International Journal of
Electrical and Computer System Design
ISSN: 2582-8134 www.ijecsd.com

DETECTION OF AUTISM SPECTRUM DISORDER USING DEEP LEARNING

[1]B.Sivadharshini, [2]Dr. P D Sheba Kezia Malarchelvi

*[1]PG Student, [2]Professor

Department of Computer Science & Engineering, Saranathan College of Engineering, Trichy, India.

Abstract: A neurodevelopmental condition known as autism spectrum disorder (ASD) impacts social interaction, behaviour, and cognitive functioning. ASD diagnosis can be challenging and time-consuming, but early identification and intervention can improve long-term outcomes. Autism spectrum disorder begins in early childhood and eventually causes problems functioning in society — socially, in school and at work, for example. Often children show symptoms of autism within the first year. Some children show signs of autism spectrum disorder in early infancy, such as reduced eye contact, lack of response to their name or indifference to caregivers. In this project. Deep Learning algorithm such as LSTM to determine the presence of disorder at an early stage. The datasets have been collected using Self-Stimulatory Behaviours Dataset (SSBD) and video dataset is used for designing the system. Blaze Pose algorithm is used for feature extraction. The datasets will be trained using deep learning algorithm and the model file has been generated. When an input image is given for disease prediction, as a result its severity will be calculated. The proposed system provides an effective solution to predict the presence of autism spectrum disorder in a more efficient way using the activity of the children.

Keywords: Autism Spectrum Disorder; Long-Short Term Memory; Blaze Pose; Batch Normalization; ReLu; Softmax

1. Introduction

Autism spectrum disorder is a condition related to brain development that impacts how a person perceives and socializes with others, causing problems in social interaction and communication. The disorder also includes limited and repetitive patterns of behavior. The term "spectrum" in autism spectrum disorder refers to the wide range of symptoms and severity. Autism spectrum disorder includes conditions that were previously considered separate — autism, Asperger's syndrome, childhood disintegrative disorder and an unspecified form of pervasive developmental disorder. Some people still use the term "Asperger's syndrome," which is generally thought to be at the mild end of autism spectrum disorder. Autism spectrum disorder begins in early childhood and eventually causes problems functioning in society socially, in school and at work, for example. Often children show symptoms of autism within the first year. A small number of children appear to develop normally in the first year, and then go through a period of regression between 18 and 24 months of age when they develop autism symptoms. While there is no cure for autism spectrum disorder, intensive, early treatment can make a big difference in the lives of many children.[1] A child or adult with autism spectrum disorder may have problems with social interaction and communication skills, including any of these signs: Fails to respond to his or her name or appears not to hear you at times, Resists cuddling and holding, and seems to prefer playing alone, retreating into his or her own world, Has poor eye contact and lacks facial expression, Doesn't speak or has delayed speech, or loses previous ability to say words or sentences, Can't start a conversation or keep one going, or only starts one to make requests or label items, Speaks with an abnormal tone or rhythm and may use a singsong voice or robot-like speech, Repeats words or phrases verbatim, but doesn't understand how to use them, Doesn't appear to understand simple questions or directions, Doesn't express emotions or feelings and appears unaware of others' feelings, Doesn't point at or bring objects to share interest, Inappropriately approaches a social interaction by being passive, aggressive or disruptive, Has difficulty recognizing nonverbal cues, such as interpreting other people's facial expressions, body postures or tone of voice.[2]

Distinguishing between the activities and behaviors of typically developing children and those with autism can help identify potential indicators of autism. While every child is unique, the following are some general differences that may be observed:

1. Social Interaction: Typically developing children tend to naturally seek social interaction and engage in reciprocal play. They initiate conversations, maintain eye contact, and respond appropriately to social cues.

In contrast, children with autism may display difficulties in social interaction. They may have limited interest in engaging with peers, struggle with reciprocal play, exhibit challenges in maintaining eye contact, and demonstrate reduced responsiveness to social cues.

2. Communication: Typically developing children develop language skills progressively, using gestures, babbling, and eventually speaking words and sentences to communicate. They also understand and respond to verbal and nonverbal cues. In contrast, children with autism may experience delays or differences in language development. They may have difficulties initiating and maintaining conversations, understanding non-literal language (such as sarcasm), or displaying limited or repetitive language patterns.

3. Repetitive Behaviors: Children with autism often exhibit repetitive or stereotyped behaviors. These may include repetitive movements (e.g., hand flapping, rocking), fixation on specific objects or topics, rigid adherence to routines, or intense interests in specific subjects. Typically developing children may engage in repetitive behaviors to a certain extent but not to the same degree on with the same intensity as children with autism

4. Sensory Sensitivities: Children with autism may display heightened or diminished responses to sensory stimuli. They may be overly sensitive to sounds, textures, or lights, leading to sensory overload or meltdowns. On the other hand, they may also seek sensory input, engaging in repetitive behaviors such as spinning or tapping objects to self-regulate. Typically developing children may exhibit normal sensory responses, although they may also have preferences or aversions to certain sensory stimuli.

5. Play Skills: Typically developing children engage in imaginative play, create narratives, and demonstrate flexible play behaviors. They can adapt their play based on the social context and interact with peers in cooperative play. Children with autism may exhibit challenges in imaginative play, struggle with pretend play scenarios or show limited flexibility in their play behaviors. They may engage in repetitive or ritualistic play patterns or prefer solitary play.

2. LITERATURE SURVEY

2.1 Functional connectivity-based prediction of Autism on site harmonized ABIDE dataset

“Madhura Ingalhalikar, Sumeet Shinde” proposed a system using larger sample sizes available from multi-site publicly available neuroimaging data repositories makes machine-learning based diagnostic classification of mental disorders more feasible by alleviating the curse of dimensionality.[3] However, since multi-site data are aggregated post-hoc, i.e. they were acquired from different scanners with different acquisition parameters, non-neural inter-site variability may mask inter-group differences that are at least in part neural in origin. Hence, the advantages gained by the larger sample size in the context of machine-learning based diagnostic classification may not be realized. Methods: Address this issue using harmonization of multi-site neuroimaging data using the ComBat technique, which is based on an empirical Bayes formulation to remove inter-site differences in data distributions, to improve diagnostic classification accuracy. Specifically, demonstrate this using ABIDE (Autism Brain Imaging Data Exchange) multisite data for classifying individuals with Autism from healthy controls using resting state fMRI-based functional connectivity data. Results: results show that higher classification accuracies across multiple classification models can be obtained (especially for models based on artificial neural networks) from multi-site data post harmonization with the ComBat technique as compared to without harmonization, outperforming earlier results from existing studies using ABIDE. Furthermore, the network ablation analysis facilitated important insights into autism spectrum disorder pathology and the connectivity in networks shown to be important for classification covaried with verbal communication impairments in Autism.

Pros: ComBat has the potential to make AI-based clinical decisionsupport systems more feasible in psychiatry.

Cons: There is a drop in accuracy when any of the given sub-networks are occluded from the analysis. Only the characteristic path-length of the auditory subnetwork showed a significant positive correlation with ADIR verbal scores in the Autism group.

2.2 Nonlinear Causal Relationship Estimation by Artificial Neural Network; applied for autism connectivity study

“Nasibeh Talebi, Ali Motie Nasrabadi, Iman Mohammad-Rezazadeh,” developed a system by detecting Quantifying causal (effective) interactions between different brain regions is very important in neuroscience research. Many of conventional methods estimate effective connectivity based on linear models. However, using linear connectivity models may oversimplify functions and dynamics of the brain. In the present study, propose a causal relationship estimator called “nCREANN” (nonlinear Causal Relationship

Estimation by Artificial Neural Network) that identifies both linear and nonlinear components of effective connectivity in the brain. Furthermore, it can distinguish between these two types of connectivity components by calculating the linear and nonlinear parts of the network input-output mapping. The nCREANN performance has been verified using synthesized data and then it has been applied on EEG data collected during rest in children with autism spectrum disorder (ASD) and Typically Developing (TD) children. Results show that overall linear connectivity in TD subjects is higher, while the nonlinear connectivity component is more dominant in ASDs. Suggest that the findings may represent different underlying neural activation dynamics in ASD and TD subjects. The results of nCREANN may provide new insight into the connectivity between the interactive brain regions.

Pros: Results show that overall linear connectivity in TD subjects is higher, while the nonlinear connectivity component is more dominant in ASDs.

Cons: Using overlapping sliding windows may increase computational cost.

2.3 Towards Automatic Anxiety Detection in Autism: A Real-Time Algorithm for Detecting Physiological Arousal in the Presence of Motion

“Akshay Puli, Azadeh Kushki(2019)”, proposed system for detecting autism spectrum disorder (ASD) due to its negative impact on physical and psychological health. Treatment of anxiety in ASD remains a challenge due to difficulties with self-awareness and communication of anxiety symptoms. To reduce these barriers to treatment, physiological markers of autonomic arousal, collected through wearable sensors, have been proposed as real-time, objective, and language-free measures of anxiety. A critical limitation of existing anxiety detection systems is that physiological arousal is not specific to anxiety and can occur with other user states such as physical activity. This can result in false positives which can hinder the operation of these systems in real-world situations[4]. The objective of this work was to address this challenge by proposing an approach for real-time detection and mitigation of physical activity effects. Methods: A novel multiple model Kalman-like filter is proposed to integrate heart rate and accelerometry signals. The filter tracks user heart rate under different motion assumptions, and chooses the appropriate model for anxiety detection based on user motion conditions. Results: Evaluation of the algorithm using data from a sample of children with ASD shows a significant reduction in false positives compared to the state-of-the-art, and an overall arousal detection accuracy of 93%.

Pros: The proposed method is able to reduce false detections due to user motion, and effectively detect arousal states during movement periods.

Cons: Several other mental, cognitive, and affective processes can also impact cardiac activity and lead to false positives.

2.4 Design of an Intelligent Agent to Measure Collaboration and Verbal-Communication Skills of Children with Autism Spectrum Disorder in Collaborative Puzzle Games

“Lian Zhang, Ashwaq Z.Amat”, proposed this system to detect autism using Collaborative puzzle games. These games are interactive activities that can be played to foster the collaboration and verbal-communication skills of children with ASD. In this paper, Designed an intelligent agent that can play collaborative puzzle games with children and verbally communicate with them as if it is another human player. Furthermore, this intelligent agent is also able to automatically measure children’s task-performance and verbal-communication behaviors throughout game play. Two preliminary studies were conducted with children with ASD to evaluate the feasibility and performance of the intelligent agent. Results of Study I demonstrated the intelligent agent’s ability to play games and communicate with children within the game-playing domain. Results of Study II indicated its potential to measure the communication and collaboration skills of human users.

Pros: This intelligent agent is also able to automatically measure children’s task performance and verbal-communication behaviors throughout game play.

Cons: This system cannot provide information unless the required information has already been stored in the system.

2.5 Automated Autism Detection based on Characterizing Observable Patterns from Photos

“Alice Z.Guo”, proposed this system Autism Detect autism using Photos. People with ASD show atypical attentions to social stimuli and gaze at human faces and complex scenes in an unusual way, and their facial expressions are often atypical as well. This work investigates the feasibility of developing an automated method to analyze the visual cues of autism using the photos taken by people with ASD, comparing to photos taken by people without ASD, in different scenarios. It was inspired by a recent study based on manual inspection of the photos. The key challenge is what and how to characterize the photos taken by people with ASD, to facilitate an automated separation from normal people. Several features are proposed to characterize the observable behaviors for ASD with experimental validations. This is the first work to perform an automatic analysis of the photos taken by people with ASD, achieving a prediction accuracy of 85.8%.

Pros: Several features are proposed to characterize the observable behaviors for ASD with experimental validations.

Cons: The lack of explanation problem may raise doubts to the patients, since it is difficult to persuade or convince them without explanations that they are diagnosed with the ASD.

Problem identification and statement:

The Autism detection employed by the existing systems have their own weaknesses. EEG and ET modalities should not be available at all the time and which requires much more computational costs. The eigen values is very difficult to find out.nCREANN is able to follow slow dynamics, and this may increase computational cost.supervised approaches may not always be possible in everyday situations where only limited training data are available. KNN motor task would not work for children less than 2 years old.[5] Using Deep neural network autism is detecting by speech, this will not be possible for the children who are all unable to speak and dumb children. GAN require a large amount of training data in order to produce good results.

To overcome from all these problems, this project work helps to detect the autism Using the video in Deep Learning. To pre processing the image will be initially resize the image. To extract the feature Blaze Pose is used. Design a suitable deep learning model architecture for autism detection, incorporating the extracted features. Explore various architectures that can handle sequential data, such as LSTM (Long Short-Term Memory) networks. Experiment with different layers, activation functions, regularization techniques, and fusion strategies to effectively utilize the extracted Blaze Pose features. Combine the values obtained from all frames in the video using an appropriate method (e.g., averaging or max pooling). The SoftMax activation function is commonly used in classification tasks to convert a set of raw predictions into probabilities. This final probability represents the likelihood of the video indicating autism.

3. Proposed System

DATASET COLLECTION: The Self-Stimulatory Behaviours Dataset refers to a specific dataset that captures and documents self-stimulatory behaviours in individuals. The outline of the data collection process for a self-stimulatory behaviours dataset:

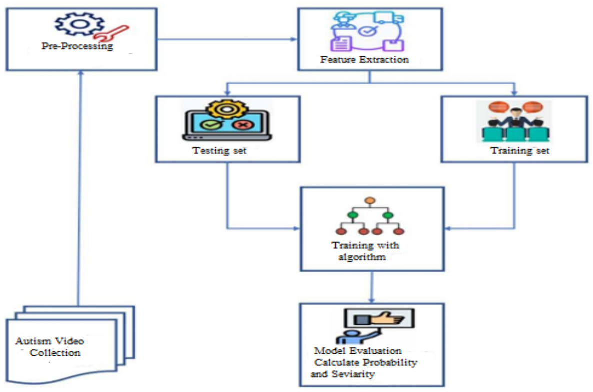


Fig.1 *Block Diagram*

1.Ethics and Consent:

Ensure compliance with ethical guidelines and obtain necessary approvals for data collection involving human subjects. Obtain informed consent from individuals or legal guardians, particularly if the dataset includes personally identifiable information.

2.Participant Selection:

Identify and recruit participants who exhibit self-stimulatory behaviours. Consider diverse demographics (age, gender, etc.) to capture a representative sample.

3.Data Recording:

Set up a controlled environment for data collection, such as a laboratory or naturalistic settings. Use appropriate recording devices, such as video cameras, to capture the behaviours. Ensure adequate lighting and audio quality for accurate behaviour capture.

4.Behaviour Observation:

Train data collectors to recognize and document self-stimulatory behaviours accurately. Employ standardized behaviour coding schemes or observation protocols for consistency. Observe and record behaviours during specified time intervals or continuously, depending on the research objectives.

5.Annotation and Labeling:

Review the recorded data and annotate the self-stimulatory behaviours of interest. Develop a labeling scheme that captures relevant information about the behaviours, such as duration, intensity, or specific characteristics. Multiple annotators can be involved to ensure inter-rater reliability.

This repository contains scripts to clip Self-Stimulatory Behaviours Dataset (SSBD) dataset. First, download the SSBD dataset release from https://rolandgoecke.net/research/datasets/ssbd/.

DATASET PRE-PROCESSING:

The OpenCV library is used to extract frames from video files in a directory and resize them to a desired output size. The resized frames are then saved as JPEG images in another directory.

First imports the necessary libraries, OpenCV and OS. It then defines the paths to the directory containing the video files, the output directory where the resized images will be saved, and the desired output size of the images.

Then enters a loop that iterates over all the video files. For each video file, the script loads the video file using OpenCV and extracts each frame of the video as a resized image. The script uses a while loop to read each frame of the video until there are no more frames to read. For each frame, the script resizes the frame to the desired output size using the cv2.resize() function and saves the resized frame as a JPEG image in the output directory using the cv2.imwrite() function.[6] Finally, releases the video capture object and moves on to the next video file.

FEATURE EXTRACTION:

In this project, BlazePose algorithm is used for feature extraction.

BlazePose (Full Body) is a pose detection model developed by Google that can compute (x,y,z) coordinates of 33 skeleton keypoints. It can be used for example in fitness applications. BlazePose has a Detector. The Detector cuts out the human region from the input image, BlazePose outputs the 33 keypoints according the following ordering convention.

Angle between Three Joints:

Angle = arccos((y^2 + z^2 - x^2) / (2 * y * z))

Calculation: Calculate the angle between three pose landmarks representing three body joints, using the law of cosines. 'x', 'y', and 'z' represent the lengths of the sides opposite to the corresponding joints.

1. Pose Detection:

Use a pose detection model to detect and localize human poses in each video frame. The output of the pose detector is typically a set of pose landmarks or key points representing the positions of specific body joints or body parts.

2. Pose Landmark Extraction:

Extract the pose landmarks from the output of the pose detection model. Each pose landmark represents the coordinates or positions of a specific body joint or body part.

3. Feature Calculation:

Calculate various features based on the extracted pose landmarks to represent different aspects of the body posture or movement patterns. These features can include angles between joints, ratios of body part lengths, velocities of body joints, or any other relevant measures.

4. Feature Representation:

Represent the extracted features of each video frames will be converted in text file. And save it as the csv file.

The Pose Detector method uses the Media pipe library to detect human poses in images or video frames. Pose Detector has three arguments: complexity, detection, and track. The complexity argument specifies the complexity of the pose estimation model, and the detection and track arguments specify the confidence thresholds for pose detection and pose tracking, respectively.

The pose Detector has two main methods: findPose and findPosition.

The findPose method takes an image as input and returns the image with the pose landmarks drawn on it. The findPosition method takes an image as input and returns a list of the x and y coordinates of the pose landmarks.

The findPose method first converts the input image from BGR format to RGB format, which is the format expected by the Media pipe library. If the pose landmarks are detected, the method uses the Draw method to draw the landmarks on the image. Finally, the method returns the image with the pose landmarks drawn on it.

The findPosition method first initializes an empty list to store the x and y coordinates of the pose landmarks. It then checks if the pose landmarks were detected in the image. If the landmarks were detected, the method iterates through the

landmarks and calculates the x and y coordinates of each landmark relative to the size of the image. The x and y coordinates are then added to the list. If the draw parameter is set to True, the method draws a small circle at each landmark position on the input image. Append the coordinates in the list Finally, the method returns the list of x and y coordinates of the pose landmarks.

This will iterates for each frame to read video file using the cap.read() function. If the function returns False, indicating that there are no more frames to read, the loop is exited. This will iterate through the list of pose coordinates and writes them to the text file using the write method of the file object. The loop then writes a newline character to the text file to separate the pose coordinates for each frame. The to _csv () function is used to save the data frame as a CSV file.

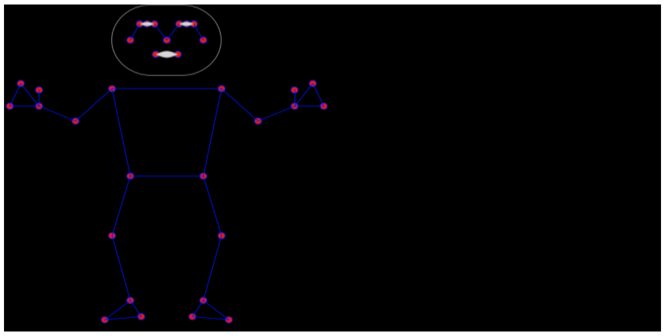


Fig. 2 BlazePose Diagram

Then reads data from three CSV files named using the read function from the pandas library. The resulting data frames are assigned to the variables. And then concatenate the three data frames vertically (i.e., along the rows) into a single data frame. The resulting data frame is assigned to the variable data.

The dictionary is used to maps target strings to labels. The keys of the dictionary are strings that represent the target values, and the values are integers that represent the corresponding labels. For example, the string "arm flapping" is mapped to the integer 0, "headbanging" is mapped to 1, and "spinning" is mapped to 2.

The decode resolution function takes a label as input and returns its corresponding value in the decode map dictionary. The input label is first converted to a string using the str function to ensure that it matches the keys of the dictionary. The function then returns the value associated with the key in the dictionary.

Then reads a dataset and splits it into features and labels using pandas' iloc method. The 'iloc' method is used to select rows and columns by integer position. In this case, : is used to select all rows, and :-1 is used to select all columns except the last one, which is assumed to be the label column. The resulting X variable contains the feature data, and the Y variable contains the label data.

Shapes of the feature and label arrays return using the shape attribute of numpy arrays. For example, if there are three classes, the label [2] would be converted to the vector [0, 0, 1].

TRAINING WITH ALGORITHM:

After the feature extraction process, these features are given to deep learning algorithm which will undergo training of the features and output accuracy. In this project, LSTM is used to train the datasets. Long short-term memory (LSTM) is an artificial Recurrent Neural Network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.[8] The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

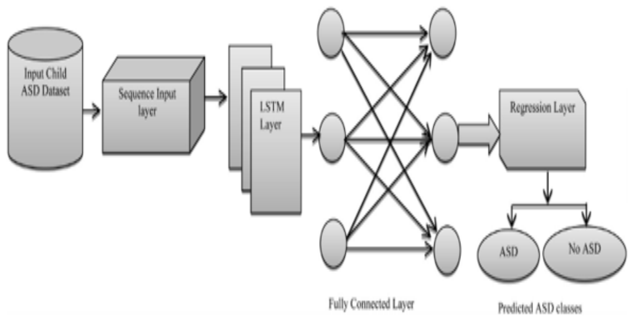


Fig. 3 LSTM Architecture

These modules are typically used to build and train deep learning models

- Sequential from keras.models: This is a class that allows us to create a linear stack of layers in the deep learning model.
- Dense and Dropout from keras.layers: These are classes that define fully connected layers and dropout regularization in the model.
- BatchNormalization, MaxPooling1D, Conv1D, and Flatten from tensorflow.keras.layers: These are classes that define batch normalization, max pooling, 1D convolution, and flattening layers in the model.
- Adam from tensorflow.keras.optimizers: This is a class that defines the Adam optimization algorithm for training the model.
- load_model from keras.models: This is a function that allows us to load a pre-trained model from a file.

ReLU (Rectified Linear Unit) and Softmax activation functions are typically employed as activation functions within a neural network architecture to introduce meaningful outputs.

The ReLU (Rectified Linear Unit) activation function is calculated element-wise for each neuron in the dense layers. The calculation is as follows:

For a given input value x, the ReLU activation function is defined as: f(x) = max(0, x)

The calculation for ReLU can be summarized as follows:

1. For the first dense layer with 81 units:

• The output value of each neuron in this layer is computed by taking the dot product of the input features with their corresponding weights, followed by the addition of a bias term.

• After obtaining the output value for each neuron, the ReLU activation function is applied element-wise.

• If the output value of a neuron is negative, it is set to zero (f(x) = 0).

• If the output value of a neuron is non-negative, it remains unchanged (f(x) = x).

2. For the second dense layer with 27 units:

• Similar to the first dense layer, the output values of each neuron are computed by taking the dot product of the input features with their corresponding weights, followed by the addition of a bias term.

• The ReLU activation function is then applied element-wise to the output values.

3. For the third dense layer with 9 units:

• Again, the output values of each neuron are calculated using the dot product of the input features, weights, and bias term.

• The ReLU activation function is applied to the output values element-wise.

It's important to note that the ReLU activation function is calculated independently for each neuron in the dense layers, allowing the model to capture non-linear relationships and introduce non-linearity into the network. This calculation is performed during the forward propagation step of the neural network, where the input passes through the layers and activations are computed.

SoftMax Activation Function:

SoftMax is commonly used in the output layer of neural networks for multi-class classification tasks, including autism detection.

SoftMax activation function, f(x) = e^(x_i) / Σ e^(x_j),

calculates the probability of an input belonging to each class i based on its corresponding output value x_i.

e^(x_i): The exponential function is applied to the output value x_i for a specific class i. This exponentiation ensures that the output value is positive.

Σ e^(x_j): The sum of the exponential values of all class outputs. This denominator is calculated by summing the exponentiated output values over all classes j.

f(x): The resulting probability distribution for each class. The Softmax function normalizes the exponentiated output values, dividing each value by the sum of all the exponentiated values. This ensures that the probabilities sum up to 1, making it a valid probability distribution.

The SoftMax function produces a probability distribution over multiple classes, indicating the likelihood of the input belonging to each class. The class with the highest probability is often selected as the predicted class.In this, the SoftMax activation function can be used in the output layer of a neural network to assign probabilities to different classes, such as "autism" and "non-autism."

Batch Normalization: To add a batch normalization layer in a neural network,

1.Import the Required Libraries: Depending on the deep learning frameworks are using (such as TensorFlow, PyTorch, or Keras), import the necessary libraries and modules.

2. Define the Batch Normalization Layer: In the model architecture, locate the position where the model want to add the batch normalization layer. This is typically after a fully connected layer or before an activation function.

Create a batch normalization layer using the appropriate function or class provided by the deep learning framework.

3. Add the Batch Normalization Layer to the Model: Insert the batch normalization layer into the model architecture at the desired position. Connect the output of the previous layer to the input of the batch normalization layer. Ensure the number of features or channels matches the expected input size of the batch normalization layer.

Batch Norm is a normalization technique done between the layers of a Neural Network instead of in the raw data. It is done along mini-batches instead of the full data set. It serves to speed up training and use higher learning rates, making learning easier.

Following the technique explained in the previous section, can define the normalization formula of Batch Norm as:

ZN= (z-mz) / s z

being mz the mean of the neurons' output and s z the standard deviation of the neurons' output.

In the following feed-forward Neural Network image: xi are the inputs, z the output of the neurons, ‘a’ the output of the activation functions, and y the output of the network:

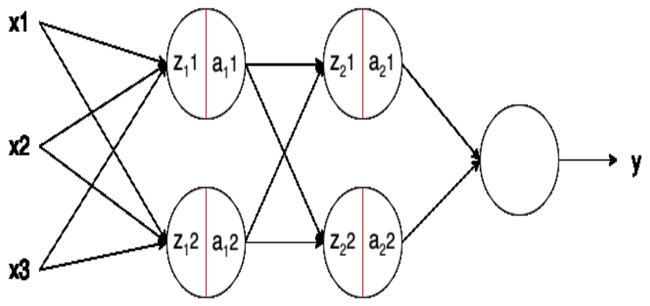


Figure 4 Batch Normalization

Batch Norm – in the image represented with a red line – is applied to the neurons’ output just before applying the activation function. Usually, a neuron without Batch Norm would be computed as follows:

z=g(w,x)+b; a=f(z)

being g() the linear transformation of the neuron, w the weights of the neuron, b the bias of the neurons, and f() the activation function. The model learns the parameters w and b. Adding Batch Norm, it looks as:

z =g(w,x); zN=(z-mz)/sz.β+γ; a=f(zN)

being zN the output of Batch Norm, mz the mean of the neurons’ output, sz the standard deviation of the output of the neurons, and γ and β learning parameters of Batch Norm. Note that the bias of the neurons (b) is removed. This is because as subtract the mean mz , any constant over the values of z – such as b – can be ignored as it will be subtracted by itself.

The parameters β and γ shift the mean and standard deviation, respectively. Thus, the outputs of Batch Norm over a layer results in a distribution with a mean β and a standard deviation of γ . These values are learned over epochs and the other learning parameters, such as the weights of the neurons, aiming to decrease the loss of the model.

The lstm model function creates a Keras Sequential model with four dense layers and three batch normalization layers. The first dense layer has 81 units and uses the ReLU activation function. The first batch normalization layer is added after the first dense layer. The second dense layer has 27 units and uses the ReLU activation function. The second batch normalization layer is added after the second dense layer. The third dense layer has 9 units and uses the ReLU activation function. The third batch normalization layer is added after the third dense layer. The fourth dense layer has 3 units and uses the SoftMax activation function. The SoftMax activation function is used to ensure that the output of the model is a probability distribution over the three classes.

Finally, the function returns the created model.

MODEL EVALUATION:

P(Autism) = e^(score_autism) / Σ(e^(score_i)) for all classes

Where:

- P(Autism) represents the probability of the input belonging to the "Autism" class.
- e^(x) represents the exponential function applied to the input value x.
- score_autism is the output or score associated with the "Autism" class from the Softmax layer.
- score_i represents the output or score associated with each class from the Softmax layer.

In this formula, the numerator calculates the exponential of the score associated with the "Autism" class. The denominator sums up the exponential values of the scores for all classes. Dividing the numerator by the denominator provides the probability of the input belonging to the "Autism" class.

The higher the probability, the more severe the autism may be perceived.

For example:

- If P(Autism) is close to 1 (e.g., 0.9 or higher), it may indicate a high likelihood of severe autism.
- If P(Autism) is around 0.5, it may suggest a moderate probability of autism with a moderate level of severity.
- If P(Autism) is close to 0 (e.g., 0.1 or lower), it may indicate a low likelihood of autism or a milder form of autism.

Four measures such as accuracy, F-1 score, precision, recall to classification algorithms' performance. All of these evaluation measures can be calculated by using those following equations:

Accuracy = ……………………………….(4)

Precision = ……………………………….....…(5)

Recall = ……..………...………..……………..…(6)

F-1 Score = ………………………………...(7)

The terms of those above equations represent,

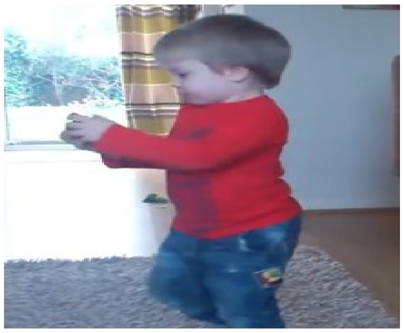
TP = True Positive

TN = True Negative

FP = False Positive

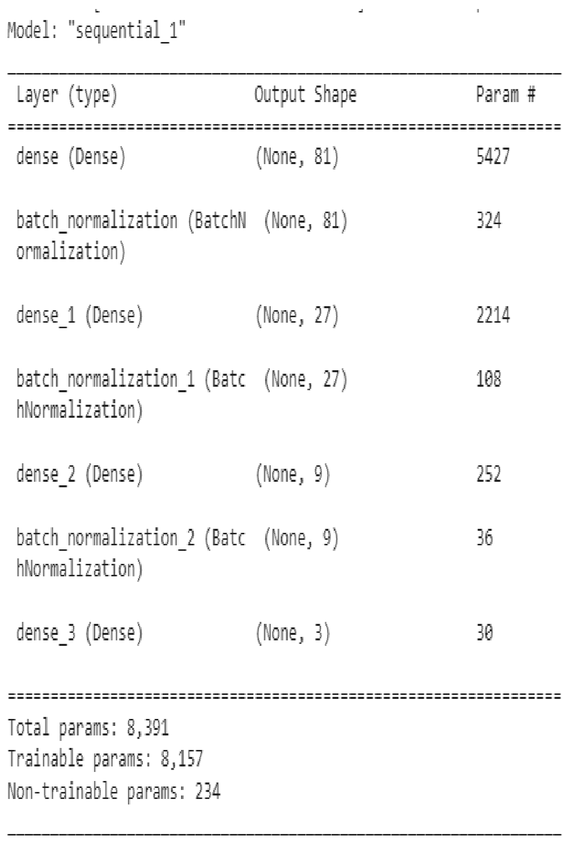
FN = False Negative

Experimental Result:



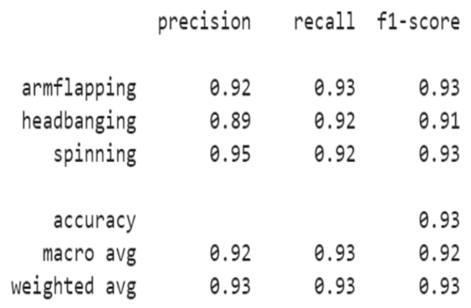
```
Probability of spinning 94.95%
```

Fig. 5-Prediction



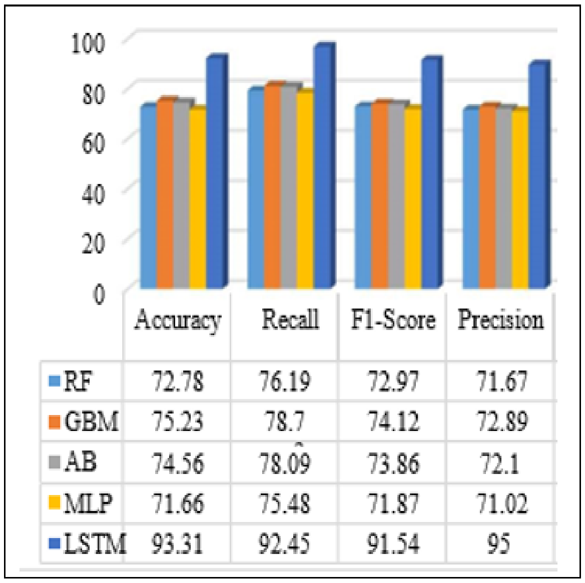
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 81)                5427

 batch_normalization (BatchN  (None, 81)               324
 ormalization)

 dense_1 (Dense)             (None, 27)                2214

 batch_normalization_1 (Batc  (None, 27)               108
 hNormalization)

 dense_2 (Dense)             (None, 9)                 252

 batch_normalization_2 (Batc  (None, 9)                36
 hNormalization)

 dense_3 (Dense)             (None, 3)                 30

=================================================================
Total params: 8,391
Trainable params: 8,157
Non-trainable params: 234
_________________________________________________________________
```

Fig. 6 LSTM training output



```
              precision    recall  f1-score

 armflapping       0.92      0.93      0.93
 headbanging       0.89      0.92      0.91
    spinning       0.95      0.92      0.93

    accuracy                           0.93
   macro avg       0.92      0.93      0.92
weighted avg       0.93      0.93      0.93
```

Fig. 7 LSTM classification report



	Accuracy	Recall	F1-Score	Precision
🟦 RF	72.78	76.19	72.97	71.67
🟧 GBM	75.23	78.7	74.12	72.89
⬜ AB	74.56	78.09	73.86	72.1
🟨 MLP	71.66	75.48	71.87	71.02
🟦 LSTM	93.31	92.45	91.54	95

Fig. 8 Comparison report

Conclusion:

Autism is the brain growth development disorder, this will leads to abnormal activities of the children. To detect this the video is used. The video collected and from the Self-Stimulatory Behaviours Dataset, which provided a valuable resource for training and evaluating the model. preprocessing step of resizing the images. The resizing process contributed to enhanced computational efficiency, reducing the computational requirements during training. By leveraging pose detection techniques, Extracted relevant features from frames, allow to capture body movement patterns that are indicative of autism. The model is based on LSTM architecture, demonstrated promising results in detecting autism. The use of ReLU activation function in the hidden layers facilitated the capturing of complex patterns and improved the model's ability to learn from the data. The Softmax activation function in the output layer provided a probability distribution over the classes, enabling reliable estimation of the likelihood of autism.The evaluation of the model on the test dataset showed encouraging performance metrics, such as accuracy, precision, recall, and F1 score. These results suggest that the approach holds promise in accurately identifying individuals with autism based on their pose-related features. In Future, it is planned to Investigate methods to capture long-term temporal dependencies in the data using more advanced recurrent neural network (RNN) architectures, such as Gated Recurrent Units (GRUs) or Transformer-based models, to capture complex temporal patterns and improve the model's performance and it is planned to implementation and deployment of the model in real-world settings, such as clinical environments or mobile applications.

Reference:

[1] Akshay Puli; Azadeh Kushki, “Toward Automatic Anxiety Detection in Autism: A Real-Time Algorithm for Detecting Physiological Arousal in the Presence of Motion”, IEEE Transactions on Biomedical Engineering [Vol no: 67, 2019]

[2] Alice Z. Guo, “Automated Autism Detection based on Characterizing Observable Patterns from Photos”, IEEE Transactions on Affective Computing

[3] Annette McWilliams; Parmida Beigi; Akhila Srinidhi; Stephen Lam; Calum E. MacAulay, “Sex and Smoking Status Effects on the Early Detection of Early Lung Cancer in High-Risk Smokers Using an Electronic Nose”, IEEE Transactions on Biomedical Engineering [Vol no: 62, 2015]

[4] Arnaud Arindra Adiyoso Setio; Francesco Ciompi; Geert Litjens; Paul Gerke; Colin Jacobs; Sarah J. van Riel, “Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks”, IEEE Transactions on Medical Imaging [Vol no: 35, 2016]

[5] Chunyu Wang; Junling Guo; Ning Zhao; Yang Liu; Xiaoyan Liu; Guojun Liu; Maozu Guo, “A Cancer Survival Prediction Method Based on Graph Convolutional Network”, IEEE Transactions on NanoBioscience [Vol no: 19, 2020]

[6] Hua Zhong; Mingzhou Song, “A Fast Exact Functional Test for Directional Association and Cancer Biology Applications”, IEEE/ACM Transactions on Computational Biology and Bioinformatics [Vol no: 16, 2019]

[7] Huan Zhao; Amy R. Swanson; Amy S. Weitlauf; Zachary E. Warren; Nilanjan Sarkar, “Hand-in-Hand: A Communication-Enhancement Collaborative Virtual Reality System for Promoting Social Interaction in Children With Autism Spectrum Disorders” IEEE Transactions on Human-Machine Systems [Vol no: 48, 2018]

[8] Koyel Mandal; Rosy Sarmah; Dhruba Kumar Bhattacharyya, “Biomarker Identification for Cancer Disease Using Biclustering Approach: An Empirical Study”, IEEE/ACM Transactions on Computational Biology and Bioinformatics [Vol no: 16, 2019]

[9] Lian Zhang; Ashwaq Z. Amat; Huan Zhao; Amy Swanson; Amy Weitlauf; Zachary Warren; Nilanjan Sarkar, “Design of an Intelligent Agent to Measure Collaboration and Verbal-Communication Skills of Children With Autism Spectrum Disorder in Collaborative Puzzle Games”, IEEE Transactions on Learning Technologies [Vol no: 14, 2020]

[10] Madhura Ingalhalikar, Sumeet Shinde, Arnav Karmarkar, Archith Rajan, D. Rangaprakash, Gopikrishna Deshpande,” Functional connectivity-based prediction ofAutism on site harmonized ABIDE dataset”, IEEE Transactions on Biomedical Engineering [vol no: 68, 2021]

[11] Mohammadreza Sehhati; Alireza Mehridehnavi; Hossein Rabbani; Meraj Pourhossein, “Stable Gene Signature Selection for Prediction of Breast Cancer Recurrence Using Joint Mutual Information”, IEEE/ACM Transactions on Computational Biology and Bioinformatics [Vol no: 12, 2015]

[12] Nasibeh Talebi; Ali Motie Nasrabadi; Iman Mohammad-Rezazadeh; Robert Coben,“nCREANN: Nonlinear Causal Relationship Estimation by Artificial Neural Network; Applied for Autism Connectivity Study”, IEEE Transactions on Medical Imaging [Vol no: 38, 2019]

[13] Nastaran Emaminejad; Wei Qian; Yubao Guan; Maxine Tan; Yuchen Qiu; Hong Liu; Bin Zheng, “Fusion of Quantitative Image and Genomic Biomarkers to Improve Prognosis Assessment of Early Stage Lung Cancer Patients”, IEEE Transactions on Biomedical Engineering [Vol no: 63, 2016]

[14] Noah Berlow; Saad Haider; Qian Wan; Mathew Geltzeiler; Lara E. Davis; Charles Keller; Ranadip Pal, “An Integrated Approach to Anti-Cancer Drug Sensitivity Prediction”, IEEE/ACM Transactions on Computational Biology and Bioinformatics [Vol no: 11, 2014]

[15] Taban Eslami, Fahad Almuqhim, Joseph S. Raiker and Fahad Saeed, “Machine Learning Methods for Diagnosing Autism Spectrum Disorder and Attention-Deficit/Hyperactivity Disorder Using Functional and Structural MRI: A Survey”, Frontiers in Neuroinformatics [Vol no: 14, 2021]