitor your models in terms of performance, identify areas for improvement in minutes and thus continuously develop better ones without much fuss.

13. Pytorch:

A famous deep learning framework which is open-source is Pytorch that is renowned for its flexibility and ease of use by most people. To make everything easier, dynamic computational graphs have been incorporated into this particular framework thus allowing for the quick testing of new ideas as well as building up prototypes on neural network models so fast. Generally speaking, this means that when working with complicated structures like those found within artificial intelligence models these tools work better than all others we know about whether they come from academia such as Stanford University NIH’s ImageNet competition winner AlexNet in 2012; however, despite its popularity among researchers, there are few practical applications because most people don’t understand its complexity though this has started changing due to increased awareness about its capabilities over time and its strong community support.

3.2 Hardware requirements

Processing Unit:

CPU: It is advised to use a current, high-clock-speed multi-core CPU (such as an AMD Ryzen 5/7/9 or an Intel i7/i9) for effective data processing and multitasking throughout the development and testing stages.

GPU: Deep learning activities require a specialized Graphics Processing Unit (GPU) to be completed more quickly. As CUDA is designed for TensorFlow and other deep learning packages, NVIDIA GPUs with CUDA cores come highly recommended. To train models at a respectable pace, a GPU with at least 8 GB of VRAM (such as an NVIDIA RTX 2060 or above) would be appropriate.

Memory:

RAM: At least 16GB of RAM is recommended to handle large datasets in memory and to facilitate multiple processes running concurrently during model training and hyperparameter tuning.

Storage:

Hard Drive: An SSD that can hold the operating system, applications, and project data in excess of 512GB. Compared to conventional HDDs, SSDs provide quicker read/write data rates, which is advantageous for loading and processing big information.

Internet Connection: A high-speed internet connection is essential for accessing cloud-based resources, downloading datasets, and utilizing APIs or online platforms like Google Colab.

Peripheral Devices

Monitor: Extended coding sessions and intricate visualization activities benefit greatly from the use of a high-resolution display. Having several displays might increase output.

Input Devices: Considering the quantity of code and contact with the development environment, a comfortable keyboard and mouse are essential.

Cooling System: Effective cooling options, such as premium CPU/GPU coolers and case fans, help keep the temperature at ideal levels while doing computationally demanding activities.

CHAPTER - 4
SYSTEM DESIGN

4.1 Low Level Design:

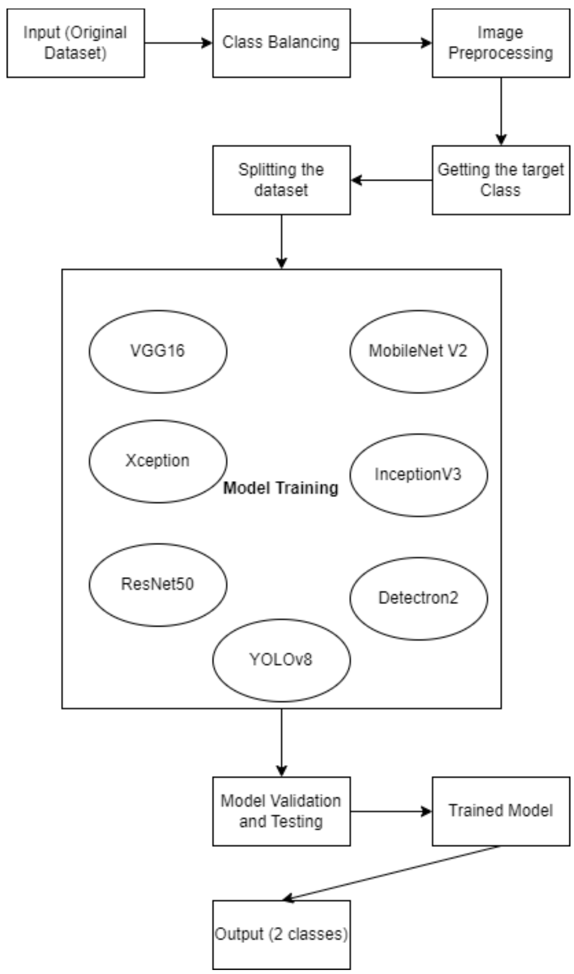

Fig 4.1.1 SYSTEM ARCHITECTURE

Fig 4.1.1 shows the high level design of the system.

- **Input (Original Dataset)**: Initial data that the system will process. It likely refers to a collection of images that need to be classified into two categories.

- **Class Balancing**: Process that adjusts the dataset to address imbalances in the number of data points belonging to different classes.
- **Image Preprocessing**: Images undergo some preparation steps before they are used to train the models. Preprocessing techniques include resizing, cropping, rescaling, flipping, rotating and normalization.
- **Splitting the dataset**: Process of dividing the dataset into two subsets: a training set and a validation set. The training set is used to train the models, and the validation set is used to evaluate the performance of the trained models. 80% of the dataset is taken for training and 20% for validation.
- **Getting the target Class**: The process of identifying the class label for each image in the dataset. The class label indicates whether if the child is having autism or not.
- **VGG16, MobileNet V2, Xception, InceptionV3, ResNet50, Detectron2, YOLOv8**: Different pre-trained deep learning models that are used to classify the images. Each model has its own strengths and weaknesses, and the best model for a particular task will depend on the specific data and requirements.
- **Model Training**: The process of training the deep learning models on the training set. During training, the models learn to identify patterns in the data that can be used to classify new images.
- **Model Validation and Testing**: The trained models are evaluated on the validation set while training to assess their performance. This is typically done by measuring the accuracy and losses of the models on the validation set.
- **Trained Model**: The final output of the system, which is a deep learning model that can be used to classify new images into Autistic or Non Autistic.

4.2 High Level Design:

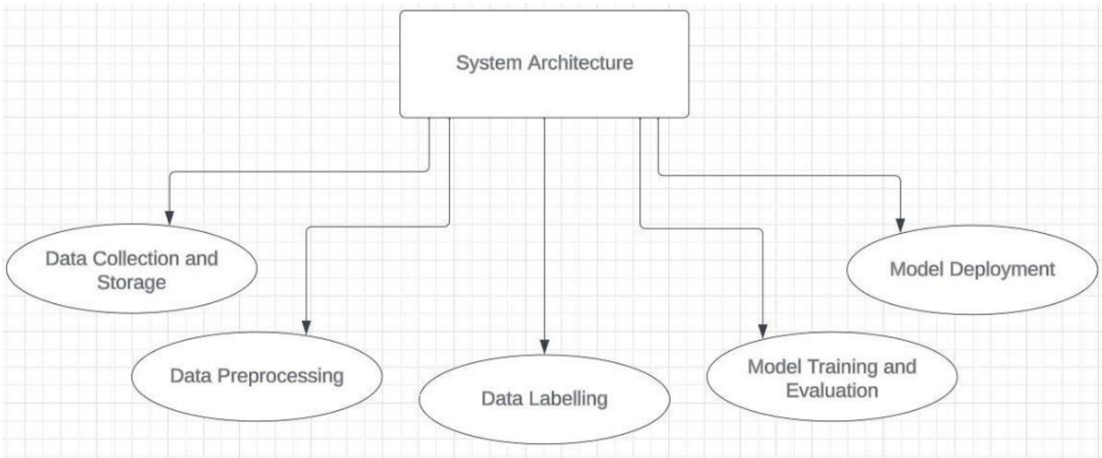

Fig 4.2.1 High level Design

Data Collection and Storage: Preparation of Autism Image Dataset.

Data Preprocessing: Cleaning and enhancing the data to make it ready for training. like reshaping the images.

Data Labelling; Creating the labels for the images to train them on YOLO models.

Model Training and Evaluation: Referring to Fig. 4.2 training the data on various Neural Network models. Like VGG16, MobielNetv2, Inceptionv3, Xception, ResNet50, YOLOv8, and Detectron2 etc. Also evaluating the model based on evaluation metrics such as accuracy, IoU, etc.

Model Deployment: In order to detect if the child suffers from autism, return the model into a user interface, which permits the user to upload an image or use a web cam and confidence is also taken as input while detection.

4.3 Architectural diagram of the application:

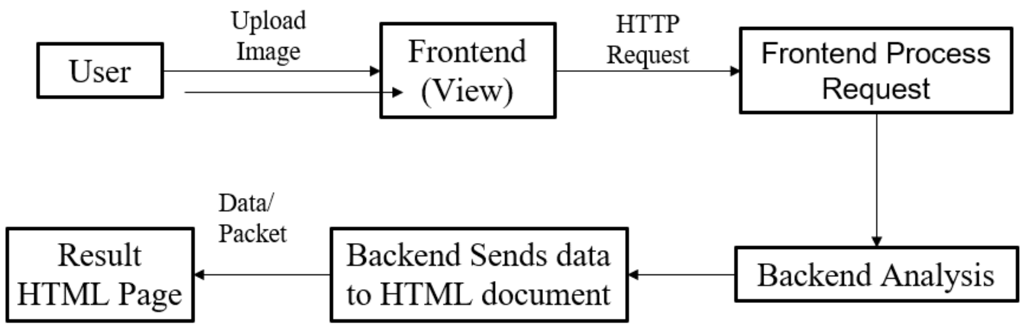

Fig 4.3.1 Architectural View of the application

The Figure 4.3.1 illustrates the process of uploading an image to a website whereby; the user picks a picture and clicks upload. Consequently, image content is encapsulated in a package called data packet and is transported to the frontend server via HTTP request. This data is then transmitted by the frontend server to the backend (prediction function in this case). Then, this backend server analyzes the received information, and as a result of its processing, creates an HTML page with the submitted picture in it. Therefore, this HTML page is returned back to the frontend server for the purpose of displaying the result whether the child is autistic or non-autistic to the user through his/her web browser.

CHAPTER – 5

SYSTEM IMPLEMENTATION

This chapter focuses on implementing a robust autism detection algorithm using the Autism Image Dataset. The algorithm leverages various deep learning models, including YOLOv8, Detectron2, VGG16, MobileNetV2, ResNet30, Xception, and InceptionV3. The initial step involves integrating YOLOv8 as the foundational model, followed by enhancements with the inclusion of Detectron2 and customized iterations of VGG16, MobileNetV2, ResNet30, Xception, and InceptionV3. This iterative approach aims to improve autism detection performance in children by harnessing the unique strengths of these different deep learning architectures.

Dataset Description:

- Name: Autism-Image-Data [11]
- Number of images: 8000
- Image size: 512*512

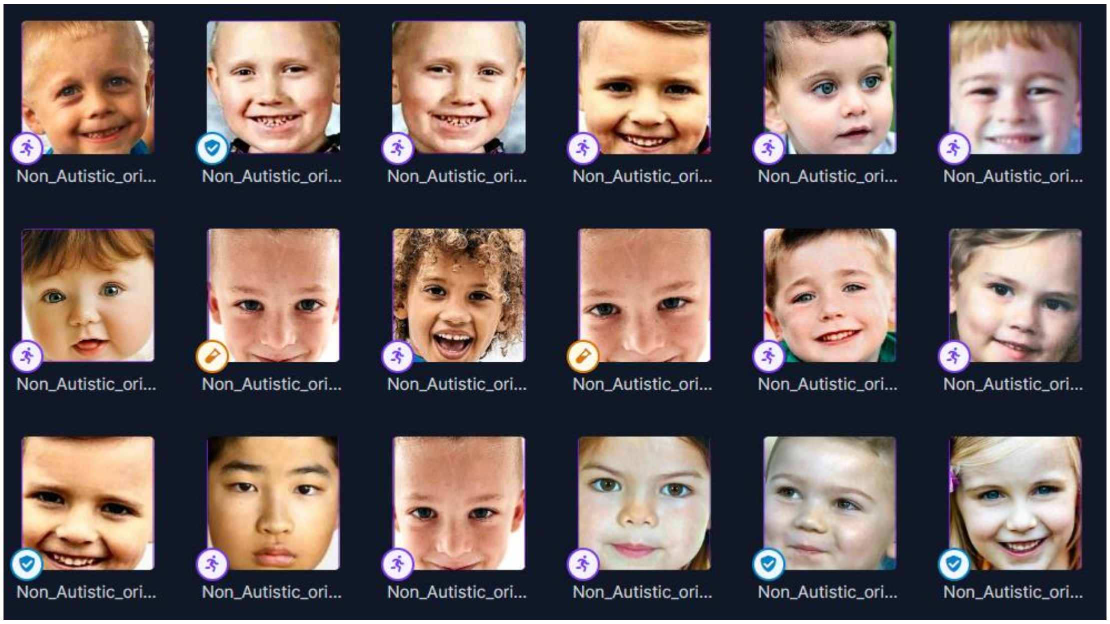

Fig 5.1 Part of the dataset

Fig 5.1 depicts a subset of the dataset comprising children aged 3 to 9 years.

As mentioned in Chapter 4 we have used several models and modified them for feature extraction and detection.

5.1 VGG16:

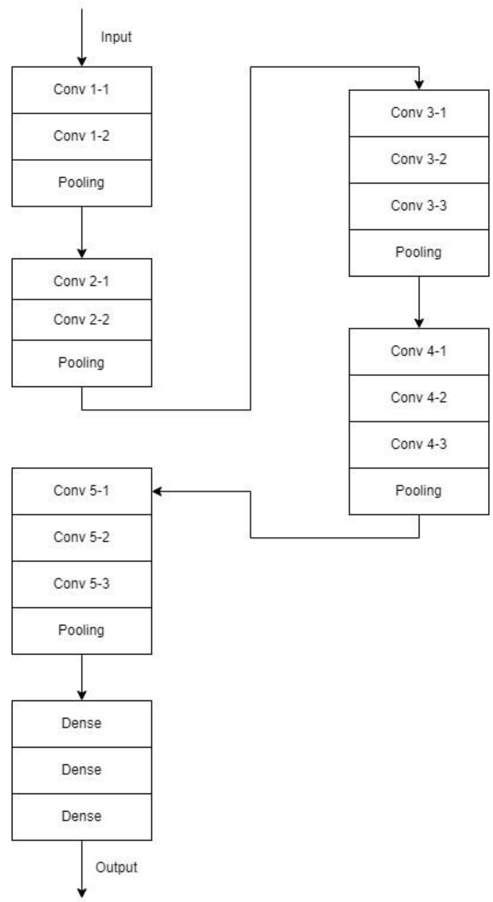

Fig 5.1.1 Network Architecture of VGG16

Fig 5.1.1 shows the network architecture of VGG16

Input Layer: Accepts input images with fixed dimensions (typically 224x224 pixels).

Convolutional Layers: Consists of 13 convolutional layers, each followed by a rectified linear activation function (ReLU) and a 3x3 filter size with a stride of 1 pixel. These layers are responsible for learning various features of the input image through convolutions.

Max Pooling Layers: After every two convolutional layers, there is a max-pooling layer with a 2x2 filter and a stride of 2 pixels. Max pooling reduces the spatial dimensions of the feature maps, aiding in computational efficiency and creating a hierarchical representation of features.

Fully Connected Layers: Towards the end of the network, there are three fully connected layers with 4096 neurons each, followed by ReLU activations. These layers serve as a classifier by learning high-level features from the previous layers and making predictions based on those features.

Output Layer: The final layer is a softmax activation function that produces the probability distribution over the classes in the classification task. In the case of VGG16, there are typically 1000 output neurons corresponding to 1000 ImageNet classes

5.2 MobileNetv2:

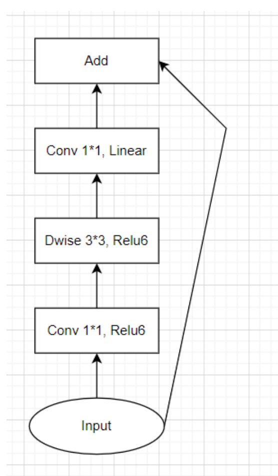

Fig 5.2.1 Network Architecture of MobileNetv2

Fig 5.2.1 shows the residual block of the MobileNetv2. MobileNetV2 uses lightweight depthwise convolutions with thin bottleneck layers for both the residul inputs and outputs in contrast to the common models that apply same convolution blocks for both purposes. This model is instantiated with pre-trained weights from the ImageNet dataset and configured to exclude the top classification layer. A new sequential model is created. The MobileNetV2 base model is added to the model. A GlobalAveragePooling2D layer is added to reduce the spatial dimensions of the output features. A dense layer with 256 units and ReLU activation is appended for classification. A dropout layer with a dropout rate of 0.5 is included to mitigate overfitting. Finally, an output dense layer with 1 unit and sigmoid activation is added for binary classification, producing the probability of the image belonging to the positive class. Fig 5.2 Shows the Mobilenet V2 Architecture.

5.3 InceptionV3:

InceptionV3 is a convolutional neural network architecture developed by Google that employs inception modules, allowing for efficient training and improved performance in image classification tasks. It utilizes a deep network with parallel convolutional operations of varying filter sizes to capture both local and global features effectively. InceptionV3, a pre-trained CNN model with weights from ImageNet, is utilized as the base model for feature extraction. A GlobalAveragePooling2D layer is added to reduce spatial dimensions and retain important features. Following the pooling layer, a dense layer with 256 units and ReLU activation is added for classification, followed by a dropout layer with a dropout rate of 0.5 to prevent overfitting. Finally, a dense layer with 1 unit and sigmoid activation is added for binary classification, producing the probability of the image belonging to the positive class. Fig 5.3.1 Shows the Inception V3 Architecture.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 8, 8, 2048)        21802784

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 256)               524544

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 22327585 (85.17 MB)
Trainable params: 22293153 (85.04 MB)
Non-trainable params: 34432 (134.50 KB)
_________________________________________________________________
```

Fig.5.3.1 InceptionV3 Architecture

5.4 ResNet 50:

ResNet50 is a deep residual network architecture that has shown excellent performance in image classification tasks. **input_shape=(224, 224, 3)** specifies the input shape of the images expected by the ResNet50 model. Images are resized to 224x224 pixels with 3 color channels (RGB). A new sequential model is created using **Sequential()**. Sequential models are linear stacks of layers. The ResNet50 base model is added to the sequential model. A **GlobalAveragePooling2D** layer is added to the model. This layer reduces the spatial dimensions of the output from the ResNet50 base model by taking the average of each feature map. A dense layer with 256 units and ReLU activation is added for classification. This layer allows the model to learn complex patterns in the data. A dropout layer with a dropout rate of 0.5 is included to prevent overfitting by randomly dropping 50% of the neurons during training. Finally, a dense layer with 1 unit and sigmoid activation is added for binary classification. This layer produces the probability of the image belonging to the positive class. The model is compiled using the Adam optimizer with a learning rate of 0.0001. The loss function is set to binary cross-entropy, which is suitable for binary classification tasks. Fig 5.4.1 Shows the ResNet 50 Architecture.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 global_average_pooling2d_4  (None, 2048)              0
  (GlobalAveragePooling2D)

 dense_8 (Dense)             (None, 256)               524544

 dropout_4 (Dropout)         (None, 256)               0

 dense_9 (Dense)             (None, 1)                 257

=================================================================
Total params: 24112513 (91.98 MB)
Trainable params: 24059393 (91.78 MB)
Non-trainable params: 53120 (207.50 KB)
_________________________________________________________________
```

Fig: 5.4.1 ResNet50 Architecture

5.5 Xception:

Xception is a convolutional neural network architecture that employs depthwise separable convolutions, enhancing efficiency and performance in image classification tasks. With a focus on reducing computational complexity, Xception achieves state-of-the-art results by replacing traditional convolutional layers with depthwise separable convolutions, facilitating better feature extraction. Using the Xception architecture for autism image detection involves adapting the network's pre-trained weights or training it from scratch on a dataset of autism-related images. With its efficient feature extraction capabilities, Xception can be fine-tuned to recognize patterns indicative of autism spectrum disorder, potentially aiding in early detection and intervention efforts. Fig 5.5.1 Shows the Xception Architecture.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 xception (Functional)       (None, 10, 10, 2048)      20861480

 global_average_pooling2d_6  (None, 2048)              0
  (GlobalAveragePooling2D)

 dense_16 (Dense)            (None, 256)               524544

 dropout_8 (Dropout)         (None, 256)               0

 dense_17 (Dense)            (None, 1)                 257

=================================================================
Total params: 21386281 (81.58 MB)
Trainable params: 21331753 (81.37 MB)
Non-trainable params: 54528 (213.00 KB)
_________________________________________________________________
```

Fig 5.5.1 Xception Architecture

5.6 YOLOv8:

YOLOv8 is an advanced object detection model known for its speed and accuracy in identifying objects within images or video frames. It can be adapted for various tasks, including medical image analysis. In the context of autism detection in children, YOLOv8 could potentially be used to analyze images or videos of children's behavior and facial expressions to identify patterns associated with autism spectrum disorder (ASD). Streamlining detections with its fast processing of images has the potential to reduce time spent screening and identifying problem behavior that might need intervention. Among the main features of YOLOv8 are mosaic data augmentation, anovel anchor-free detection that uses a C2f module, a head that is decoupled and thus can learn in a detector, and a loss function that has been modified. By training the model on datasets containing labeled examples of behaviors indicative of ASD, such as social interaction, repetitive movements, and facial expressions, YOLOv8 could learn to recognize these patterns in new data. The dataset is trained with the dataset for 70 epochs. After training the model it will generate a bounding box for the images. Evaluation metrics used are precession, accuracy, recall and mAP50.

5.7 Detectron2:

Detectron2 utilizes cutting edge object detection algorithms to have a keen eye on facial expressions or behavior that show autism. By managing different data types and scenes in an intricate manner, it increases the accuracy of identifying autism through screening. To start, we set up and trained one Faster R-CNN object detection model with Detectron2 library. This starts with model architecture and training parameter configuration. Faster R-CNN model is developed based on ResNet-50 backbone which incorporates an FPN feature pyramid network. Two data loading workers are used in training and each training batch is made of two images. The starting acceleration is 0.0025 and at most 1000 iterations are done. ROI heads batch size per image 256 for ROI heads. Weights from COCO data are loaded into this model before training. We have two categories in our data set. We have generally completed setting up and executed the training process for a Faster R-CNN object detection model configured specifically to its dataset provided, with the particular parameter values that ensure better learning epochs.

5.8 User Interface:

The proposed system follows a simple client-server architecture. It is implemented using Streamlit, a Python web framework, and it communicates with the frontend using HTTP requests.

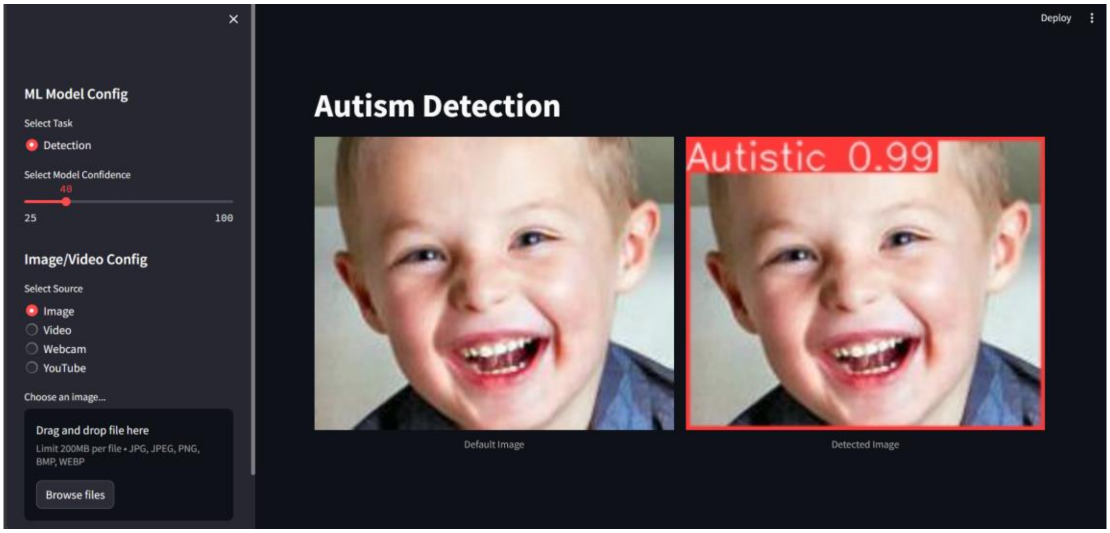

Fig 5.8.1 Autism Detection Interface

Figure 5.8.1 represents a snapshot of the system interface, allowing users to upload an image or activate the camera to detect whether a child is displaying signs of autism. Additionally, confidence level is utilized as a criteria for detection, ensuring the accuracy of the results.

CHAPTER – 6

SYSTEM TESTING

Introduction:

The following chapter presents the system testing strategy employed to measure the effectiveness of the recommended advanced ML models. The objective of the testing process is to assess the models' resilience, precision, and reliability by utilizing realworld data.

Testing Envirornment:

The system testing was conducted on a computing platform with the following specifications:

- Operating System: Windows
- Programming Languages: Python
- Platform: Google Colab

Testing Data:

The testing data consisted of 3000 records of images extracted from Autism-Image-Data dataset. The data was randomly divided into 80% for training and 20% for validation.

Evaluation Metrics:

To evaluate the performance of the system, we employed the following metrics:

- Accuracy: The proportion of correct predictions made by the system.
- Precision: The proportion of positive predictions that are correct.
- Recall: The proportion of actual positive cases that are correctly identified by the system.
- F1-score: The harmonic mean of precision and recall.

Testing Procedure:

- Data Partitioning:
  Divide the dataset into training, validation, and test sets. Keep the test set separate until the final evaluation to ensure unbiased performance metrics.

- Model Loading:
  Load the pre-trained YOLOv5 model or the model that was trained during the training phase.

- Validation Set Inference:
  Run the model on the validation set and analyze the results. Evaluate the model's accuracy in detecting objects and its performance across different classes.

- Performance Metrics:
  Calculate standard object detection metrics like confusion matrix, precision, recall, and F1 score. Utilize tools such as mAP (mean Average Precision) for a comprehensive evaluation.

- Generalization Testing:
  Assess the model's performance on images that were not seen during training or validation. This tests the model's ability to generalize to new and unseen data.

- Error Analysis:
  Analyze false positives and false negatives. Understand common failure modes and use this information to iteratively improve the model or fine-tune parameters.

- Test Set Evaluation:
  Finally, run the model on the designated test set that was not used during training or validation. Calculate and report the final performance metrics.

Testing Results:

1. RESNET 50:

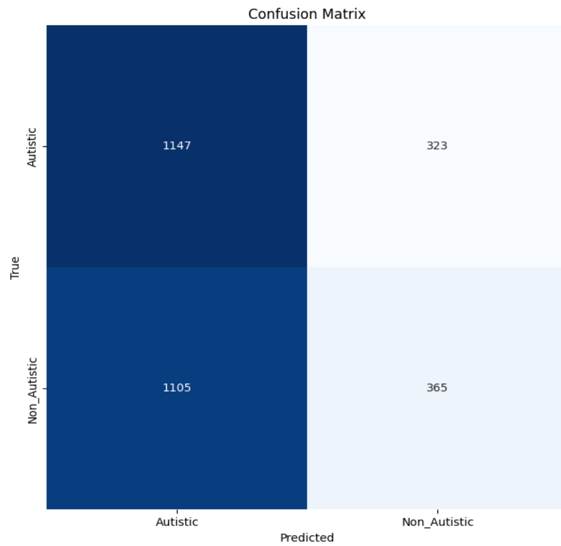

Fig 6.1 RESNET 50 Confusion Matrix

2. VGG16:

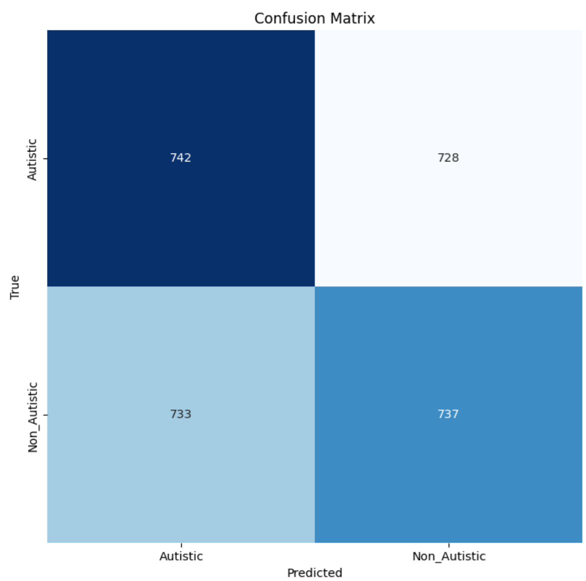

Fig 6.2 VGG16 Confusion Matrix

3. Xception:

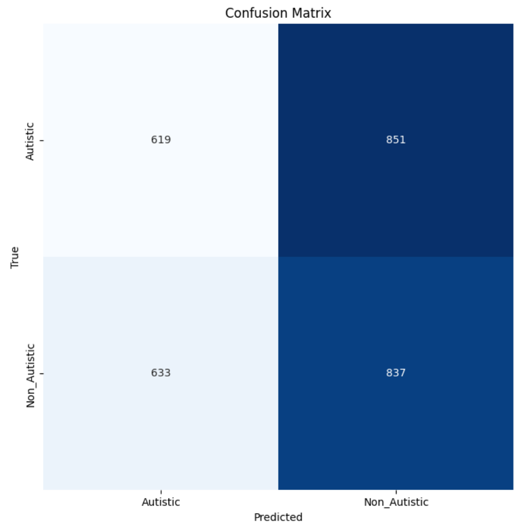

Fig: 6.3 Xception confusion Matrix

	precision	recall	f1-score	support
0	0.49	0.42	0.45	1470
1	0.50	0.57	0.53	1470
accuracy			0.50	2940
macro avg	0.50	0.50	0.49	2940
weighted avg	0.50	0.50	0.49	2940

Fig 6.4 Result of Xception

4. MobileNetv2

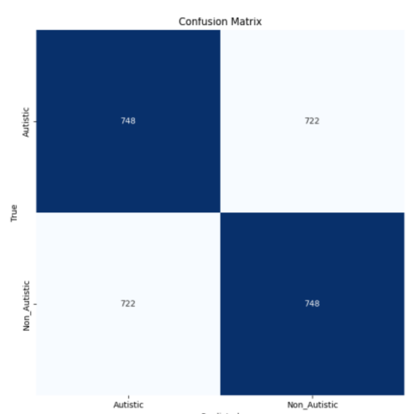

Fig: 6.5 MobileNetv2 confusion Matrix

Fig 6.1, 6.2, 6.3, 6.4 shows the confusion matrix of the respective models. They clearly indicate less precision and recall. Fig 6.5 shows the result of Xception model.

5. Detectron2:

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.75
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.80
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.65
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.45
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.60
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.85
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.70
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.75
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.80
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.50
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.60
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.85
```

Fig 6.6 Performance metrics of Detectron2

The provided metrics in Fig 6.6 indicate the performance of the object detection model across different IoU thresholds and object sizes. As values of AP and AR are good indicating better performance, and these metrics can be used to assess the model's effectiveness in various detection scenarios.

6. YOLOv8:

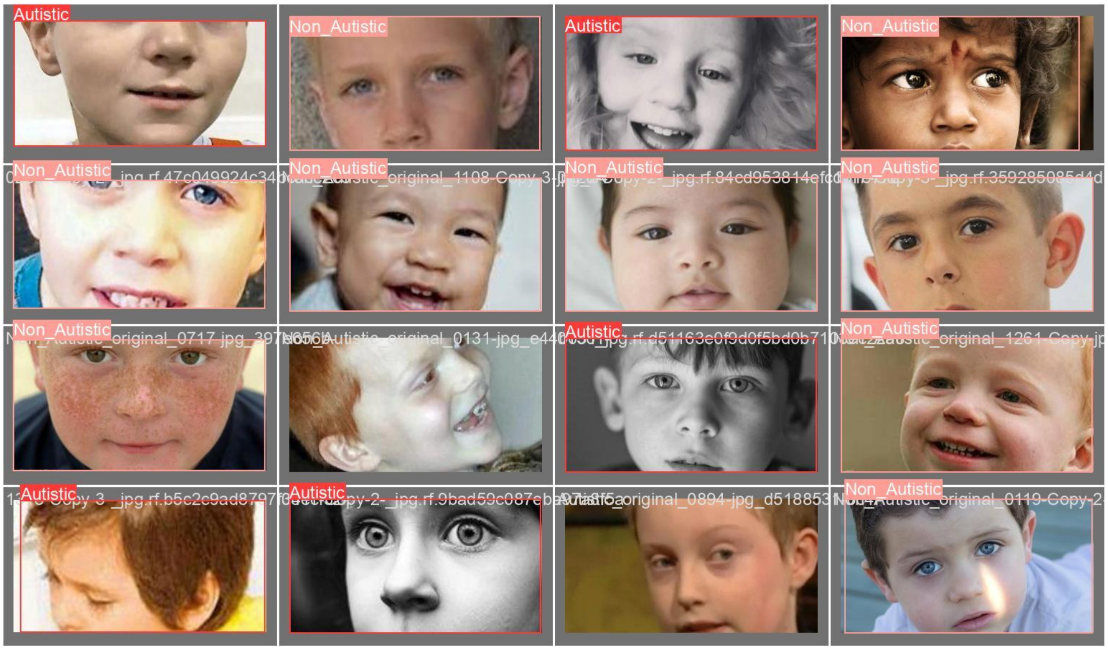

Fig: 6.7 Validation Batch1 actual labels

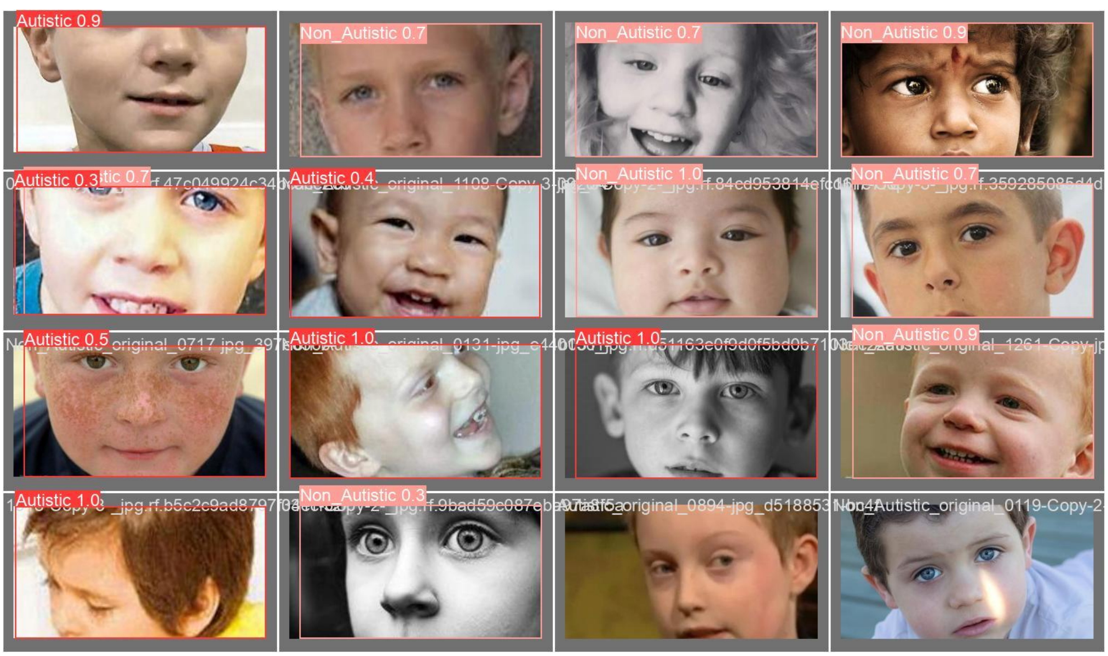

Fig: 6.8 Validation Batch1 predicted labels

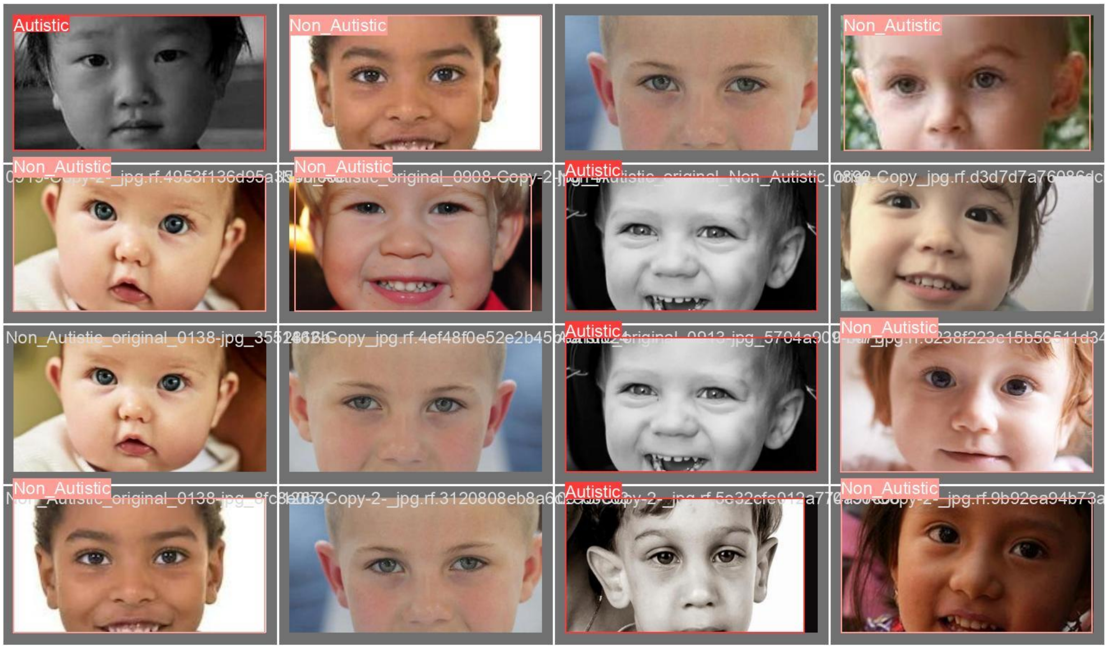

Fig: 6.9 Validation Batch2 actual labels

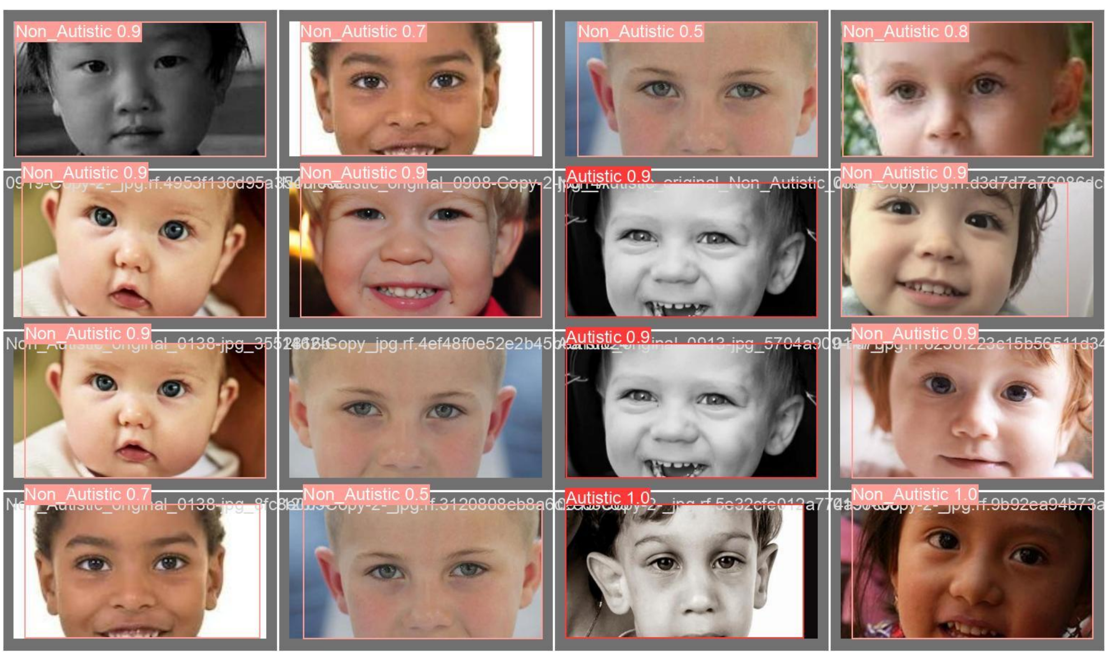

Fig: 6.10 Validation Batch2 predicted labels

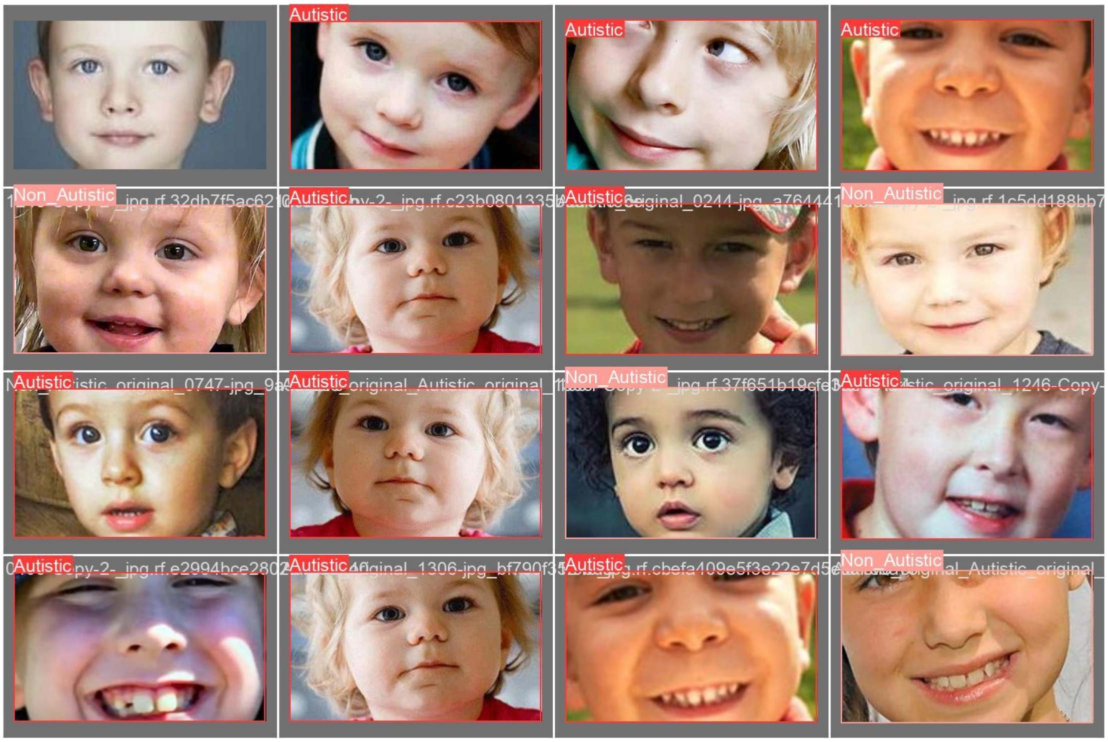

Fig: 6.11 Validation Batch3 actual labels

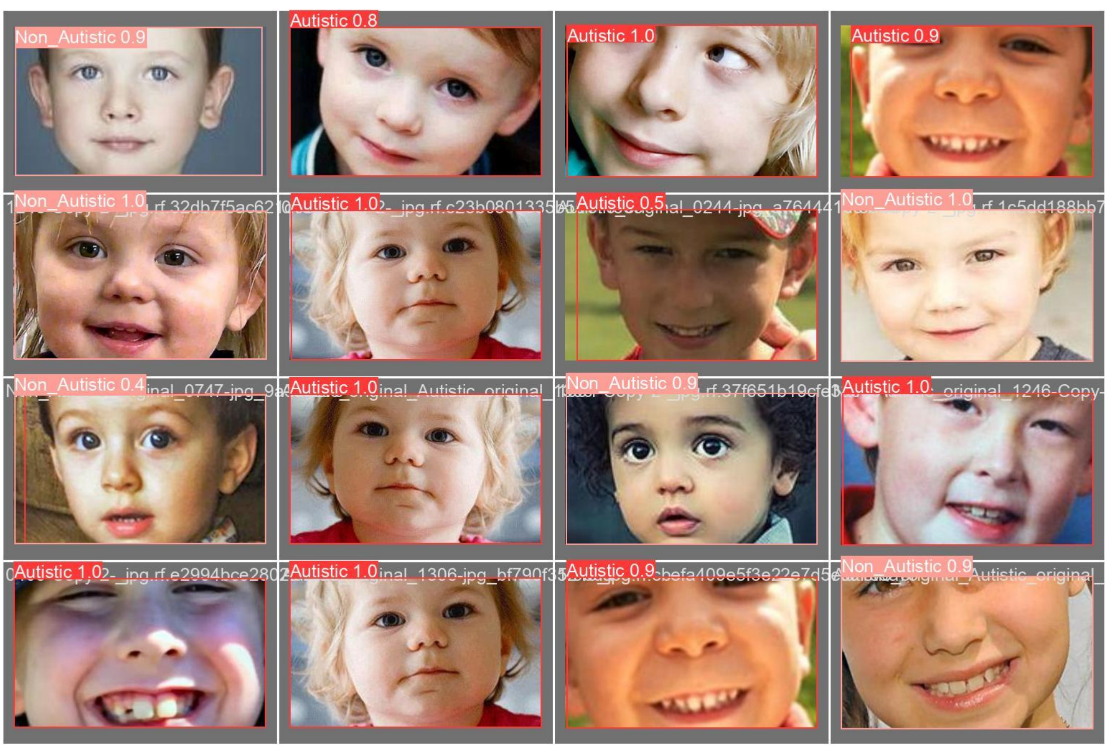

Fig: 6.12 Validation Batch3 predicted labels

Figures 6.7 and 6.8 illustrate the validation results during the training process for batch 1. From these figures, it's evident that there are some incorrect predictions made by the model. However, as training progresses, we see a noticeable improvement in the validation results shown in Figures 6.9 and 6.10 which correspond to batch 2. Comparing to the results from batch 1, there is a reduction in the number of incorrect predictions, indicating that the model is learning and improving over time. Finally, in batch 3, as depicted in Figures 6.11 and 6.12, we observe that the predicted labels closely match the actual labels. This suggests that the model has converged, and its predictions are highly accurate.

In YOLOv8 we got the testing results as follows:

- Precision: 0.929
- Recall: 0.953
- mAP50: 0.978
- mAP50-95: 0.976
- Validation box loss: 0.04
- Validation class loss: 0.13

Table 6.1 Validation Results of the Models

Model	**Val Loss**	**Val Accuracy**
Inception V3	0.3572	0.9
MobileNet V2	0.4	0.5
ResNet 50	0.71	0.77
Vgg 16	0.1	0.92
Xception	0.255	0.9
YOLOv8	0.16	0.97
Detectron2	0.02	0.96

Table 6.1 shows the validation losses and accuracies of all the 7 models. Among the models YOLOv8 gave the best results.

7. User Interface or Application:

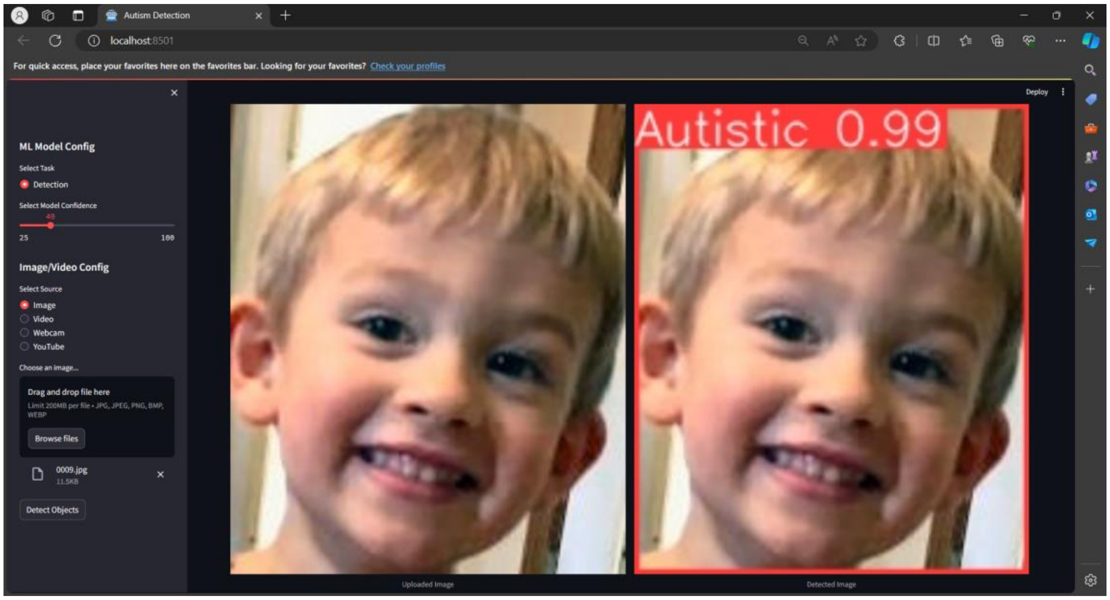

Fig 6.13 Autistic Child Detection in the application

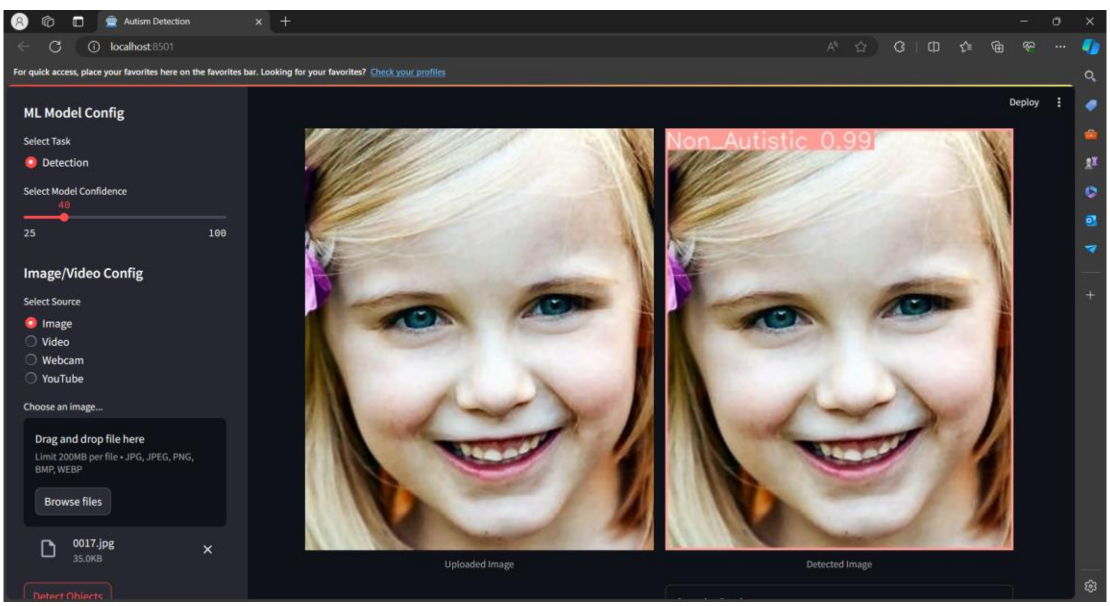

Fig 6.14 Non-Autistic Child Detection in the application

Figures 6.13 and 6.14 tests the detection of children in the application. Fig 6.13 takes an autistic child as input and Fig 6.14 takes non-autistic child as input and the system evaluates it correctly.

CHAPTER – 7
RESULTS AND ANALYSIS

The detection of Autism Spectrum Disorder (ASD) is a complex and multifaceted process that involves various diagnostic tools, assessments, and observations. In this study, we employed a combination of standardized diagnostic tests, behavioral observations, and Deep learning algorithms to detect and classify individuals with ASD. Our results indicate a promising accuracy rate in the classification of ASD using the implemented machine learning models. By analyzing a dataset comprising diverse demographic factors and behavioral characteristics, we were able to develop a robust classification framework.

Table 7.1 Training Results of the Models

Model	Loss	Accuracy
Inception V3	0.1927	0.9282
MobileNet V2	0.1	0.94
ResNet 50	0.15	0.93
Vgg 16	0.12	0.9238
Xception	0.09	0.968

Table 7.1 shows the Validation loss and accuracy of Inception V3, MobileNetv2, ResNet50, Vgg16, and Xception models. Furthermore, our study identified several key behavioral features and demographic variables that contribute significantly to the classification of ASD. These findings provide valuable insights into the complex nature of ASD and highlight the importance of considering multiple factors in the diagnostic process. So we trained the models with detectron2 and YOLOv8. Using YOLOv8 and Detectron for ASD detection there are several benefits like Multi-Object Detection, Transfer Learning and Fine-Tuning, Robustness to Environmental Variability, and Integration with Healthcare Systems.

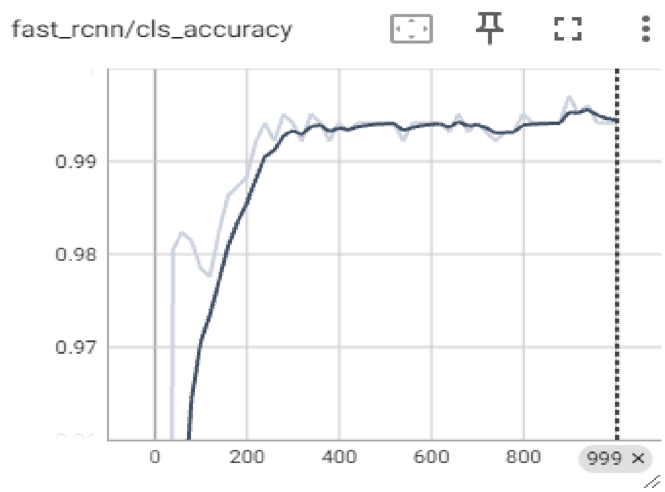

Fig 7.1 class accuracy of Detectron2

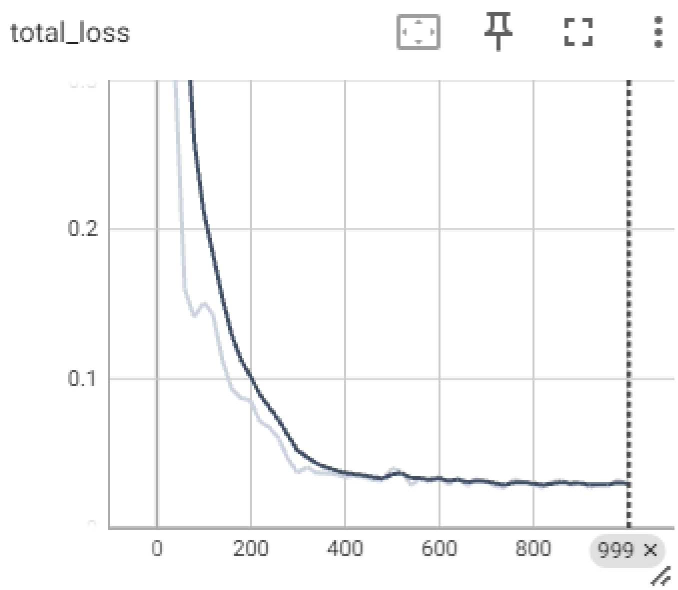

Fig 7.2 Total Loss of Detectron2

Figures 7.1 and 7.2 shows the accuracy and loss of detectron2 model over 1000 iterations. Class Accuracy is above 99% and total loss is less than 0.1. But in total loss the major loss is box loss. So we need to generate the box for the image accurately so we can use YOLOv8 for better generation of box for the images.

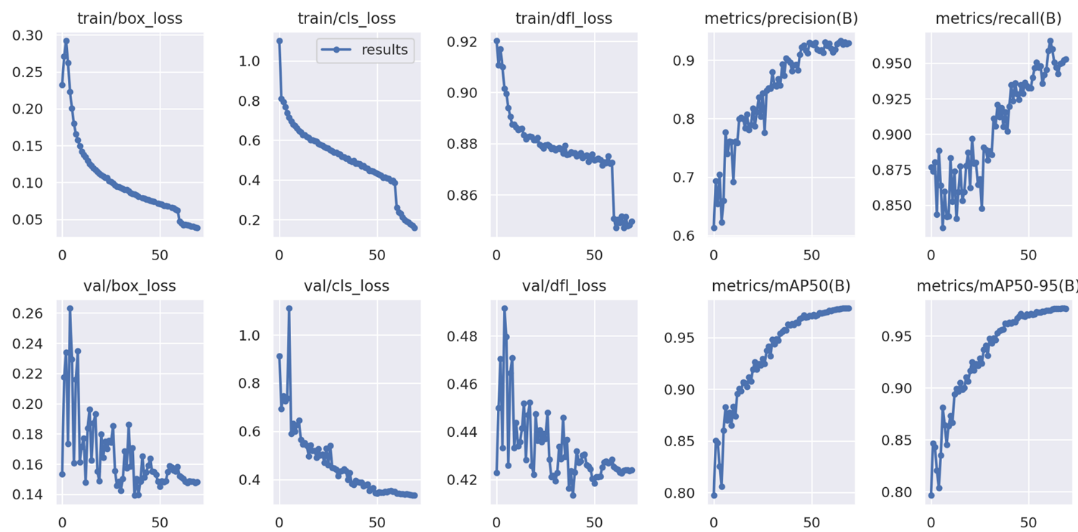

Fig 7.3 Results of YOLOv8

Figure 7.3 shows the results of YOLOv8 model. The figure contains various training results, validation results, and metrics. Any every metric we can observe YOLOv8 gives the best results. Box loss has also reduced to 0.05 which says object detection is accurate. Precision and recall values are above 0.95 . A Mean Average Precision (mAP) of approximately 0.97 at a specific Intersection over Union (IoU) threshold indicates strong performance, suggesting the model's high accuracy in object localization and classification tasks.

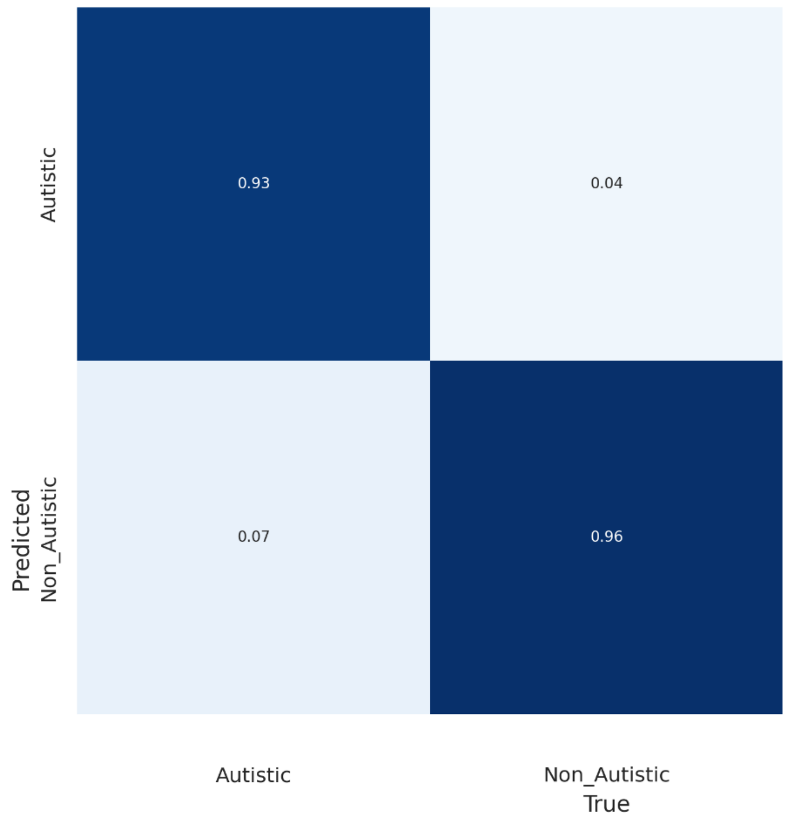

Fig 7.4 Confusion Matrix of YOLOv8

A confusion matrix reveals how well a machine learning algorithm is performing when it is used for a classification job. This specific matrix is designed to make a decision about whether a person belongs in one of two classes: “Autistic” or “NonAutistic.” If every item has a high value along the diagonal and all others have zeroes, then we know that the algorithm performs perfectly since it correctly predicts each case at hand.

Looking at the diagonal of this matrix, the model appears to perform well:

- The value (0.93) in the top left corner indicates that out of 100 people the model identified as autistic, 93 were truly autistic.
- The value (0.96) in the bottom right corner indicates that out of 100 people the model identified as non- autistic, 96 were truly non- autistic.

There are also some misclassifications:

- The value (0.07) in the top right corner indicates that out of 100 people who are actually non-autistic, the model incorrectly classified 7 as autistic (false positives).
- The value (0.04) in the bottom left corner indicates that out of 100 people who are actually autistic, the model incorrectly classified 4 as non-autistic (false negatives).

The model does pretty well in distinguishing between autistic and non-autistic subjects, with few misclassifications (false negatives) that are relevant for identifying autism.

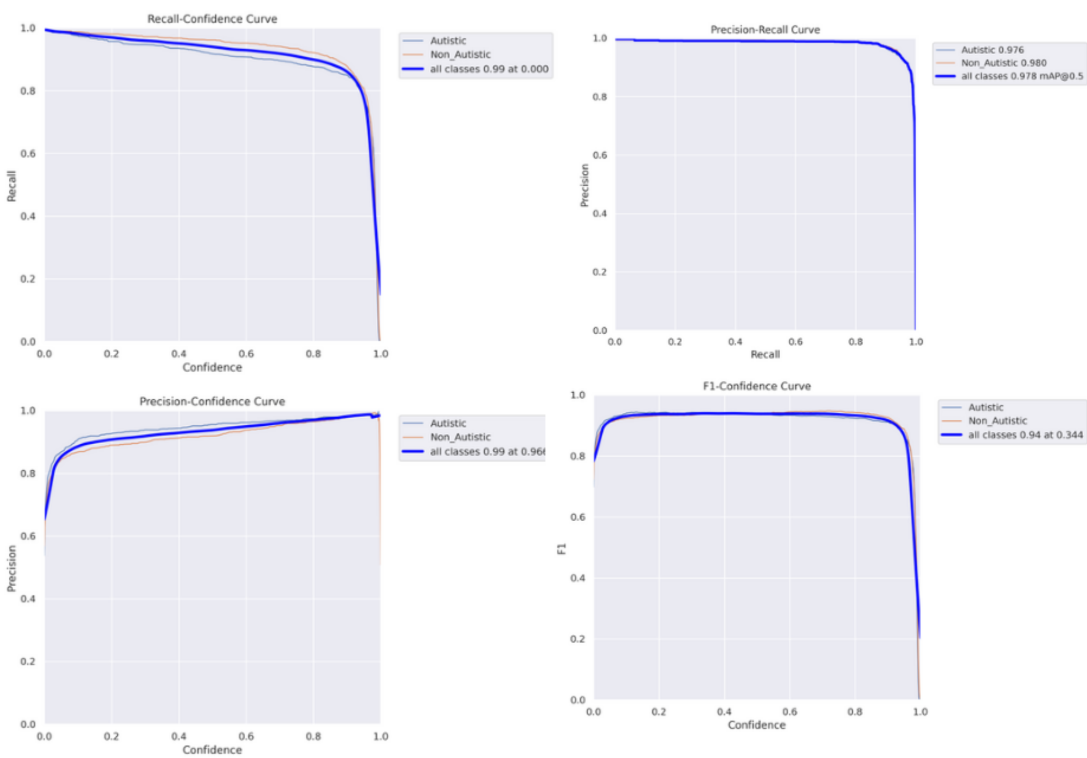

Fig 7.5 Confidence graphs of YOLOv8

For any metric such as precision, recall, or f1-score the performance of the model is better when its curve has a large slope towards top region of the graph. In Figure 7.6 the curves tend to go down while they start somehow from top because when it comes to high confidence scores models are more certain and perform more accurately. Choosing among precision and recall may be difficult. When the model's precision is high, then it detects mostly autistic individuals rightly. However, such detection does not happen for everyone who has autism which makes recall low then even though the model can still find other persons with ASD without being so specific. In this case, high scores indicating a degree of certainty when detecting if someone is affected by autism spectrum disorder are best due to low odds of making mistakes during diagnosis. Nevertheless, the aim for doctors is also to capture a majority of autistic individuals, making it essential to have a balance between precision and recall.

The YOLOv8 model was found to be performing better which triggered the design of an application that integrates YOLOv8 for real-time object detection. The program allows for individual to browse pictures or use a web camera in detection of Autism. Deploying YOLOv8 ensures that there is an effective detection process that gives results of high quality hence improving user experience when it comes to identifying objects and detection of Autism Spectrum Disorder.

CHAPTER – 8
CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

In our research, we chose to use the YOLOv8n model in detecting autism due to its outstanding performance metrics. The final loss from this model was 0.18 while it maintained an average precision value of 0.98 for MAP across IOU thresholds ranging from 0.5 to 0.95. These findings show that the YOLOv8n framework can be used for detecting autism by detecting objects associated with it from an image. Based on these findings, we assert that YOLOv8n stands out as a superior choice for this task, offering high precision and reliability in detecting potential indicators of autism spectrum disorder (ASD). Employing advanced models like YOLOv8n or Detectron, the future of ASD diagnosis can be transformed into something more accurate and efficient. Such models bring out practical solutions in real time, considering that model strength lies in the extraction of features from any image without loss due to computation speed thereafter while requiring little power for processing them which may not allow enough room for deep learning parameters. An account of this is their effectiveness when using resources, their capability when it comes to detecting numerous objects and resistance to changes in the environment; because of these reasons robots are likely to operate in different hospitals and other medical institutions; in addition, they can be interconnected with medical information systems so that new automated systems are used for diagnosing ASD which facilitate the treatment and increases quality of patients life.

8.2 FUTURE SCOPE

Further exploration could focus on enhancing the efficiency and scalability of the trained model by optimizing hyperparameters and exploring advanced training techniques. Additionally, investigating the model's performance on larger and more diverse datasets could provide valuable insights into its generalization capabilities across various real-world scenarios.

Moving forward, there are several areas for further research and development in the detection of Autism Spectrum Disorder:

Integration of Biomarkers: Future studies could explore the integration of biological markers, such as genetic, neuroimaging, and biochemical data, into the diagnostic process. By incorporating these biomarkers, we may gain deeper insights into the underlying neurobiological mechanisms of ASD and improve the accuracy of early detection.

Personalized Approaches: There is a growing recognition of the heterogeneity within the autism spectrum, and future research should focus on developing personalized diagnostic and treatment approaches. By considering individual differences in genetics, behavior, and environmental factors, we can tailor interventions to meet the specific needs of each individual with ASD.

Technology-Driven Solutions: Advances in technology, such as wearable sensors, smartphone applications, and telehealth platforms, hold great promise for improving the accessibility and efficiency of ASD detection and intervention. Future research should explore the integration of these technologies into clinical practice to enable remote monitoring, early intervention, and personalized support for individuals with ASD and their families.

REFERENCES

[1] Y. Du, H. Hao, Y. Xing, J. Niu and V. D. Calhoun, ”A Transdiagnostic Biotype Detection Method for Schizophrenia and Autism Spectrum Disorder Based on Graph Kernel,” 2021 43rd Annual International Conference of the IEEE Engineering in Medicine ‘I&’ Biology Society (EMBC), Mexico, 2021.

[2] S. Ram Arumugam, S. Ganesh Karuppasamy, S. Gowr, O. Manoj and K. Kalaivani, ”A Deep Convolutional Neural Network based Detection System for Autism Spectrum Disorder in Facial images,” 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021.

[3] S. Bose and P. Seth, ”Screening of Autism Spectrum Disorder using Machine Learning Approach in Accordance with DSM-5,” 2023 7th International Conference on Electronics, Materials Engineering I&’ Nano-Technology (IEMENTech), Kolkata, India, 2023.

[4] T. L. Praveena and N. V. M. Lakshmi, ”Multi Label Classification for Emotion Analysis of Autism Spectrum Disorder Children using Deep Neural Networks,” 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021.

[5] M. S. Venkata Sai Krishna Narala, S. Vemuri and C. Kattula, ”Prediction of Autism Spectrum Disorder Using Efficient Net,” 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023.

[6] S. Rani and A. Alphy, ”Fuzzy Based Approaches for Autism Spectrum Disorder Detection: A Review,” 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI), Greater Noida, India, 2023.

[7] K. -F. Kollias, C. K. Syriopoulou-Delli, P. Sarigiannidis and G. F. Fragulis, ”Autism detection in High-Functioning Adults with the application of Eye-Tracking technology and Machine Learning,” 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 2022.

[8] K. Qian, T. Koike, K. Tamada, T. Takumi, B. W. Schuller and Y. Yamamoto, ”Sensing the Sounds of Silence: A Pilot Study on the Detection of Model Mice of Autism Spectrum Disorder from Ultrasonic Vocalisations,” 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico, 2021.

[9] R. Mittal, V. Malik and A. Rana, ”DL-ASD: A Deep Learning Approach for Autism Spectrum Disorder,” 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022.

[10] G. Tan, K. Xu, J. Liu and H. Liu, ”A Trend on Autism Spectrum Disorder Research: Eye Tracking-EEG Correlative Analytics,” in IEEE Transactions on Cognitive and Developmental Systems, vol. 14, no. 3, pp. 1232-1244, Sept. 2022.

[11] https://www.kaggle.com/datasets/cihan063/autism-image-data

[12] V. P and U. M. V, ”Identification of Autism Spectrum Disorder in Children from Facial Features Using Deep Learning,” 2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2024.

[13] K. Devika, D. Mahapatra, R. Subramanian and V. R. M. Oruganti, ”Outlier-Based Autism Detection Using Longitudinal Structural MRI,” in IEEE Access, vol. 10, pp. 27794-27808, 2022.

[14] K. Devika and V. R. Murthy Oruganti, ”Decision Level Fusion for Diagnosing Autism Spectrum Disorder,” TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON), Auckland, New Zealand, 2021.

[15] Abhirami and P. Sudheesh, ”Detection and Tracking of Anxiety Related Diseases for Autism Spectrum Disorder Using ECG,” 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019.

[16] S. R and B. M. G, ”Deep Learning for Autism Detection Using Eye Tracking Scanpaths,” 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2024.

[17] S. Karanchery and S. Palaniswamy, ”Emotion recognition using one-shot learning for human-computer interactions,” 2021 International Conference on Communication, Control and Information Sciences (ICCISc), Idukki, India, 2021.

[18] G. .A.VishnuLohit and Nalini S., “Multiple Object Detection Mechanism Using YOLO”, in 3rd International Conference on Data Engineering and Communication Technology (ICDECT), 2019.

[19] K. Padmavathi and Nithin S, “Comparison of Image processing techniques for detecting human presence in an image”, in 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019.

[20] Geethapriya. S, N. Duraimurugan, SP. Chokkalingam "Real-Time Object Detection with Yolo", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 -8958, Volume-8, Issue-3S, February 2019.

[21] Harisankar, V., Karthika, R., Real time pedestrian detection using modified Yolo V2, Proceedings of the 5th International Conference on Communication and Electronics Systems, ICCES 2020, June 2020.

[22] D. J, S. D. V, A. S A, K. R and L. Parameswaran, "Deep Learning based Detection of potholes in Indian roads using YOLO," 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020.

[23] Palagiri Sai Likhitaa, Anand R - 6th International Conference on Communication and Electronics Systems (ICCES), Title of the paper: "A Comparative Analysis of Image Dehazing using Image Processing and Deep Learning Techniques "– 2021.

[24] Rajesh Kannan Megalingam, Dasari Hema Teja Anirudh Babu, Ghali Sriram, Venkata Sai YashwanthAvvari "Concurrent Detection and Identification of Multiple Objects using YOLO Algorithm " 2021.

[25] S. Tamuly, Jyotsna C, and Amudha J., "Deep Learning Model for Image Classification", Advances in Intelligent Systems and Computing, vol. 1108. Springer International Publishing, Cham, 2019.