Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 13, No. 3, Juli 2026

ISSN 0216 – 0544
e-ISSN 2301– 6914

Deep learning algorithms for autism spectrum disorder detection using eye-tracking patterns

Muhamad Syukron [a], Rangga Alif Faresta [b]

[a] Department of Statistic, Institut Teknologi Sepuluh Nopember, Indonesia
[b] Digital Learning, Monash University, Australia
E-mail: muhamad.syukron@its.ac.id

Abstract

Early diagnosis of autism spectrum disorder (ASD) plays a crucial role in facilitating prompt interventions and assessing post-therapy progress, both of which can greatly improve developmental trajectories. The emergence of artificial intelligence—particularly deep learning—has opened new possibilities for clinicians to detect ASD with improved precision and speed. This study investigates the utilization of deep learning models to recognize ASD among 13-year-old children based on eye movement data collected as participants observed static images and short video sequences. The dataset included visual and numerical variables, such as gaze position and pupil diameter, allowing for a multimodal analytical approach. For the numerical dataset, a multilayer perceptron (MLP) neural network produced the best outcomes, yielding an accuracy of 91.7% and a recall rate of 83.3% in ASD classification. Meanwhile, the Vision Transformer (ViT) model performed best for image-based analysis, reaching an accuracy of 78.2% and a recall rate of 88.6%. Overall, the findings highlight the potential of deep learning frameworks as objective, data-driven tools for ASD detection in both clinical and research contexts.

Key words: Autism, Computer Vision, Deep Learning Model, Vision Transformer.

INTRODUCTION

Autism Spectrum Disorder (ASD) is marked by ongoing challenges in social communication and in developing and maintaining interpersonal relationships, which are central diagnostic indicators [1]. A thorough understanding of these criteria is essential for accurate diagnosis and effective intervention. Enhancing knowledge ASD characteristics among both healthcare professionals and the public is essential to strengthen early recognition, ensure appropriate support, and improve overall management of the condition. Individuals with ASD often face significant challenges in educational settings, particularly in tasks involving multiple complex steps due to limited proficiency in daily living skills (DLS) [2]. This highlights the importance of embedding life skills education within the curriculum to foster the holistic growth of learners with ASD. In Australia, 85% of children diagnosed with ASD experience challenges in educational settings, with 63% struggling with social interactions and 62% facing difficulties with learning [3]. The data underscore the extensive difficulties encountered by students ASD and emphasize the urgent necessity for targeted, effective interventions., particularly in high school, where issues such as adapting to new situations, maintaining social relationships, processing information, and managing time become more pronounced [4].

Gender also plays a significant role in the challenges faced by students with ASD. Stereotypes and biases can influence teachers' perceptions, leading to unequal support for male and female students [5]. Empirical evidence indicates that male students diagnosed that has ASD frequently encounter increased difficulties in attaining social integration and are more susceptible to social isolation than their female counterparts, who generally experience smoother social acceptance [6]. This points to the need for gender-sensitive approaches in social skills training to promote better social integration for all students with ASD.

Current treatment for ASD encompasses a range of therapies, including behavioural, pharmacological, and psychological interventions. A holistic approach that integrates these modalities is crucial for effectively addressing the complex challenges associated with ASD [7]. Behavioural interventions play a vital role in improving patients' ability to manage behaviours, develop vocational skills, and enhance social, cognitive, and daily living abilities [8]. Additionally, A range of interventions—including educational initiatives, family-based support, speech and occupational therapy, social skills development programs, and the Picture Exchange Communication System (PECS)—offer comprehensive assistance that addresses the varied developmental needs of individuals ASD [9].

As individuals' cognitive skills progress, the need for external assistance generally decreases, especially among those with high-functioning autism who can benefit from mainstream educational programs and broader support systems [10]. This transition underscores the necessity of tailoring intervention approaches to correspond with changing cognitive capacities, thereby fostering autonomy and improving both social and academic inclusion. Implementing inclusive pedagogical practices that cater to diverse learning styles and individual requirements is vital for advancing academic attainment and social growth in students with ASD. A thorough insight into the mental and cognitive traits of people with ASD, combined with the application of targeted interventions, can significantly improve their overall well-being while promoting a more equitable and a nurturing setting for individuals with ASD and their loved ones [11].

Considering the considerable difficulties encountered by students with ASD, early intervention is critical to support their social, cognitive, and academic development. Recent advances in artificial intelligence and deep learning offer promising opportunities to enhance early detection and personalized intervention strategies. By leveraging deep learning models to analyse behavioural, social, and educational data, we can identify early signs of ASD more accurately and tailor interventions to individual needs.

Furthermore, these models can be applied post-therapy to evaluate the effectiveness of interventions and monitor whether ASD-related symptoms persist, allowing for adaptive and continuous support. The implementation of technology-based interventions can promote timely therapeutic actions, enhance learning effectiveness, and ultimately enhance the overall quality of life for children identified with ASD over time

Patterns in eye movement can serve as important indicators for recognizing individuals with autism spectrum disorder (ASD). These critical movements include fixation, where the eyes focus on a specific object; saccades, characterized by rapid scanning; and blinks, where the eyes temporarily lose focus [12]. Numerous studies have analyzed eye movements to detect ASD using machine learning techniques. For instance, researchers asked participants to watch videos or films [13-17], engage in focused tasks [18, 19], or recorded eye movements during interviews [20]. Other methods included using public datasets [21] or employing virtual reality (VR) technology [22, 23].

SVM model ranks as one of the most widely applied algorithms in machine learning for ASD identification [14, 18, 20]. This approach distinguishes data into separate categories using hyperplanes—both nonlinear and linear—that effectively partition the dataset. Other scholars [13, 17] have employed other machine learning approaches, including Decision Trees and ensemble methods like Random Forest, which categorize data by dividing it at various nodes according to relevant attributes. Finding the optimal depth of the Decision Tree is essential to balance performance and overfitting. Several studies [19, 22, 23] have also utilized distance-based methods such as K Means and K Nearest Neighbors (KNN). The KNN algorithm operates on labeled datasets, while K-Means functions as an unsupervised method that necessitates predefining the number of clusters.

Lately, deep learning approaches have attracted significant interest in detecting ASD because they can automatically identify important features from large and complex datasets. Among these approaches, CNN and also RNN are among the most utilized frameworks for ASD-related classification tasks. CNN model is particularly effective in processing visual data, including eye-tracking information and facial recognition, through convolutional operations that capture patterns in input matrices [15, 21]. RNNs, well-suited for handling sequential data such as text and audio, help capture temporal dependencies in ASD-related datasets [21]. Autoencoders, an unsupervised deep learning technique, have also been employed for feature extraction in ASD research. These models compress input data into lower-dimensional representations, which can then be clustered to detect patterns in eye movements. For example, Elbattah et al. [19] used autoencoders on scan-path data, confirming coherent clusters for ASD identification.

Other studies employed deep neural networks (DNNs) to analyze eye movement velocity, acceleration, and jerk. Kanhirakadavath & Chandran [17] demonstrated the power of DNNs by achieving 78.6% accuracy in classifying ASD individuals, with a significant boost in performance after applying data augmentation techniques. In a similar vein, deep learning techniques have been applied to identify ASD by examining gaze patterns, frequently outperforming conventional machine learning techniques in terms of accuracy. For instance, Lin et al. [23] integrated deep learning methods with eye movement metrics to predict elevated levels of autistic traits, achieving 70% accuracy, further showcasing the effectiveness of deep learning in detecting ASD.

MATERIAL AND METHODS

Neural Network

A neural network represents a computational architecture inspired by the human brain, in which each neuron is interconnected to process information and generate corresponding outputs or decisions. At its core, a neural network is composed of a minimum of two layers: an input layer and an output layer, forming the simplest model known as a perceptron. Any layer situated between the output and input layers is termed a hidden layer. The total number of hidden layers may differ, resulting in what is referred to as a multi-layer perceptron (MLP) [24]. As more hidden layers are added and the overall network structure becomes increasingly complex, the system is categorized as a deep neural network. Figure 1 illustrates the links running from the input layer to the hidden layer and then from the hidden layer to the output layer, representing the associated weights within the network. During the training phase, these weights are iteratively adjusted to determine the most optimal values that yield accurate output predictions. The output layer then applies an activation function—such as using a linear function for regression, a sigmoid function for two-class classification, or a SoftMax function for multiple-class classification—to generate the final results.

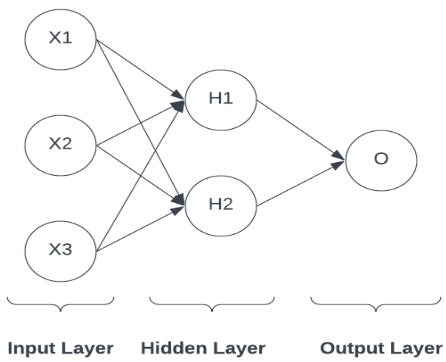


Fig. 1. Neural network architecture

Convolutional Neural Network (CNN)

CNN is a model specifically created to handle and analyze input data structured in matrix form, such as images represented as pixel matrices. Image data commonly contain three channels—red, green, and blue—for coloured images, or a single grayscale channel, which makes CNNs highly efficient and reliable for visual data analysis. Initially, CNNs were developed for tasks such as handwriting recognition [25]. Later, this algorithm and its improvements were widely used for image classification [26-30]. A Convolutional Neural Network (CNN) comprises three fundamental components: the feature-extraction layer, the down sampling layer, and the fully connected layer (see Figure 2). The feature-extraction layer is responsible for identifying of distinctive patterns such as edges, curves, or other complex visual characteristics by applying a kernel filter $K$to the input matrix or feature map I, which produces an output feature map O. Mathematically, this process can be represented as follows:

$$O(a,b) = \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} I(a+p, b+q)\cdot K(p,q) \qquad (1)$$

Here, (a, b) represents the coordinates in the output feature map, while (p, q) correspond to the dimensions of the kernel. Following the convolution step, pooling is applied to reduce the size of the feature map by choosing either the maximum or average value from a subregion, a procedure known as max pooling. The max pooling operation can be described mathematically as:

$$P(a,b) = \max\{O(a+p, b+q)\} \qquad (2)$$

where $P(a, b)$ denotes the pooled feature map, and (r, s) represents the dimensions of the pooling window. This operation generates a more compact feature map while preserving the essential information.

After the pooling stage, the resulting matrix is flattened into one-dimensional vector, which is then sent to the dense layer for further processing. In this layer, the weights are optimized to carry out classification tasks, such as distinguishing between ASD and TD. The dense layer performs the role of a multi-layer perceptron, where the quantity of layers and the number of neurons serve as hyperparameters that must be fine-tuned to attain optimal model performance.

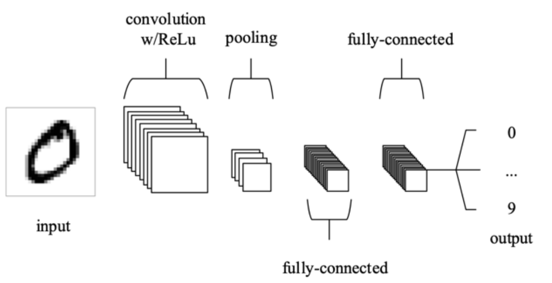


Fig. 2. Convolutional neural network architecture

Residual Network (ResNet)

ResNet is a complex neural framework engineered to mitigate gradient disappearance and overflow problems that commonly emerge in highly deep neural structures. Vanishing gradients result from small weights. These cause the signal to shrink as it moves through the layers, leading to a loss of information. On the other hand, exploding gradients occur when large weights cause the signal to grow excessively. This makes the model unstable.

ResNet introduces an innovative concept known as *residual learning* [31], which concentrates on learning the residual, defined as the deviation between the original input and the intended output. In this context, the network models the difference between what is desired and the input rather than the entire mapping directly. Rather than requiring each layer to learn the complete transformation, the network focuses on refining the residual $F(x)$, while the input $x$is transmitted directly to the output via a shortcut path. This mechanism, referred to as a *skip connection*, integrates the input $x$with the output after the residual function has been applied, producing $H(x) = F(x) + x$. Such an approach enables the network to preserve the integrity of the original input information as it propagates through deeper layers, thereby minimizing the loss of crucial data (see Figure 3).

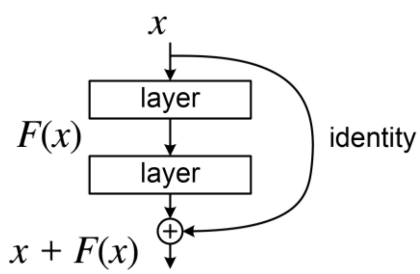


Fig. 3. Skip connection in ResNet

The original ResNet-152 model comprises 152 layers, consisting of 50 residual blocks along with a fully connected layer. Each residual block is composed of a varying number of layers and filters, which are designed to capture features at different levels of abstraction and complexity.

Dense Network (DenseNet)

Similar to ResNet, DenseNet was introduced to mitigate the diminishing gradient problem frequently observed in profoundly layered neural architectures [32]. DenseNet includes multiple variants such as DenseNet-169, DenseNet-201, DenseNet-264 and

DenseNet-121. Each version comprises convolutional layers followed by pooling operations, dense blocks, and a final classification layer.

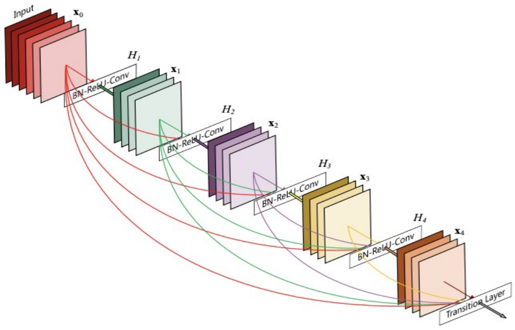


Fig. 4. Dense connection in DenseNet architecture

The fundamental idea behind DenseNet lies in the use of direct connections from each layer to all subsequent ones. Mathematically, for a given layer l, the resulting feature map is computed as:

$$Y_l = C_l([y_0, y_1, \ldots, y_{l-1}]) \quad (3)$$

Here, $[y_0, y_1, \ldots, y_{l-1}]$ stands for the combined feature maps from all preceding layers, starting from layer 0 up to layer $l-1$. Figure 4 provides a visual overview of the DenseNet layout.

Transformer

The Transformer is an advanced neural architecture engineered to handle sequential information through parallel processing, making it a widely used model for NLP applications. This approach has predominantly replaced Recurrent Neural Network (RNN)-based architectures in many NLP tasks. Unlike RNNs, which process sequences recursively and can struggle with long texts due to information loss over time, Transformers utilise positional embeddings and multi-head attention to address this issue [33]. The attention mechanism allows Transformers to effectively manage the relationships between distant elements in a sequence. Transformers are not limited to text data; they can also be applied to images. Images may be converted into sequential data by progressing in a sequential sweep from the upper-left quadrant toward the lower-right extremity, while Transformer's attention mechanism allows the model to concentrate on spatial coordinates, within an image, helping to extract important features and understand the content more effectively.

Vision Transformer

Adapting the Transformer architecture for computer vision tasks requires specific modifications, unlike its application in natural language processing. Since the Transformer lacks inherent spatial awareness, the Vision Transformer introduces positional embeddings for each image patch [34]. These patches, analogous to tokens in NLP, are the smallest units of the image. Typically, an image is divided into multiple patches, such as 16x16, as outlined in the original paper. The input to the Vision Transformer consists of these patch embeddings combined with positional embeddings, which are then processed through an encoder-decoder structure like the original Transformer. In conclusion, a decision-making layer is incorporated at the terminal stage to yield the resultant predictions.

Swin Transformer

Like the Vision Transformer, the Swin Transformer requires specific modifications to adapt the original transformer architecture for computer vision tasks. It achieves this through a hierarchical structure and a shifted windows mechanism [35]. The Swin Transformer constructs hierarchical feature maps to effectively handle visual representations of images at various scales, enabling the capture of patterns from small to large objects. Additionally, it modifies the attention mechanism of the original Transformer, shifting from a global to a non-overlapping local window approach. To ensure continuity of information across windows, the Swin Transformer employs a shifting scheme (see Figure 5). By using local attention, the Swin Transformer significantly reduces computational complexity, especially when processing high-resolution images.

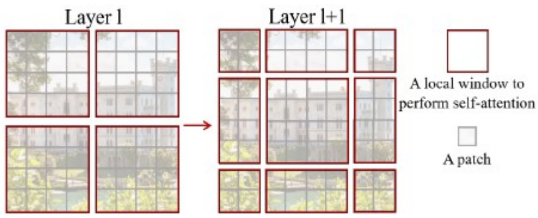


Fig. 5. Local attention mechanism without window shifting (left) and with window shifting (right)

Experiment Setup

Datasets

This study utilizes eye-tracking data reported by Cilia et al. [36], comprising 59 individuals in total, including 30 children identified with ASD and 29 typically developing (TD) counterparts. The participants had an average age of 8 years, and the mean Childhood Autism Rating Scale (CARS) [37] score for the ASD group was 32.97. The dataset consists of 2,252,943 records, distributed across 25 CSV files. This large number of records is due to the frequent eye movement tracking for each participant, with multiple recordings per second. Table 1 encapsulates the main variables analyzed within this investigation.

Table 1. Variable used for ASD detection

Varible	Description
T	Recording time of eye movements (in milliseconds)
ID	ID number for each participant
Category	Type of eye movement (Fixation, Saccade, Blink)
CPD	Circular pupil diameter measured in millimeters
HPD	Horizontal dimension of the pupil size in pixels
VPD	Vertical dimension of the pupil size in pixels
HGC	Horizontal coordinate of the gaze point
VGC	Verical coordinate of the gaze point

In addition to using numerical record data, this study also utilises image data generated by Cilia et al. [36]. The process begins by creating a blank black image, with the first data point for each participant serving as the starting point on the image. The eye gaze coordinates at time $i+1$ are then connected to the coordinates at time i, and this continues until all records are plotted. The lines are colored based on velocity (red), acceleration (green), and jerk (blue)[38].

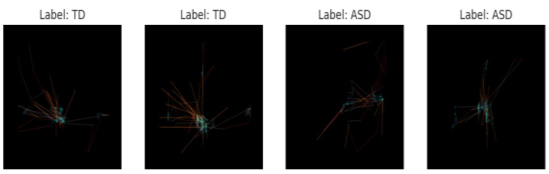


Fig. 6. Eye tracking dataset from TD and ASD participants

The results can be seen in Figure 6. In total, 219 images were constructed and stored in the TSImages folder for ASD children and the TCImages folder for TD children.

Model Setup

In this study, five architectures will be used to predict ASD: Neural Network, ResNet, DenseNet, Vision Transformer, and Swin Transformer. The architecture developed for this study comprises a linear input layer, a pair of intermediate hidden layers, and a terminal output layer. The initial hidden layer contains 64 processing units, whereas the second comprises 32 neurons. A sigmoid activation mechanism is employed in the terminal layer to facilitate the binary classification task. This Neural Network will use numerical data, while the other models will use image data.

We employed the ResNet-50 architecture, which comprises 50 layers. The network starts with a convolutional layer applying a 7×7 filter with a stride of 2, then performing a 3×3 max-pooling operation. Each residual block is composed of two 1×1 convolutional layers that compress and restore feature dimensions, along with a 3×3 convolutional layer for feature extraction. Batch normalization is applied after every convolution within the residual blocks. After these blocks, a global max-pooling operation is performed ahead of integration with the fully connected stage. In comparison, DenseNet structure places a transition layer after each dense block, which includes a 1×1 convolution and 2×2 average pooling. Similar to ResNet-50, global average pooling is applied before the final classification layer. For the Vision Transformer (ViT), the input is divided into 16×16 patches, which are then fed into a classic transformer system featuring both encoding and decoding units, ending with a binary classification head. Meanwhile, the Swin Transformer employs 4×4 patches and leverages a local attention window of 7×7.

Implementation Details

For the tabular data experiments, we applied three models. The neural network was trained using the Adam optimizer with a learning rate of 0.001, Binary Cross Entropy loss, one hundred epochs, full batch training, ReLU activations in the hidden layers, and a sigmoid output. The SVM model used a Gaussian kernel with a regularization value C equal to one and gamma defined as one divided by the product of the number of features and their variance. The k Nearest Neighbors model used five neighbors, uniform weighting, and the Minkowski distance metric. For the image based experiments, all models shared the same training configuration, which consisted of the Adam optimizer with a learning rate of 0.001, full batch training, one hundred epochs, and Cross Entropy loss. The training process was conducted on Google Colab using 12 gigabytes of RAM and an NVIDIA Tesla T4 GPU.

Metric Evaluation

Recall measures the fraction of genuine positive instances accurately captured by the model. It is useful when missing positive cases (e.g., correctly identifying ASD individuals) is more critical than falsely identifying negatives.

$$Recall = \frac{TP}{TP + FP} \quad (4)$$

Precision indicates the share of true positive predictions among all positive forecasts, proving vital when erroneous positive predictions are costly. (e.g., when minimizing the misclassification of TD children as ASD is crucial).

$$Precision = \frac{TP}{TP + FN} \quad (5)$$

Accuracy reflect the fraction of predictions that are correct across all classes, making it a trustworthy indicator in evenly distributed datasets. However, it can be misleading in cases of class imbalance.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

The F1 Score synthesizes precision and recall into a single harmonic measure, serving as an equitable evaluation tool in imbalanced datasets, and is ideal when mitigating both types of classification errors holds equal importance. Besides that, the F1 Score can avoid misleading interpretations because of imbalanced datasets.

$$F1\,Score = 2 \times \frac{Precision \times Recall}{Pr\,ecision + Recall} \quad (7)$$

RESULT AND DISCUSSION

Data Pre-Processing

The initial step before training the model was data pre-processing. For tabular data, the process began with removing observations that contained missing values. This decision was made because the percentage of missing data was only 4% of the total observations, which was low enough that deletion did not significantly impact the dataset, unlike imputation, which could have led to misleading results. Next, the percentage for each type of eye movement was calculated for each participant. For example, if Participant ID 1 had 10 records, 4 of which were fixation, 2 were blink, and 4 were saccade, the respective percentages for Participant ID 1 would be 0.4, 0.2, and 0.4. Percentages were used instead of raw counts because the number of observations varied across participants.

For other variables, such as pupil diameter or point of regard, central tendency and spread measures were calculated, including the mean, median, and standard deviation. This resulted in each variable being represented by three new columns, such as "Mean Pupil Size Right," "Median Pupil Size Right," and "Standard Deviation Pupil Size Right." After aggregation, the number of observations matched the number of participants, which was 59, and the dataset contained 34 columns. This included three columns for each variable except for Participant ID, Recording Time (ms), and Right/Left Category, as well as three columns for the percentage of each eye movement type. For the image dataset, we first performed size standardization to 224 x 224 pixels.

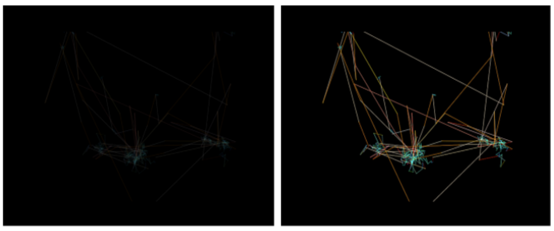


Fig. 7. Comparison of eye-tracking data before (left) and after preprocessing (right).

Due to the low RGB pixel values, which made it difficult to distinguish between eye movement lines and the background, pixel enhancement was applied by adjusting brightness, contrast, exposure, saturation, and temperature. Subsequently, the pixel intensities were standardized leveraging the mean and dispersion metrics extracted from the ImageNet corpus, which serves as foundation for pretraining deep learning architectures. Figure 7 presents a depiction of the processed images following the preprocessing steps.

Model Comparison

The dataset in tabular form was partitioned at random, allocating 80% for model training and 20% for evaluation. The training process was conducted using a neural network and two machine learning models. For deep learning, after 100 iterations, the training loss converged to 0.48. From Figure 8, it is evident that the training process could have been stopped at epoch 40, as there was no significant improvement afterward. Extending the training process beyond this point, even after the loss has converged, can result in unnecessary computational overhead and might induce overfitting, characterized by strong performance on the training set but poor adaptability to previously unseen data.

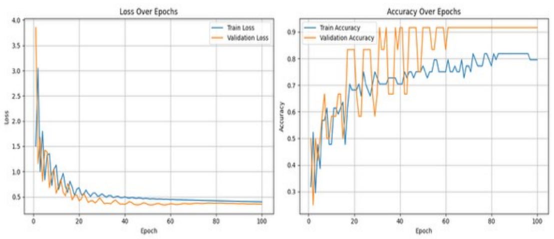


Fig. 8. Training and validation loss and accuracy for ASD detection using tabular data with hold out validation.

On the evaluation data, the model attained an accuracy level of 91.7%. For ASD class, the precision was 100%, the recall was 83.3%, and the F1 Score was 90.9%. This indicates that the model can effectively distinguish between adolescent with TD and ASD. The precision of 100 percent for the ASD class means that every sample the model predicted as ASD was actually ASD, and the model made no false positive predictions for that class. For the machine learning models, which are SVM and KNN, all performance metrics are lower than those of the neural network model. On the validation data, SVM achieves only 58.3% accuracy, while KNN reaches 83.3%. Further details are presented in Table 2.

Table 2. Metrics evaluation for ASD detection using tabular data

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
SVM	58.3	66.7	33.3	44.4
KNN	83.3	83.3	83.3	83.3
Neural Network	91.7	100	83.3	90.9

To test the robustness of the best model which is neural network, we perform 2-fold validation. For the first and second fold, we see that the loss is decreasing along the training process. The same performance also in val data, this means that there is no overfitting in the training process. For the accuracy in val data, both of fold can reach around 70%. This is because the number of data used in training is less than hold out validation. However, this result still shows that good result for small data. The plot between first and second fold also similar indicate that this model is robust.

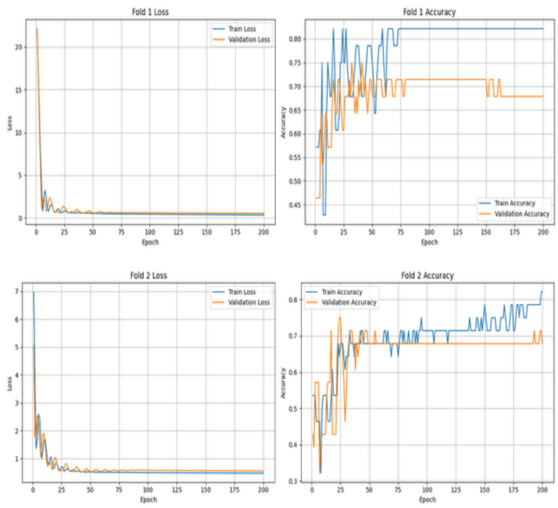


Fig. 9. Training and validation loss and accuracy for ASD detection using tabular data with 2-fold validation.

A comparison of four models, ResNet 50, DenseNet 121, Vision Transformer, and Swin Transformer, was conducted for image data. As shown in Figure 10, ResNet-50 and DenseNet-121 produced lower training losses compared to Vision Transformer and Swin Transformer. However, in the validation data, ResNet and DenseNet produced higher loss and lower accuracy. This indicates that these models experienced overfitting, while Vision Transformer and Swin Transformer did not.

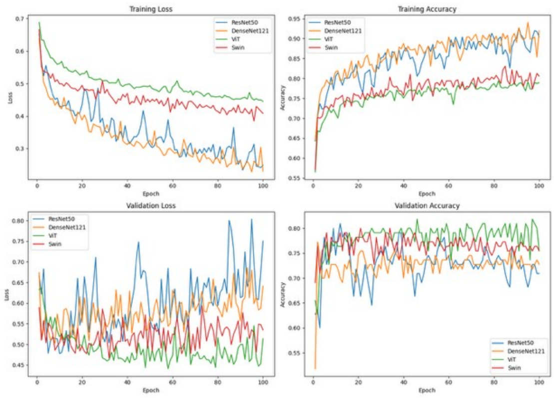


Fig. 10. Training and validation loss and accuracy for ASD detection using image data with hold out validation.

From the metric comparisons in Table 3, DenseNet-121 emerged as the weakest model, as it failed to achieve the highest score in any metric. Overall, Swin Transformer stands out as the best model, as it maintained performance above 70% across all metrics. However, if the priority is to avoid misclassifying individuals with ASD as TD, the model with the highest ASD recall—Vision Transformer—should be chosen.

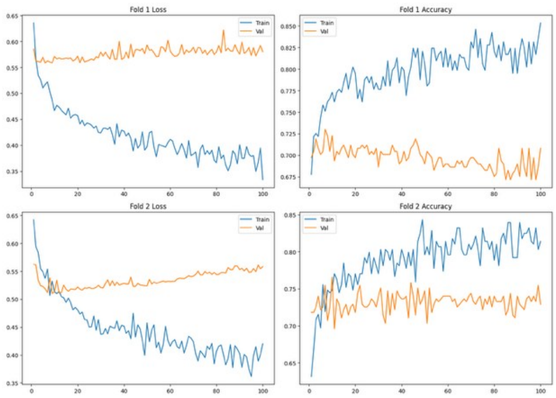


Fig. 11. Training and validation loss and accuracy for ASD detection using image data with 2-fold validation.

Table 3. Metrics evaluation for ASD detection using image data

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
ResNet-50	76.4	70.5	70.5	70.5
DenseNet-121	73.7	64	72.7	68.1
Vision Transformer	78.2	67.2	88.6	76.5
Swin Transformer	79.9	70.6	81.8	75.8

Figure 11 shows the loss and accuracy across epochs using two fold cross validation with ViT as the best model. From the figure we can see that the training loss decreases while the training accuracy increases. For the validation data, both loss and accuracy remain relatively flat with some volatility. This indicates that image data requires more samples for training and validation because extracting patterns from raw images is more challenging than working with tabular data. However, the validation accuracy is already reasonably strong at around seventy to seventy five percent. The similar trends in the first and second fold also indicate that ViT as the best model is robust.

Limitation and Future Works

This study has several limitations that should be acknowledged. First, the dataset size is relatively small, which may restrict the model's ability to generalize to more diverse samples. Second, the experiments were conducted offline, and real time testing is needed to evaluate how the models perform in practical deployment settings. Third, the current approach relies on single modality inputs, while future studies would benefit from incorporating multimodal fusion to combine features from text, images, audio, and behavioural cues for improved robustness and accuracy. These directions offer valuable opportunities to strengthen and extend the findings of this research.

CONCLUSION

Early identification of ASD is essential to ensure that children receive timely and appropriate interventions. Additionally, this detection can be applied post-therapy, such as after virtual reality or other treatments [39, 40]. ASD can be identified through various modalities, including eye movement analysis by observing fixation, saccades, and blinks. Data can be collected through methods like short videos, AR games, interviews, or tasks requiring eye engagement. The collected data can then be processed numerically using machine learning or neural networks or by directly analysing eye movement trajectories.

This research illustrates how ASD detection can be effectively performed using both numerical and visual data. The analysis showed that processing numerical data resulted in better performance than models using image-based inputs. Neural networks trained on numerical eye movement data achieved an accuracy of 91.7%. However, in terms of ASD recall, the Vision Transformer model, which used eye movement data, performed the best. ASD recall indicates the model's capacity to accurately recognize individuals diagnosed with ASD, whereas a lower recall indicates that ASD individuals are more likely to be misclassified as typically developing (TD). Ultimately, selecting the most suitable model rely on whether the priority is balanced accuracy across both classes or ensuring that children with ASD are not misclassified as TD.

Each model used in this study has specific strengths and limitations. SVM and KNN are computationally efficient but showed limited performance because they are relatively simple machine learning algorithms. The neural network achieved the highest accuracy on numerical eye tracking data, making it suitable for small datasets, although it relies on handcrafted features. ResNet 50 and DenseNet 121 can be effective for large scale image datasets, but in this study, the limited data caused these models to overfit. Vision Transformer and Swin Transformer demonstrated better generalization, with Vision Transformer achieving the highest ASD recall and Swin Transformer providing more balanced performance. Although these transformer based models require higher computational resources, the results remain strong.

REFERENCES

[1] American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders: DSM-5. Arlington, VA, USA: American Psychiatric Publishing, 2013. https://doi.org/10.1176/appi.books.9780890425596.

[2] M. S. Aljehany and K. D. Bennett, "A comparison of video prompting to least-to-most prompting among children with autism and intellectual disability," J. Autism Dev. Disord., vol. 50, no. 5, pp. 1714–1724, 2020, doi: 10.1007/s10803-019-03929-x.

[3] A. Farrell, "A spectrum of inclusion: Strategies for students with ASD," Agora, vol. 54, no. 2, pp. 40–43, 2019. [Online]. Available: https://www.proquest.com/scholarly-journals/spectrum-inclusion-strategies-students-with-asd/docview/2599639618/se-2

[4] V. Hees, T. Moyson, and H. Roeyers, "Higher education experiences of students with autism spectrum disorder: Challenges, benefits and support needs," J. Autism Dev. Disord., vol. 45, pp. 1673–1688, 2015, doi: 10.1007/s10803-014-2324-2.

[5] Y.-W. Nah and J. W.-L. Tan, "Brief report: The effect of gender on teachers' perceptions of behaviours in students with autism spectrum disorder (ASD)," Educ. Dev. Psychol., vol. 38, no. 2, pp. 249–254, 2021, doi: 10.1080/20590776.2021.1985378.

[6] M. Dean, C. Kasari, W. Shih, F. Frankel, R. Whitney, R. Landa, et al., "The peer relationships of girls with ASD at school: Comparison to boys and girls with and without ASD," J. Child Psychol. Psychiatry, vol. 55, no. 11, pp. 1218–1225, 2014, doi: 10.1111/jcpp.12242.

[7] T. Ma, H. Wu, and Y. Wang, "Application of different treatment methods for autism," in Proc. 3rd Int. Conf. Biol. Eng. Med. Sci., 2023, doi: 10.54254/2753-8818/20/20230705.

[8] R. Lordan, C. Storni, and C. A. De Benedictis, Autism Spectrum Disorders: Diagnosis and Treatment. Brisbane, Australia: Exon Publications, 2021. https://doi.org/10.36255/exonpublications.autismspectrumdisorders.2021.diagnosis

[9] S. Wyver, "Book review: Autism spectrum disorders: Interventions and treatments for children and youth," Autism, vol. 10, pp. 118–119, 2006. https://doi.org/10.1177/136236130601000110

[10] M. De, R. Chakraborty, M. Bhattacharya, and I. Basu, "Challenges for teachers in inclusive education vis-à-vis the barriers faced by students with autism spectrum disorder (ASD)," Harvest (Online), vol. 3, no. 1, 2018. [Online]. Available:

https://www.harvestjournal.net/wp-content/uploads/2018/07/Mitu-De.pdf
[11] J. M. Gillis and R. Beights, "New and familiar roles for clinical psychologists in the effective treatment for children with an autism spectrum disorder," Cogn. Behav. Pract., vol. 19, no. 3, pp. 392–400, 2012, doi: 10.1016/j.cbpra.2011.02.007.
[12] R. Simeoli, A. Rega, M. Cerasuolo, R. Nappo, and D. Marocco, "Using machine learning for motion analysis to early detect autism spectrum disorder: A systematic review," Rev. J. Autism Dev. Disord., 2024, doi: 10.1007/s40489-024-00435-4.
[13] M. Jiang, S. M. Francis, D. Srishyla, C. Conelea, Q. Zhao, and S. Jacob, "Classifying individuals with ASD through facial emotion recognition and eye-tracking," in Proc. 2019 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), 2019, pp. 6063–6068, doi: 10.1109/EMBC.2019.8857005.
[14] G. Wan, X. Kong, B. Sun, S. Yu, Y. Tu, J. Park, C. Lang, M. Koh, Z. Wei, Z. Feng, Y. Lin, and J. Kong, "Applying eye tracking to identify autism spectrum disorder in children," J. Autism Dev. Disord., vol. 49, no. 1, pp. 209–215, 2019, doi: 10.1007/s10803-018-3690-y.
[15] F. Cilia, R. Carette, M. Elbattah, G. Dequen, J.-L. Guérin, J. Bosche, L. Vandromme, and B. Le Driant, "Computer-aided screening of autism spectrum disorder: Eye-tracking study using data visualization and deep learning," JMIR Hum. Factors, vol. 8, no. 4, p. e27706, 2021, doi: 10.2196/27706.
[16] K. J. Tsuchiya, S. Hakoshima, T. Hara, M. Ninomiya, M. Saito, T. Fujioka, H. Kosaka, Y. Hirano, M. Matsuo, M. Kikuchi, Y. Maegaki, T. Harada, T. Nishimura, and T. Katayama, "Diagnosing autism spectrum disorder without expertise: A pilot study of 5-to 17-year-old individuals using Gazefinder," Front. Neurol., vol. 11, p. 603085, 2021, doi: 10.3389/fneur.2020.603085.
[17] M. R. Kanhirakadavath and M. S. M. Chandran, "Investigation of eye-tracking scan path as a biomarker for autism screening using machine learning algorithms," Diagnostics, vol. 12, no. 2, p. 518, 2022, doi: 10.3390/diagnostics12020518.
[18] W. Liu, M. Li, and L. Yi, "Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework," Autism Res., vol. 9, no. 8, pp. 888–898, 2016, doi: 10.1002/aur.1615.
[19] M. Elbattah, R. Carette, G. Dequen, J.-L. Guérin, and F. Cilia, "Learning clusters in autism spectrum disorder: Image-based clustering of eye-tracking scanpaths with deep autoencoder," in Proc. 2019 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), 2019, pp. 1417–1420, doi: 10.1109/EMBC.2019.8856904.
[20] Z. Zhao, H. Tang, X. Zhang, X. Qu, X. Hu, and J. Lu, "Classification of children with autism and typical development using eye-tracking data from face-to-face conversations: Machine learning model development and performance evaluation," J. Med. Internet Res., vol. 23, no. 8, p. e29328, 2021, doi: 10.2196/29328.
[21] S. Liaqat, C. Wu, P. R. Duggirala, S. S. Cheung, C.-N. Chuah, S. Ozonoff, and G. Young, "Predicting ASD diagnosis in children with synthetic and image-based eye gaze data," Signal Process. Image Commun., vol. 94, p. 116198, 2021, doi: 10.1016/j.image.2021.116198.
[22] M. Alcañiz, I. A. Chicchi-Giglioli, L. A. Carrasco-Ribelles, J. Marín-Morales, M. E. Minissi, G. Teruel-García, M. Sirera, and L. Abad, "Eye gaze as a biomarker in the recognition of autism spectrum disorder using virtual reality and machine learning: A proof of concept for diagnosis," Autism Res., vol. 15, no. 1, pp. 131–145, 2022, doi: 10.1002/aur.2636.
[23] Y. Lin, Y. Gu, Y. Xu, S. Hou, R. Ding, and S. Ni, "Autistic spectrum traits detection and early screening: A machine learning-based eye movement study," J. Child Adolesc. Psychiatr. Nurs., vol. 35, no. 1, pp. 83–92, 2022, doi: 10.1111/jcap.12346.
[24] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford, U.K.: Clarendon Press, 1995. https://doi.org/10.1093/oso/9780198538493.001.0001
[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE,

vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[26] Swastika, W., Prilianti, K. R., Irawan, P. L. T., and Setiawan, H., "Comparative analysis of random forest and deep learning approaches for automated acute lymphoblastic leukemia detection using morphological and textural features," Jurnal Ilmiah Kursor, vol. 13, no. 1, Jul. 2025. doi: 10.21107/kursor.v13i1.427.

[27] Putri, F. N. R., Isnanto, R. R., & Sugiharto, A., "Skin Rash Classification System Using Modified DenseNet201 through Random Search for Hyperparameter Tuning," Jurnal Ilmiah Kursor, vol. 12, no. 4, Dec. 2024. doi: 10.21107/kursor.v12i4.418.

[28] Badri, F., Alawiy, M. T., & Yuniarno, E. M., "Deep Learning Architecture Based on Convolutional Neural Network (CNN) in Image Classification," Jurnal Ilmiah Kursor, vol. 12, no. 2, pp. 83–92, Dec. 2023, doi: 10.21107/kursor.v12i2.349.

[29] Sabila, S. S. and Tjahyaningtyas, H. P. A., "The Multiple Brain Tumor with Modified DenseNet121 Architecture Using Brain MRI Images," Jurnal Ilmiah Kursor, vol. 12, no. 3, 2024, doi: 10.21107/kursor.v12i3.379.

[30] Hasan Basri, I. Syarif, S. Sukaridhoto, and M. F. Falah, "Intelligent System for Automatic Classification of Fruit Defect Using Faster Region-Based Convolutional Neural Network (Faster R-CNN)," Jurnal Ilmiah Kursor, vol. 10, no. 1, Jul. 2019, doi: 10.28961/kursor.v10i1.187.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[33] [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Adv. Neural Inf. Process. Syst., vol. 30. Red Hook, NY, USA: Curran Associates, Inc., 2017.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929 .

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," arXiv preprint arXiv:2103.14030, 2021. [Online]. Available: https://arxiv.org/abs/2103.14030 .

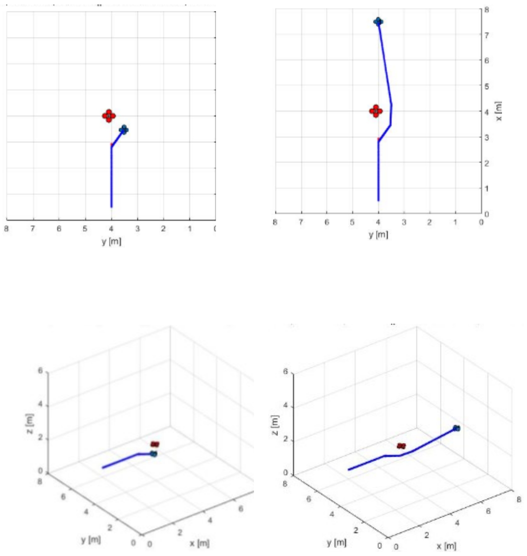
[36] F. Cilia, R. Carette, M. Elbattah, J.-L. Guérin, and G. Dequen, "Eye-tracking dataset to support the research on autism spectrum disorder," Figshare, 2022, doi: 10.6084/m9.figshare.20113592.v1.

[37] E. Schopler, R. J. Reichler, and B. R. Renner, The Childhood Autism Rating Scale (CARS) Manual. Los Angeles, CA, USA: Western Psychological Services, 1988.

[38] R. Carette, M. Elbattah, G. Dequen, J.-L. Guérin, and F. Cilia, "Visualization of eye-tracking patterns in autism spectrum disorder: Method and dataset," in Proc. 2018 13th Int. Conf. Digit. Inf. Manage. (ICDIM), 2018, pp. 248–253, doi: 10.1109/ICDIM.2018.8846967.

[39] S. S. Dcouto and J. Pradeepkandhasamy, "Multimodal deep learning in early autism detection—Recent advances and challenges," Eng. Proc., vol. 59, no. 1, p. 205, 2023, doi: 10.3390/engproc2023059205.

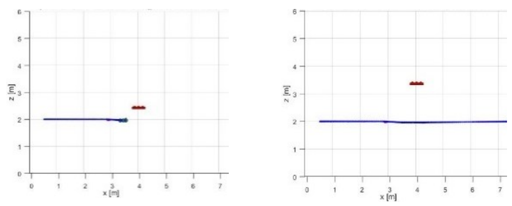
[40] N. Neethu, K. R. Remya, and B. M. Varghese, "Early detection of autism spectrum disorder via deep-learning application of fMRI and machine learning for ASD children identifications," 2024, pp. 1–15.

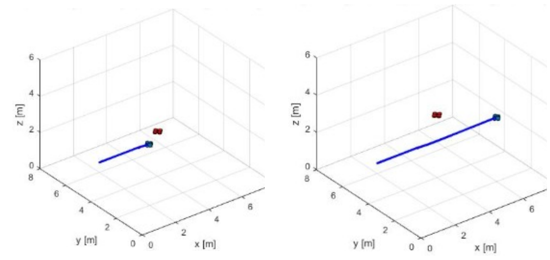


0.5	0.6	0.6	0.1	8	0.1999

12&15	0.45	0.68	0.52	0.15	0.076
12&16	0.46	0.72	0.48	0.14	0.0893

0.4	0.64	0.56	0.2	Right
0.5	0.72	0.48	0.1	Right
0.52	0.8	0.4	0.08	Right

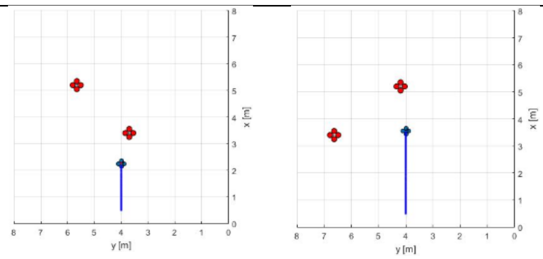


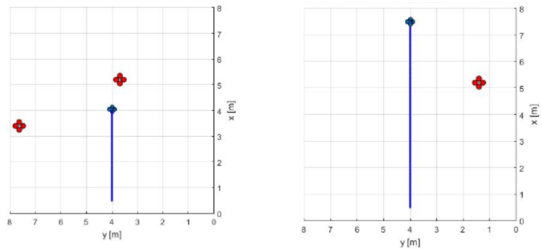


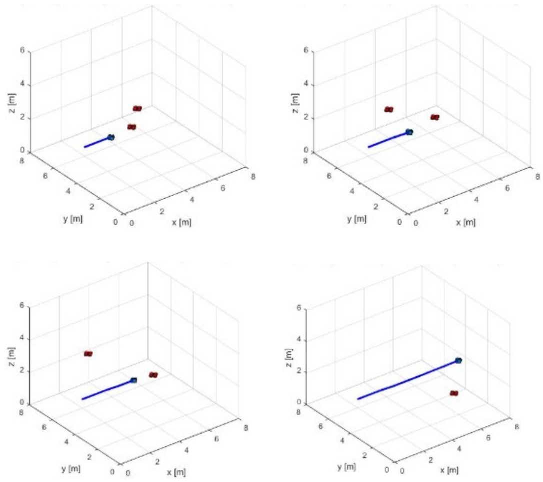
0.29	0.6	0.6	0.06	8	0.0261

11 & 12	0.3	0.6	0.6	0.05	0.0178
11 & 6	0.29	0.6	0.6	0.06	0.0343

0.3	0.56	0.64	0.05	Down
0.3	0.64	0.56	0.05	Down
0.27	0.64	0.56	0.08	Down







0.5	0.15	0.05	0.1	9		0.1106

8 & 7	0.55	0.18	0.02	0.045	0.0292
8 & 4	0.58	0.187	0.01	0.022	0.0316

0.56	0.19	0.01	0.04	Right

0.55	0.17	0.03	0.05	Right
0.6	0.19	0.01	0.002	Down

0.1	0.19	0.56	0.5	1		0.2887

5 & 6	0.04	0.07	0.67	0.55	0.2194
5 & 7	0.05	0.1	0.65	0.55	0.2317

0.042	0.05	0.7	0.558	Left
0.048	0.1	0.65	0.552	Left
0.06	0.15	0.6	0.54	Left