Diagnosing children with ASD using Computer Vision

M. Abbas Ansari

Summer Research Internship work at ITMR Lab, IIIT-A under Prof. Anupam Agarwal

Survey of the field

Taken from a literature review [1]

de Belen, R.J., Bednarz, T.P., Sowmya, A., & Del Favero, D. (2020). **Computer vision in autism spectrum disorder research: a systematic review of published studies from 2009 to 2019**. Translational Psychiatry, 10.

Different Behavioral/Biological Markers

1. Magnetic Resonance Imaging (MRI/fMRI/EEG) [Brain activity]
2. **Eye Gaze data** [Visual Attention]
3. **Facial Expression**/Emotion
4. Motor Control/Movement Pattern
5. Stereotyped Behaviours
6. **Multimodal data**

Above are the different markers that have been used to detect a signal from using computer vision to diagnose children/people with ASD uptil now.

Bolded markers are relevant to our work

Relevant Downloadable Datasets

Eye-Gaze Data/Visual Attention:

1. Saliency4ASD:
   Dataset of eye movements [2] of children with Autism Spectrum Disorder. It consists of 300 natural scene images and the corresponding eye movement data collected from 14 children with ASD and 14 healthy controls. In particular, fixation maps and scanpaths are available in the dataset. 200 test images with ground truth is also available on request.
2. Kanner autism: MIE Fo and MIE No:
   two new eye-tracking datasets of ASD [3] people in order to cover a large part of the autism spectrum, going from high-level functioning (e.g. Asperger) to low-level functioning (e.g. Kanner) autism
3. SALICON: (non-ASD)
   SALICON dataset [4] offers a large set of saliency annotations on the popular Microsoft Common Objects in Context (MS COCO) image database. The current release comprises 10,000 training images and 5,000 validation images with saliency annotations. For training and validation sets, we provide the color images in JPG format, image resolution, and ground truth (including gaze trajectory, fixation points, and saliency map). The test set with 5,000 images is released without ground-truth. All images are selected from the 2014 release of the COCO dataset.
4. MIT Saliency Benchmarks: (non-ASD)
   MIT300 [5]- This was the first data set with held-out human eye movements, and is used as a benchmark test set.
   CAT2000 [6]- This dataset contains two sets of images: train and test. Train images (100 from each category) and fixations of 18 observers are shared but 6 observers are held-out. Test images are available but fixations of all 24 observers are held out.

The great availability of large amount of non-ASD specific datasets allows for pretraining of deep-learning based networks, which then can be fine-tuned on the ASD based datasets.

Relevant Downloadable Datasets

Face Analysis:

1. ### Autistic Children dataset:

   It was uploaded to kaggle. It had 2940 autistic and non-autistic images. This dataset has been **deleted by Kaggle.** There have been a few research papers published on the dataset, but the dataset was collected through unethical means i.e. no consent was taken from the children involved since all of the data was collected through web-scraping. Also, there are lots of children with down syndrome in the dataset which is a very distinct neurological atypicality from ASD, hence any ML model trained would learn the wrong features to identify children with ASD. There are many duplicates as well.
   Reference: https://www.kaggle.com/code/melissarajaram/concerns-with-detect-autism-dataset/notebook

Trying to do scholarly work with such a dataset is unethical. There are no other downloadable datasets which involves facial analysis of ASD children. Thus, research cannot be done in this direction.

Initial Review of Papers

Attention-based Autism Spectrum Disorder Screening with Privileged Modality [7]

Shi Chen, Qi Zhao

Focus:

Using two visual attention privileged modalities:
1. Photo-taking task
2. Image-viewing (fixation)

Methodology:

[Independent Training]
1. To capture the signal that photos taken by PASD have different characteristics, sequence of photos were taken by subjects and fed into an LSTM. final hidden state classified as 0/1. [PT]
2. Through image viewing task, sequential fixations on different locations was tracked. Features of image extracted using CNN, and a feature vector is extracted for closest location to each fixation which is fed sequentially to an LSTM. final hidden state classified as 0/1 [IV]
[Shared space learning]
3. Embedding layers for each IV and PT Encoders are made for a shared space transfmtn, and a common classifier is attached to the embeddings. Encoders weights fixed, and embedding and common classifier is trained.
[Distillation from shared space]
4. Different modalities are disentangled by unfreezing encoders and freezing embedding and common classifier to distil knowledge from the shared space.

Architecture & Training

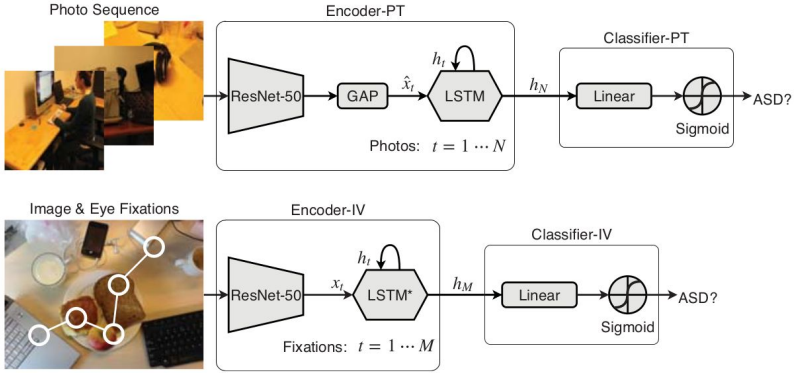


Figure 1: High-level architectures for attention based ASD screening models on photo-taking (top) and image-viewing (bottom) modalities. GAP denotes the global average pooling layer. $\hat{x}_t$ in photo-taking is the features for image t, while in image-viewing x_t is the features extracted at the proximity of fixation t. N and M represent the number of images and fixations in photo-taking and image-viewing data.

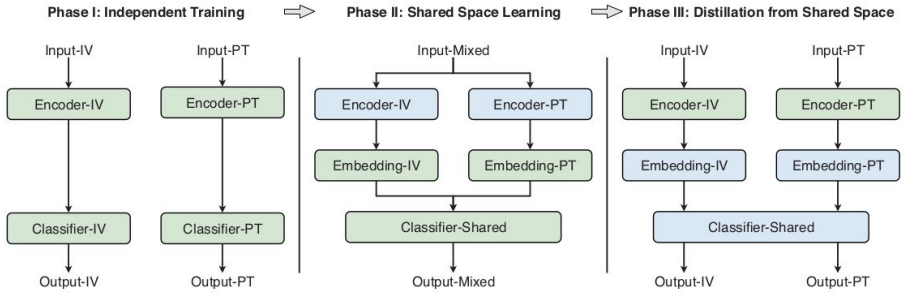


Figure 3: Process for the proposed ASD screening with privileged modality framework. Different training phases are highlighted with bold text at the top. Encoder-IV, Encoder-PT, Classifier-IV and Classifier-PT are the same as those presented in Figure 1. Modules with blue color are fixed during a training phase while those colored in green are being optimized.

Dataset & Results

For photo-taking, 22 individuals with ASD and 23 controls. They were instructed to take photos in both indoor and outdoor scenarios, and each took 40 photos on average. For image-viewing eye-tracking data from 20 ASDs and 19 controls.

Used Saliency4ASD dataset for benchmarking

	Acc.	Sen.	Spe.	AUC
Liu *et al.* [22]	0.89	0.93	0.86	0.89
Jiang *et al.* [17]	0.92	0.93	0.92	0.92
IV-Independent	0.97	1.00	0.95	1.00
IV-Full	**0.99**	**1.00**	**0.98**	**1.00**
IV-Independent (Saliency4ASD)	0.89	0.86	0.93	0.92
IV-Full (Saliency4ASD)	**0.93**	**0.93**	**0.93**	**0.98**
Human Expert [35]	0.65	-	-	-
PT-Independent	0.76	0.77	0.74	0.82
PT-Full	**0.84**	**0.77**	**0.91**	**0.84**

Table 1: Inter-model comparison on ASD screening. Results on our image-viewing dataset, Saliency4ASD [9] and our photo-taking dataset are divided by the horizontal lines and listed from top to bottom. IV-Independent and PT-Independent are our single-modal models on image-viewing and photo-taking. Our full models with multi-modal distillation are denoted as IV-Full and PT-Full for both modalities. Four evaluation metrics are used, including Accuracy (ACC.), Sensitivity (Sen.), Specificity (Spe.) and AUC. Best results are highlighted in bold text.

Visual Attention Analysis and Prediction on Human Faces for Children with Autism Spectrum Disorder [8]

Huiyu Duan, Xiongkuo Min, Yi Fang, Lei Fan, Xaiokang Yang, and Guangtao Zhai

Focus:

Analyse and predict the visual attention of children with ASD when looking at human faces.

Methodology:

Constructed VAFA Database. Quantified effect of face size, facial features, face pose, facial expressions for resulting in different fixation points on faces between ASD and TD subjects.

Features extracted from face image by:

i. A gaussian kernel placed at center of face, left eye, right eye, nose and mouth region to extract features.

ii. Using VGG-16 to extract features from face image.

The features are then concatenated and fed into CASNet which predicts saliency map for the face image.

Architecture

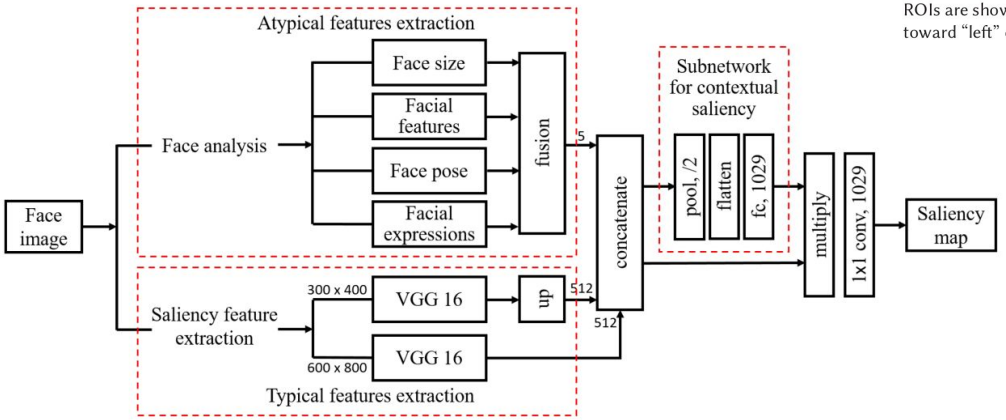


Fig. 9. Flowchart of our proposed method. "5" represents five feature maps we extracted, including face, left eye, right eye, nose, and mouth, since they are the most salient regions in natural images with faces in them.

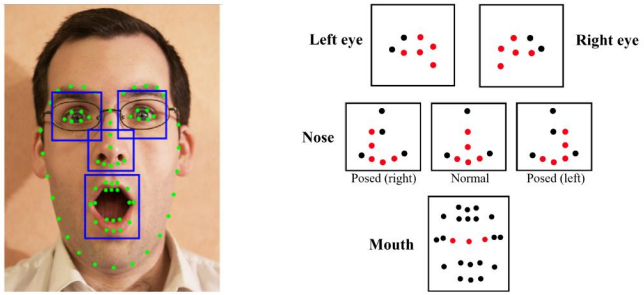


Fig. 10. Example of facial features extracted from stimuli. Chosen facial marks (marked as red points) for all ROIs are shown. Note that when the face pose changed, we use different features for the nose region. Posed toward "left" or "right" is defined based on the viewer's coordinate system.

pose is turned to the left or right. Then we calculate the feature map by placing a uniform Gaussian kernel at each extracted feature point and the feature map of region k can be expressed as

$$S_k(x) = \mathcal{N}\left(\sum_p \exp\left[-\frac{(x-x_p)^2}{2\sigma^2}\right]\right), p \in \mathbb{P}_k, \tag{2}$$

$$\sigma = w_p \cdot w_e \cdot 40, \tag{3}$$

where k represents the ROI (left eye, right eye, nose, or mouth region). p denotes each feature point in region k and $\mathbb{P}_k$ denotes the feature point sets in region k. x can be any two-dimensional

Results

Table 7. Results on Testing Set of the VAFA Database

Models	AUC		sAUC		CC		NSS	
	Original	Fine-tuned	Original	Fine-tuned	Original	Fine-tuned	Original	Fine-tuned
SALICON [29]	0.7856	0.8087	0.5419	0.5552	0.5628	0.6448	1.3816	1.4237
mlnet [15]	0.8175	0.8186	0.5509	0.5598	0.6768	0.6955	1.5957	1.6011
SAM-VGG [16]	0.8297	**0.8369**	0.5529	0.5644	0.7171	**0.7710**	1.6900	1.7594
SAM-ResNet [16]	0.8288	0.8155	0.5595	0.5585	0.7537	0.6873	**1.7764**	1.5838
SlaGAN [49]	0.8182	0.8256	0.5824	0.5752	0.6926	0.7422	1.5654	1.6811
CASNet [25]	0.8272	**0.8376**	**0.5825**	**0.5832**	0.7283	**0.7791**	1.6418	**1.7812**

"Original" represents the original model designed for healthy people. "Fine-tuned" represents the model fine-tuned based on the SPCA database and VAFA database. AUC, sAUC, CC, and NSS are used to evaluate the performance of these models. We highlight the best two results under each evaluation criterion in bold.

Face-Based Attention Recognition Model for Children with Autism Spectrum Disorder [9]

Bilikis Banire, Dena Al Thani, Marwa Qaraqe and Bilal Mansoor

Focus:
Recognising attention/inattention in children with ASD based on their facial features

Methodology:

A dataset was created where children with ASD/TD were asked to attend to a screen. Videos were recorded of their face through a webcam, and each frame was annotated as attention/inattention by the researchers.

1. For each face of child, facial landmarks were extracted using landmark detection algorithms. Coordinates of landmarks were the features that were used downstream.

2. Best geometrical information which distinguishes attention/inattention frame were identified using distance-based thresholding. The best information was used as features for SVM classifier.

3. A picture where facial landmarks were white dots against a black background was used as input to a CNN classifier.

SVM Feature Selection

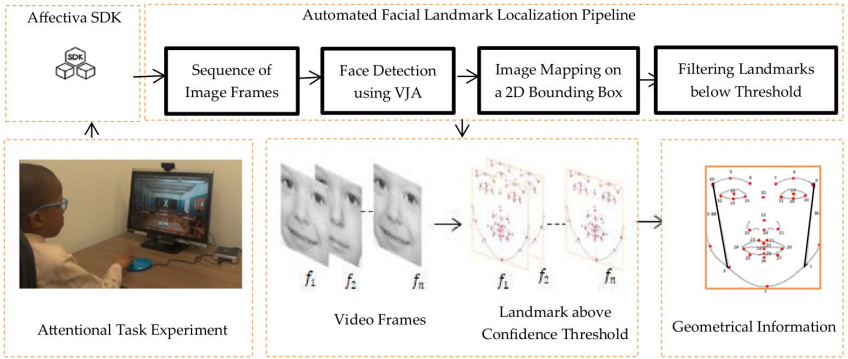


Fig. 2 Block diagram of geometric-based feature extraction

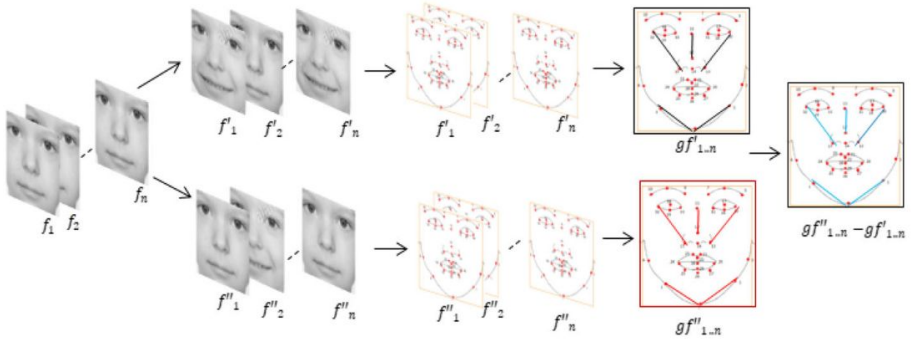


Fig. 4 Feature selection process

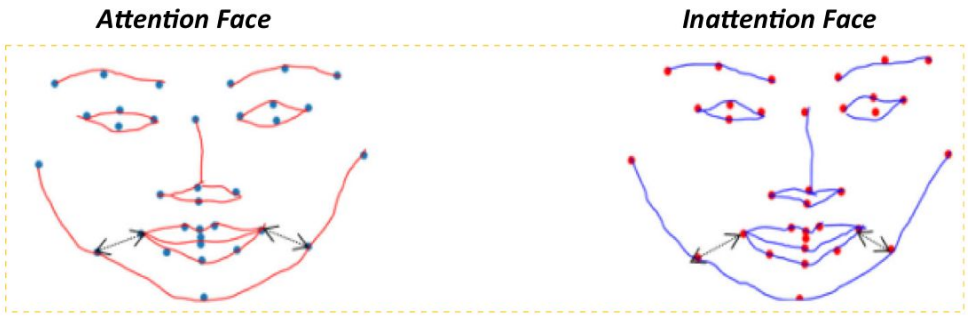


Fig. 5 Mean intensity frame for attention and inattention

CNN Architecture

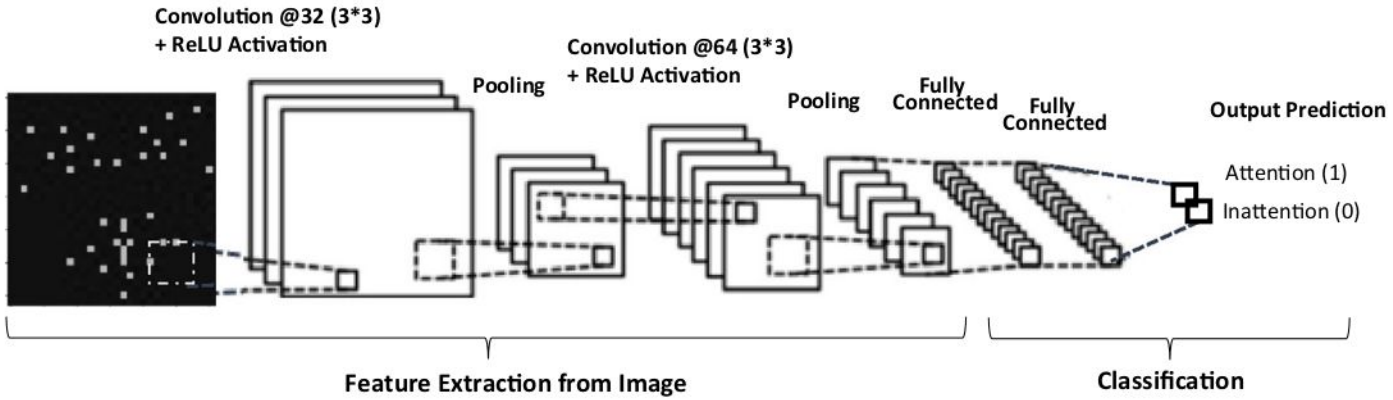


Fig. 6 Attention classification using a CNN model structure

The structure of the CNN architecture has an input image generated from time-domain spatial features (i.e., the frame by frame facial landmarks coordinates) to 2D spatial images with the size of 32 by 32 pixels.

The facial landmarks are represented as white dots on a black background to reduce the noise of the image

Results

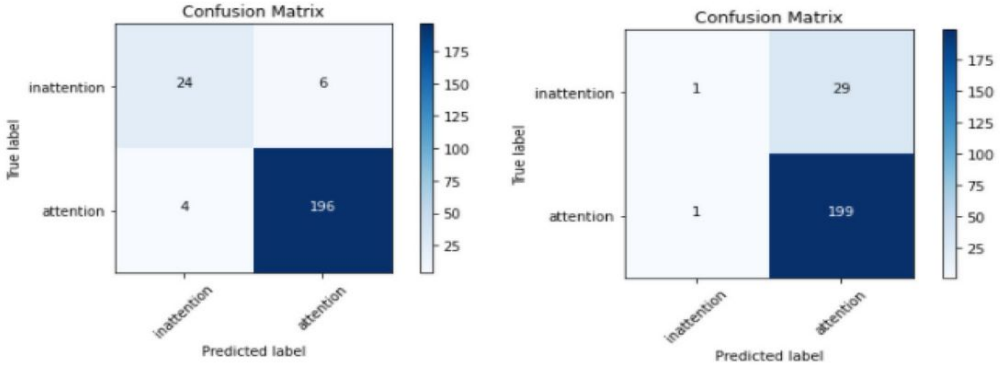


Fig. 7 Confusion matrix for participant-specific model for P1 (left: SVM; right: CNN)

Hampered the success of CNN by giving it a very limited signal (just a bunch of white dots against a black background).

Also the architecture is very basic, hence has limited representational capacity. Here we CNN only predicts attention which is a skewed datapoint in the dataset. {Children are more attentive than inattentive}

Deep Learning for Autism Diagnosis and Facial Analysis in Children [10]

Mohammad-Parsa Hosseini, Madison Beary, Alex Hadsell, Ryan Messersmith and Hamid Soltanian-Zadeh

Focus:
Classifying a child with ASD based on their face image

TABLE 4 | Dataset breakdown.

Data set	Composition	Overall data composition %
Train	1,327 autistic 1,327 healthy	88
Validation	80 autistic 80 healthy	5.3
Test	140 autistic 140 healthy	9.3
Total	1,507 autistic 1,507 healthy	100

FIGURE 2 | Some images used in the deep learning training step. **(Top)** Children who have autism. **(Bottom)** Children who do not have autism.

Architecture & Results

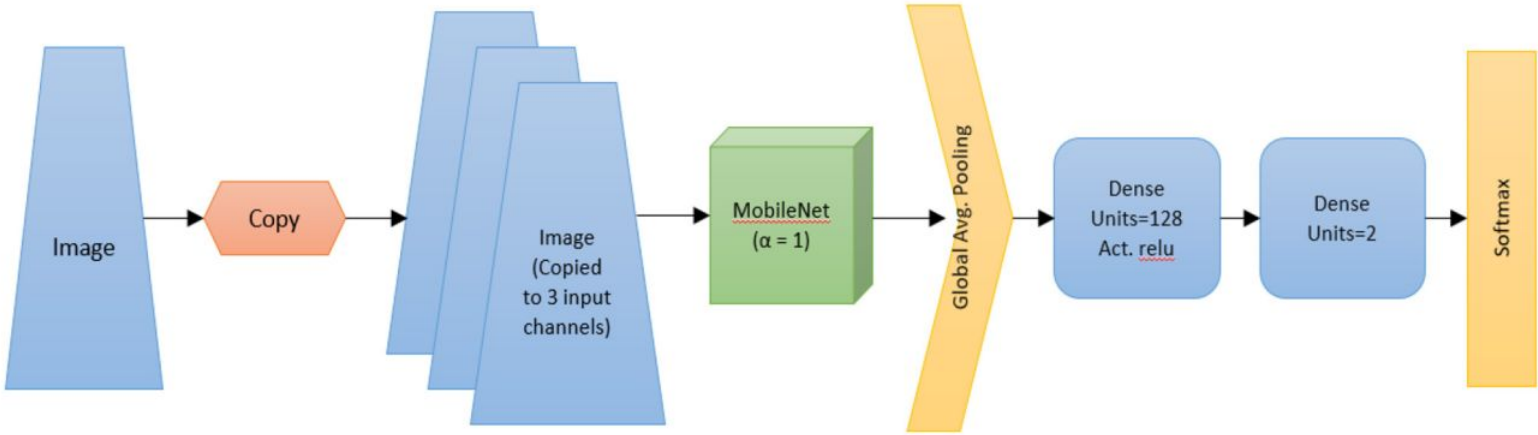


FIGURE 3 | The algorithm architecture of the proposed model, illustrating the use of MobileNet, followed by two dense layers to perform image recognition. MobileNet uses CNN to predict what is the shape of the object present and what is matched with it from the images.

The training was completed after ~15 epochs, yielding a test accuracy of **94.64%**

Prediction of Autism Spectrum Disorder in Children using Face Recognition [11]

Sajeev Ram Arumugam, Balakrishna R, Rashmita Khilar, Oswalt Manoj and Shylaja CS

Focus:
Classifying a child with ASD based on their face image

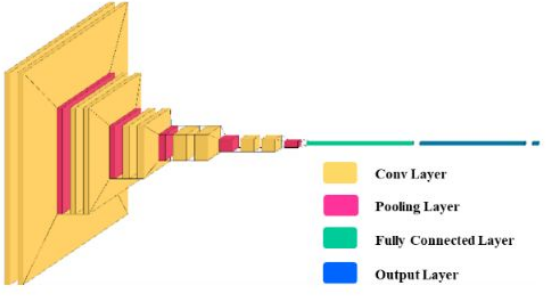


Fig. 2. Visualization of the proposed Neural Network

VGGFace Model Architecture

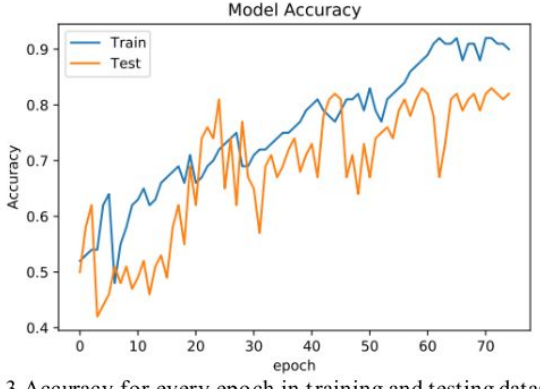


Fig. 3 Accuracy for every epoch in training and testing dataset

When epoch reaches 70, the system accuracy is 91%, and then the system starts to overfit

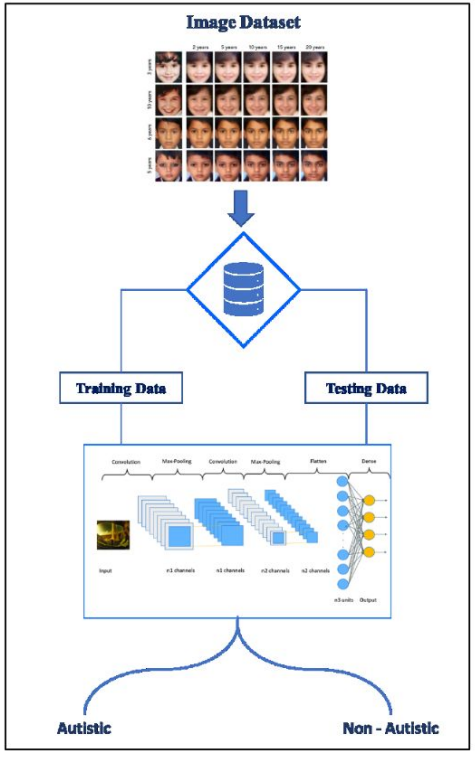


Fig. 1 The architecture of the proposed ASD detection

Some Conclusions

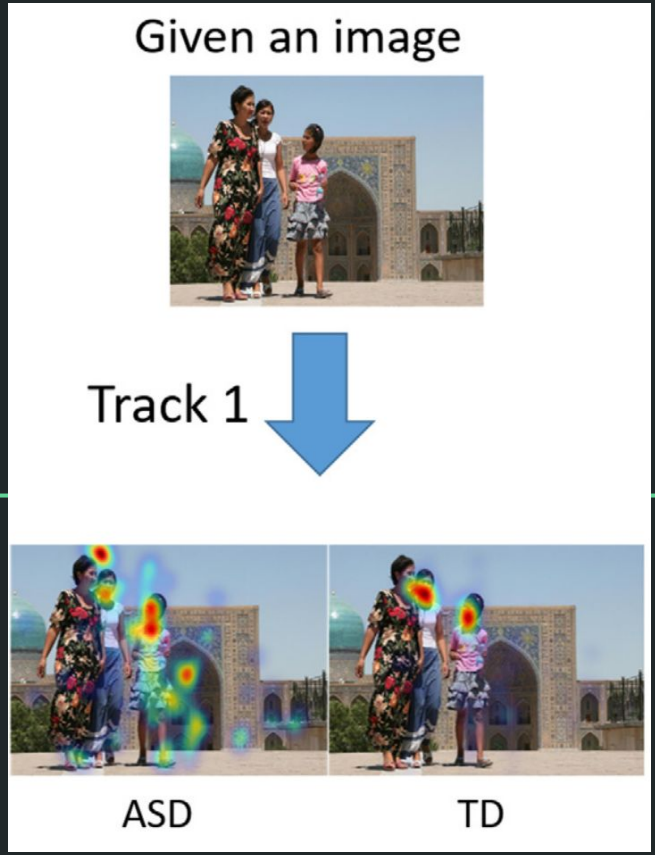
1. Due to the sensitive nature of the data i.e. it involves children with ASD, it requires the consent of parents and relevant medical authority to collect and experiment with data. **Most dataset is therefore not public**. [Biggest Drawback]

2. Lot of work done in facial analysis of ASD children involves using **a Kaggle dataset which has been deleted by Kaggle**. The reason for deletion can be that no consent has been taken from the parents of the children and the quality of the dataset is very bad since it has lots of duplicates and also children with down syndrome which is a very distinct neurological disease compared to ASD.

3. **The most accessible dataset is Saliency4ASD**. It is specifically made for the task of trying to classify fixation maps and scanpaths of a child on scene images into having ASD or not. The dataset comprises 14 ASD and 14 TD samples. The task is broadly under the field of saliency/visual attention modelling. A lot of work has been done and there are several datasets available like SALICON. Trying to come up with a novel deep-learning technique can be the goal of possibly pursuing this is as a research direction. To beat the SOTA on Saliency4ASD dataset.

Survey of Saliency4ASD Challenge [12]

Track 1:
Predicting Saliency Maps for children with ASD [12]



Saliency Prediction via Multi-Level Features and Deep Supervision For Children with Autism Spectrum Disorder [13]

Weijie Wei, Zhi Liu, Lijin Huang, Alexis Nebout and Olivier Le Meur

Methodology:

A backbone extracts features. From different depths of the backbone, feature maps go through dilated convolutions and then deconvolutional operations to give feature maps as same size of inputs called AMi. During the formation of each AMi, deep supervision is done using a learned prior map controlling the 2D Gaussian distribution to fit the location-bias of the dataset.

Three multi-level feature maps AM1, AM2, AM3 are also futher convolved 1x1 to give saliency map SM. Single-side clipping is performed on ground truth saliency maps. By using SSC, the salience of the regions where most ASD observers pay attention are reserved whereas lesser observed compared to TD are suppresed.

Using the SSC GT, loss is computed accross the 4 feature maps. It is a sum of CC, KL, Pearson Coefficient and NSS.

Pretrained on MITI1003 and then finetuned on Saliency4ASD

Architecture

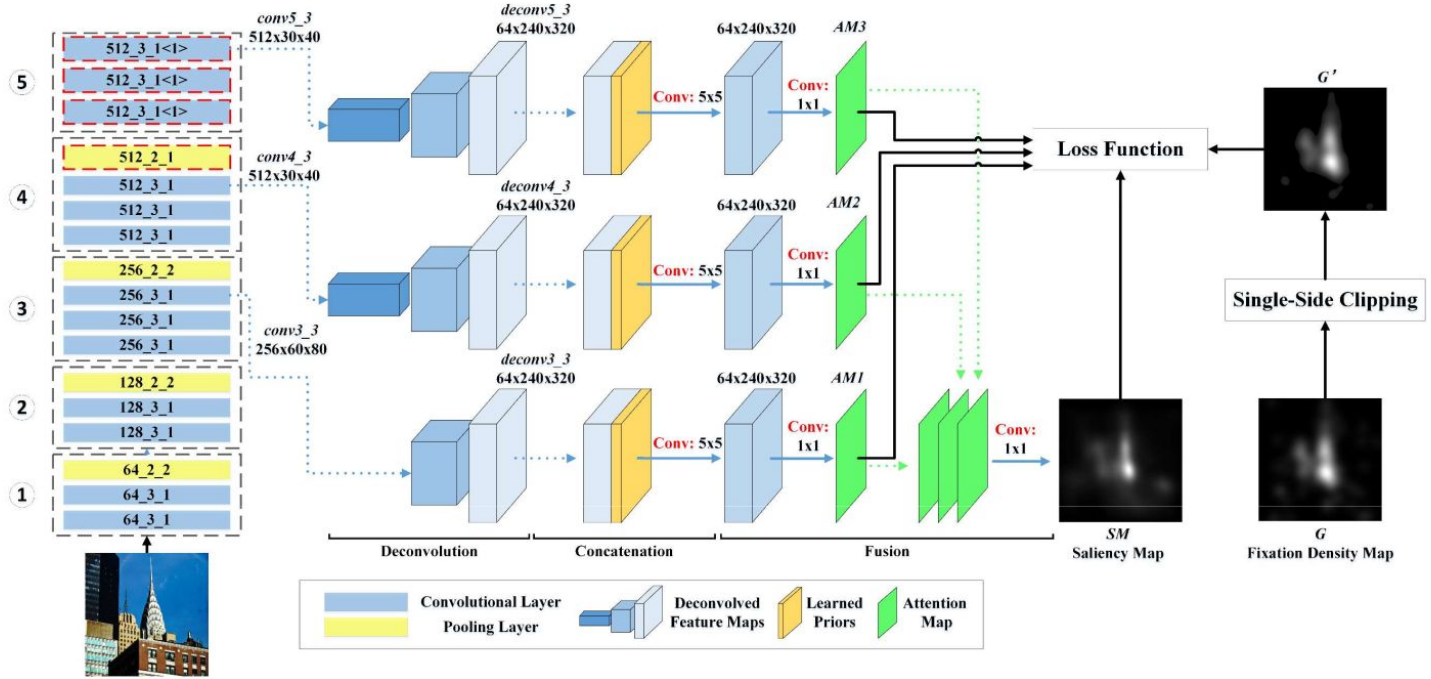


Fig. 1. Overview of the proposed network architecture. The left part is the dilated convolutional network (DCN) where the layers are expressed in terms of *channels_kernel_stride<holes>*. The red dashed boxes indicate the modified layers.

Results

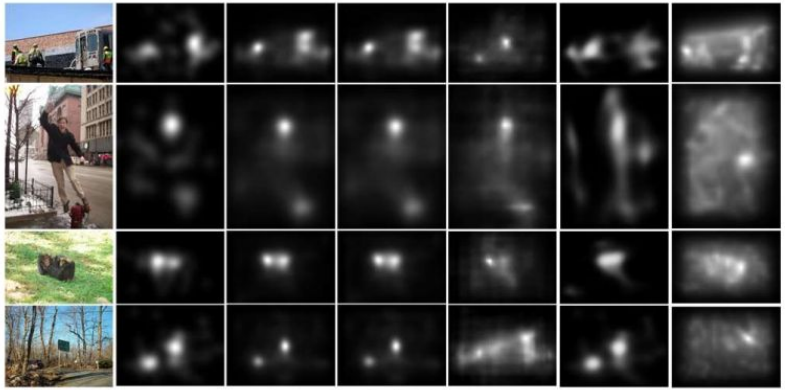


Fig. 2. Qualitative comparison with other state-of-the-art models.

Table 1. Comparison with state-of-the-art models on the test set with 30 images in the dataset [21]. The best two scores are marked in **bold** and underlined.

Model	SIM	CC	KL	NSS	AUC-J
GBVS	0.599	0.554	0.543	0.992	0.764
SalGAN	0.635	0.687	1.565	1.307	0.783
SAM-VGG	0.643	0.705	0.586	1.377	0.797
Our(w/o SSC)	0.671	0.734	0.465	1.459	0.808
Our	**0.678**	**0.769**	**0.421**	**1.738**	**0.834**

Predicting Saliency Maps for ASD People [14]

Alexis Nebout, Weijie Wei, Zhi Liu, Lijin Huang, and Olivier Le Meur

Methodology:

From a full size input image, features are extracted using VGG-16 to get fine-scale information and a half size input image features are also extracted using VGG-16 to get coarser-scale information. These different features maps are concatenated and using a 2D Locally convolutional layers, relative importance of each map is decided [Channel Attention] and then finally saliency map is determined

Pretraining on MITI1003 and then finetuning on Saliency4ASD gives best results.

Fig. 3: Different data augmentation methods. From left to right: original, blurred, flipped, noisy and grayscale image.

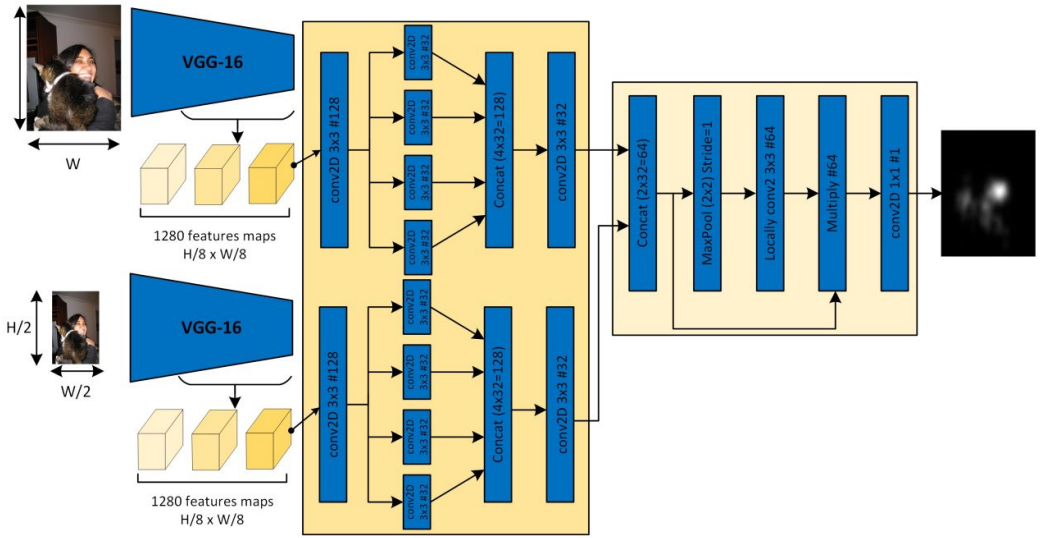


Fig. 1: Proposed deep architecture.

Flawed data augmentation technique (possibly)

Results

Table 1: Performance of the proposed model and comparison with existing saliency models. SalGAN$^{(*)}$=SalGAN model finetuned by MIT1003 and training dataset. RI= Random Initialization; FT=Fine-Tuning. The last line present the results of FT-M2 on test set of 200 natural images from [19]. Results on bold show best scores, while results on italic in the last line show the greater score on the other dataset.

Model	SIM ↑	CC ↑	KL↓	NSS ↑	AUC-J ↑	AUC-B ↑
Existing models						
RARE2012	0.5317	0.4240	0.7754	0.8632	0.7224	0.7058
Hou	0.5304	0.3934	0.7127	0.7452	0.7088	0.6940
AWS	0.5178	0.3777	0.8024	0.7551	0.6973	0.6897
SUN	0.4834	0.2442	0.8842	0.5144	0.6376	0.6301
GBVS	0.5990	0.5541	**0.5426**	0.9919	0.7642	0.7555
SAM-VGG	0.5453	0.5961	3.3719	**1.3182**	0.7758	0.6576
SalGAN$^{(*)}$	0.6353	0.6866	1.5651	1.3074	0.7829	0.7551
Proposed methods						
RI - M1	0.6237	0.6808	2.5995	1.2709	0.7833	0.7520
RI - M2	0.6211	0.6587	2.0173	1.2140	0.7810	0.7540
RI - M3	0.5833	0.5655	1.9500	0.9714	0.7573	0.7297
FT - M1	**0.6590**	**0.6983**	0.9480	1.2637	**0.7955**	**0.7739**
FT - M2	0.6099	0.5883	0.6368	1.0274	0.7712	0.7468
Results on other dataset						
FT - M1	0.6308	0.6822	0.9023	*1.4193*	*0.8106*	*0.7850*

Fig. 4: Predicted saliency maps for the different tested saliency models. From left to right, first row: original image and ground truth; second row presents the predictions from RARE2012, Hou, AWS, SUN, GBVS, SAM-VGG. From left to right, third row presents the predictions from SalGAN, RI-M1, RI-M2, RI-M3, FT-M1, FT-M2. The red frame indicates the prediction having the highest CC.

Visual attention prediction for Autism Spectrum Disorder with hierarchical semantic fusion [15]

Yuming Fang, Haiyan Zhang, Yifan Zuo, Wenhui Jiang, Hanqin Huang, Jiebin Yan

Methodology:

1. Features extracted using VGG-16. These features are then used in Spatial
Feature Module [SFM] and a pseudo sequential Feature Module (PSFM) which involves 2 ConvLSTMs. The features from different modules are fused by the heirarchical semantic fusion module (ASD-HSF).

2. Pretraining done on TD adult saliency map datasets: MITI1003 and CAT2000,
model is finetuned on Saleincy4ASD dataset.

3. During pre-training KL-divergence and MSE used as loss function. whereas during fine-tuning, KL-divergence and PN-MSE is used as loss.

of ground truth maps, we designed a new loss function called Positive and Negative Equilibrium Mean Squared-Error (PN-MSE) to help train the network, as follows

$$L_{PN-MSE} = \frac{1}{2}\left(\frac{1}{M^+}\sum_{i\in C^+}(\boldsymbol{G}_i - \boldsymbol{S}_i)^2 + \frac{1}{M^-}\sum_{j\in C^-}(\boldsymbol{G}_j - \boldsymbol{S}_j)^2\right), \quad (1)$$

where L_{PN-MSE} represents PN-MSE loss. $\boldsymbol{G}$ denotes ground truth map and $\boldsymbol{S}$ represents predicted saliency map. C^+ is the set of indexes of points with positive values in $\boldsymbol{G}$ and C^- is the set of indexes of points with zero value in $\boldsymbol{G}$. M^+ and M^- denote the number of elements in C^+ and C^-, respectively.

Architecture

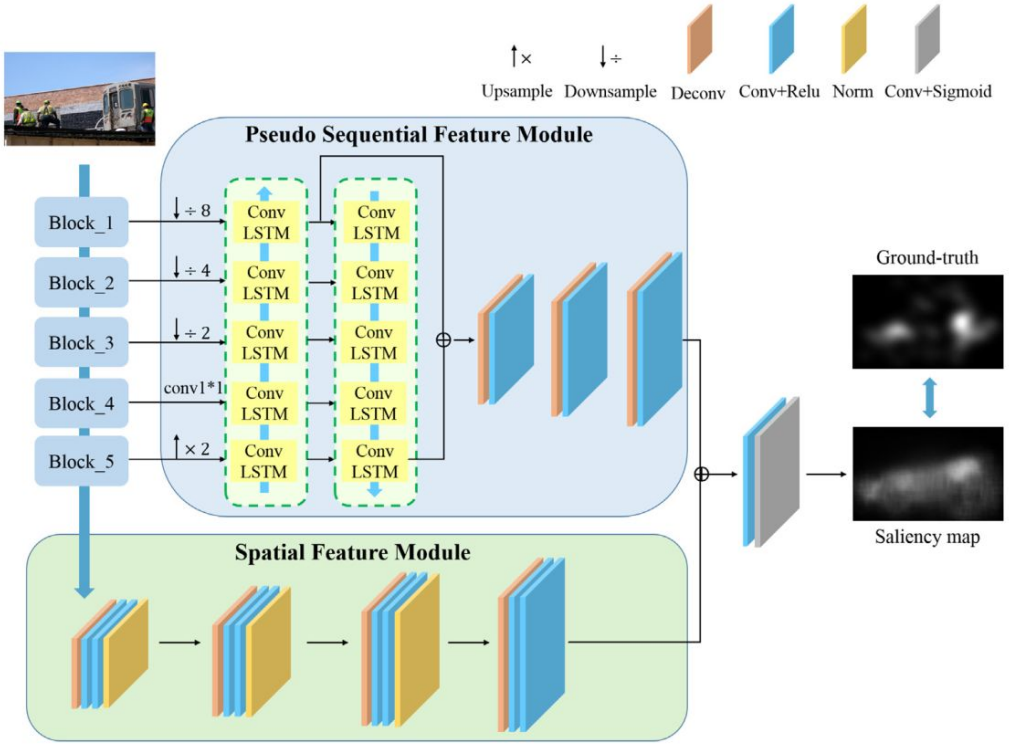


Fig. 3. The overall model. The model mainly consists of Spatial Feature Module and Pseudo Sequential Feature Module. Spatial Feature Module regards an image as a single input and utilizes a fully convolutional network to extract spatial semantic features. Pseudo Sequential Feature Module regards an image as a pseudo sequence and adopts two ConvLSTMs to extract pseudo sequential features. Then the spatial features and pseudo sequential features are fused to generate the final saliency map.

Results

First place in Saliency4ASD challenge for track 1 i.e. predicting saliency maps for ASD children.

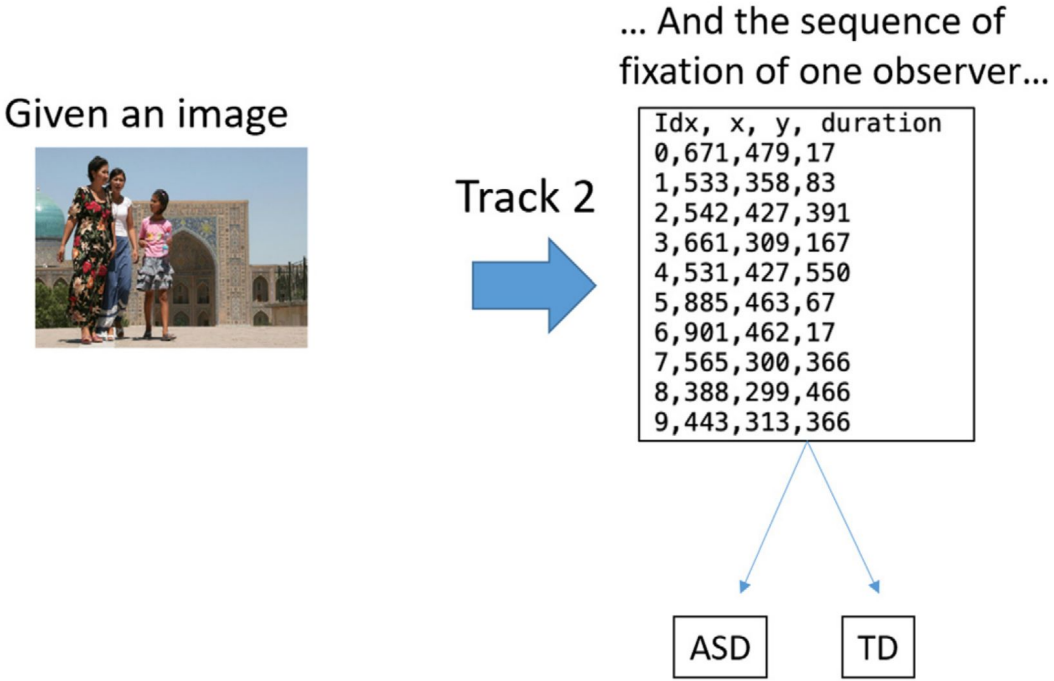
Table 1
Comparison with ten state-of-the-art general saliency prediction methods on Test set. The best results are bolded.

Methods	CC	SIM	NSS	KL	AUC_J	AUC_B
COV [11]	0.5711	0.5755	1.3660	1.0572	0.8781	0.7858
GBVS [14]	0.5561	0.5847	1.2557	0.5936	0.8584	0.8057
HFT [12]	0.5071	0.5602	1.2279	0.6645	0.8419	0.7828
PQFT [13]	0.2793	0.4687	0.6835	0.9703	0.7182	0.6534
MLNet [18]	0.5483	0.5492	1.4430	3.9224	0.8198	0.7459
DeepGazeII [47]	0.6260	0.6018	1.6857	0.9830	0.8996	0.7916
Sal-DCNN [48]	0.6908	0.5406	1.8021	5.9772	0.8978	0.7543
DINet [49]	0.6592	0.5585	1.7239	3.9798	0.9154	0.7429
SalGAN [50]	0.7076	0.6235	1.8771	1.5999	0.9166	0.8262
MSINet [51]	0.7142	0.6440	1.8279	0.6452	0.9311	0.8149
Ours	**0.7659**	**0.6815**	**1.9548**	**0.3946**	**0.9321**	**0.8653**

Table 2
Comparison with ASD saliency prediction methods on Saliency4ASD benchmark dataset. The best two results are shown in red and blue.

Methods	CC	SIM	NSS	KL	AUC_J	AUC_B
ASD_Wei [17]	0.6807	0.6231	1.5103	0.5904	0.8179	0.7864
ASD_Nebout [39]	0.6822	0.6308	1.4193	0.9023	0.8106	0.7850
ASD_Fang [38]	0.6000	0.5886	1.2446	0.6320	0.7899	0.7689
Ours	0.7020	0.6405	1.4656	0.4720	0.8175	0.7901

Track 2:
Classifying a fixation sequence as ASD or TD's fixation [12]



Submitted models for Track 2 and brief description of the proposed approaches.

Team		Model
Technical University of Munich (TUM)	[16]	Random forest classifier using features from: scanpaths, saliency (computed with SAM-ResNet) and image content (using a CNN-based face detector).
Roma Tre University (R3U)	[17]	TreeBagger classifier (based on random forest) using features from: image content (using YOLO object detector), saliency (based on SDSP), fixations, and center bias.
University of Miami (UM)	[18]	Network based on CNN and Long Short-Term Memory (LSTM) architectures. SalGAN model is first used to estimate image saliency. Three different models were proposed combining two variants of CNN structures and with/without batch normalization.
Univ. of California Davis & Univ. of Kentucky (UCD&UK)	[19]	Two models were proposed based on a fully connected dense network (FCN) trained on real and synthetic (using STAR-FC) scanpaths, one using a small set of high-level features and the other using all available info. A third model was proposed using an architecture with two branches (based on Resnet) to extract image features and to process data points.
East China Normal University (ECNU)	[20]	Simple classification model based on gaze-deviation distance (using a non-parametric visual model to obtain image saliency) and gaze duration time. Three versions of the model were submitted changing the weights for these two elements.

Performance of the submitted models for the metrics used for classification performance in Track 2. For all the metrics, a higher value indicates a better performance. The last column shows the final ranking of the participating teams resulting from the point-based strategy described in Section 6.

Team	Acc.	Recall	Precision	F1	Cohen's κ	AUC	Specificity	Team Rank
TUM	0.598	0.717	0.574	0.632	0.201	0.644	0.484	1
R3U	0.593	0.684	0.570	0.616	0.189	0.595	0.506	2
UM	0.557	0.877	0.532	0.658	0.127	0.564	0.251	3
	0.579	0.592	0.563	0.570	0.158	0.579	0.566	
	0.574	0.594	0.568	0.568	0.149	0.575	0.556	
UCD&UK	0.551	0.635	0.527	0.546	0.106	0.613	0.471	4
	0.542	0.741	0.522	0.610	0.091	0.575	0.351	
	0.539	0.807	0.519	0.629	0.089	0.544	0.282	
ECNU	0.516	0.705	0.504	0.585	0.041	0.521	0.337	5
	0.446	0.397	0.429	0.412	−0.110	0.445	0.493	
	0.420	0.442	0.413	0.427	−0.159	0.421	0.399	

Best Deep Learning Method: on Track 2:

SP-ASDNET: CNN-LSTM based ASD Classification Model using Observer Scanpaths [18]

Yudong Tao, Mei-Ling Shyu

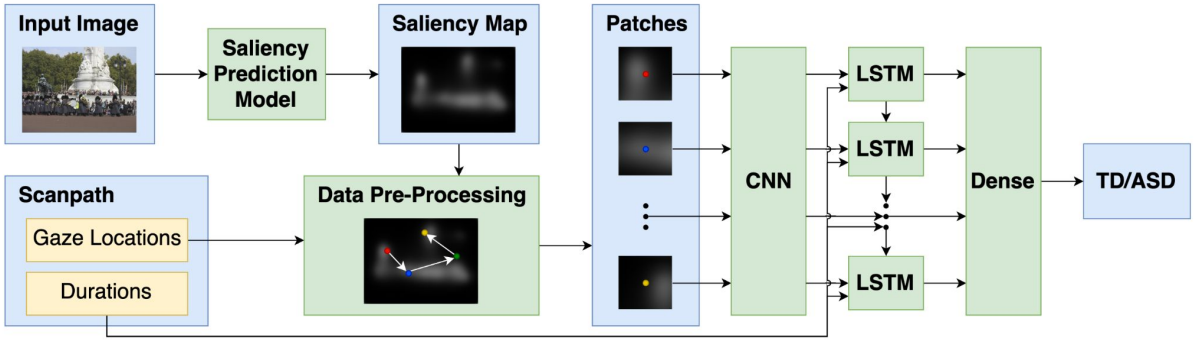


Fig. 1. The proposed SP-ASDNet framework.

Methodology:
A saliency map is extracted using SalGAN [21] (trained on SALICON dataset [4]). A patch of saliency map is extracted around the gaze location and fed into a shallow CNN which is then fed into LSTMs. Gaze locations' visual embeddings are concatenated with respective duration feature and are then sequentially processed using an LSTM. Outputs of each cell are then fed through dense layers for classification.

Week 1 Progress
12/6/22 - 18/6/22

1. Read through all of track 2 papers
2. Implementing Chen 2019's work in google colab and understanding the code thoroughly
3. Loading Saliency4ASD dataset and exploratory data analysis was done.

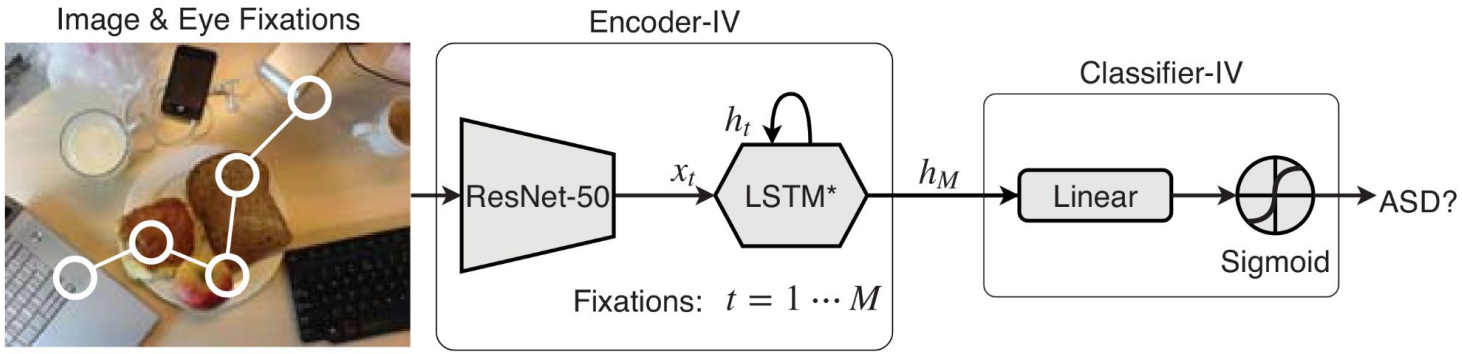
1. **Read through all of Track 2 papers:**

   Understood that the best methods currently involve using ensemble methods (Random Forests and Decision Trees) and extracting lots of hand-crafted features.

   Deep Learning methods are being possibly beaten by Random Forests due to limited amount of data to train a classifier. DL methods require lots of data to give comparable results to random forests. To fix the issue, we must make use of **transfer learning**. To train our DL network on a lot of data and then fine-tune on the saliency classification dataset.

   The best deep learning methods perform sequential processing of gaze locations using LSTMs. They also try to exploit duration as a feature to train the network. The idea of sequential processing is **sound**, in my opinion. The analysis in TUM paper [18] revealed that **duration was a very important feature** for ASD/TD classification.
   The current methods do a bad job of encoding duration as a feature for a deep learning network, hence **a better time-sensitive DL network is needed** to better exploit the duration feature.

2. Replicating Chen 2019 network as a baseline



PyTorch code was available for the above network architecture on github. The codebase was understood and then replicated into Colab notebook. Debugging was done to understand the inner workings of the Dataloader and Model architecture.

It used leave-one-out cross validation. Its statistical significance and how it has been implemented in code was understood.

Colab Notebook Link:
https://colab.research.google.com/drive/12t4zeDJT_1kZjsbPfhuN4Vd4fCCAk75y?usp=sharing

3. **Saliency4ASD EDA:**

The dataset was downloaded into a colab environment using the wget library. How to load the dataset using python was understood [Code from Shi Chen].

EDA was done to understand that the data was loaded from csv files into a python dictionary. There are 14 ASD and 14 TD subjects. There are 300 images in the dataset.

Separate csv files for ASD and TD viewings of the 300 images. For each 300 images, there is a csv file. Content of a csv file:

Idx, x, y, duration
0,323,343,416
1,303,190,58
2,456,203,8
3,991,249,550
4,323,274,283
...

Idx can range from 0-33, and a change of index from any number to 0 refers to a change in subject who is viewing that particular image. Thus we can extract fixation points and duration by each subject per images.

Mode of number of fixations by a subject = 8

Mean ‘’ ‘’ = 7.8

Week 2 Progress
19/6/22 - 25/6/22

1. Searching through internet to find how to encode duration into LSTMs.
2. Implementing Time-Dependent Representation for Neural Event Sequence Prediction [22] in PyTorch
3. Setting up validation metrics and logging for validation and experimentation
4. Experimentation

Time-Dependent Representation for Neural Event Sequence Prediction [22]:

There are two ways to encode duration into a RNN:

Time Mask:
Using a non-linear transformation, duration is encoded into a context vector which is formed as a mask after applying sigmoid function. The mask is then element-wise multiplied with the event token embedding to be fed into the RNN/LSTM.

Event-Time Joint embedding:
Duration project to a larger feature space, on which softmax is performed to give a soft one-hot encoding. This encoding is then projected into same space as event token after which it is added to it.

The above methods would be referred to as ‘mask’ and ‘joint’ respectively hereafter.

These were successfully implemented in our pytorch code in colab notebook.

Validation metrics and logging during training

We are using the binary cross entropy loss to train the network since it outputs a probability between 0 and 1 on whether or not the given fixation sequences comes from a person with ASD or not.

To log the metrics during training, *Weight and Biases* (wandb.ai) is being used.

A training set and validation set was split from the original dataset. Validation set is 10% of the original set. We are logging the bce loss on the training set every 25 iterations.

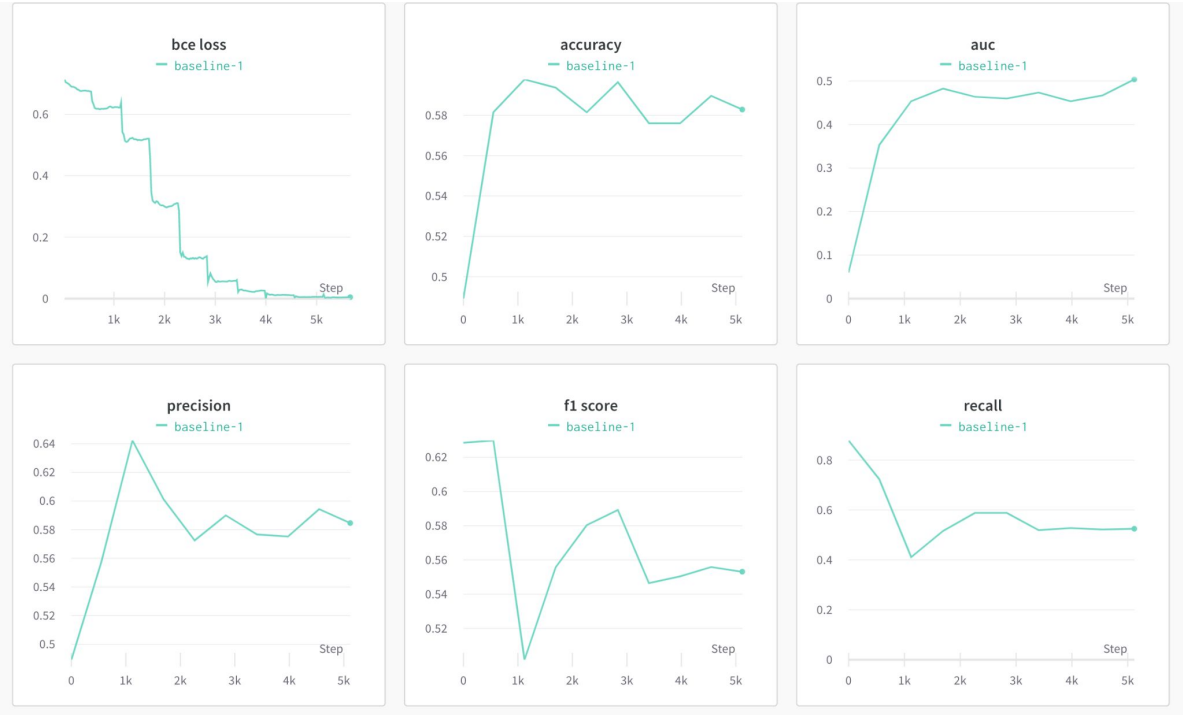
After each epoch validation is performed on the validation set and the following metrics are computed using the torchmetrics library:

1. Accuracy
2. AUC
3. Precision
4. Recall
5. F1 score

Experiments

1. Baseline results: [acc: 0.5829 | auc: 0.5038 | f1: 0.5532 | pre: 0.5846 | rec: 0.5249]

```
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
hidden_size = 512
clip = 10
img_height = 600
img_width = 800
```



Experiments

2. Only time-mask results: [acc: 0.6064 | auc: 0.5628 | f1: 0.6038 | pre: 0.6087 | rec: 0.5989]

```
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
hidden_size = 512
mask = True
joint = False
time_proj_dim = 64
clip = 10
img_height = 600
img_width = 800
```

Experiments

3. Only time-joint results: [acc: 0.5762 | auc: 0.4815 | f1: 0.5859| pre: 0.58 | rec: 0.5918]

```
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
hidden_size = 512
mask = False
joint = True
time_proj_dim = 64
clip = 10
img_height = 600
img_width = 800
```

Experiments

4. mask + joint results: [acc: 0.624 | auc: 0.3748 | f1: 0.6166 | pre: 0.6376 | rec: 0.5969]

```
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
hidden_size = 512
mask = True
joint = True
time_proj_dim = 64
clip = 10
img_height = 600
img_width = 800
```

Experiments

Summary and some thoughts:

	Accuracy	AUC	F1 score	Precision	Recall
Baseline	0.5829	0.5038	0.5532	0.5846	0.5249
mask	0.6064	0.5628	0.6038	0.6087	0.5989
joint	0.5762	0.4815	0.5859	0.58	0.5918
mask + joint	0.624	0.3748	0.6166	0.6376	0.5969

Mask + joint gives the best results. These results are comparable to those submitted in the Saliency4ASD challenge but still are not better than those achieved by SP-ASDNet. There is transfer learning occurring in SP-ASDNet since it uses SalGAN which has already been trained on 10000 images from the SALICON dataset.

Our results are purely from training from scratch (Resnet50 had initial weights from the VGG classification task) on just 270 images which demonstrates the **robustness** of our methodology and potential to give **state of the art results if pretrained on a larger dataset**.

Week 3 Progress
26/6/22 - 2/7/22

1. Finding and downloading all possible Saliency-related datasets for pretraining
2. Preprocessing datasets not containing explicit fixation durations.
3. Trying to pose a self-supervised learning task so that our network can learn useful representations

Saliency Datasets:

The original Saliency4ASD dataset has explicit duration for each fixation coordinate (x,y). There are 3 kinds of saliency datasets:

1. Only provide fixation coordinates (x,y),
2. Provides (x,y,dur) for each fixation => The most useful for our task
3. Provide raw gaze points recorded at a particular frequency. Fixations will have to be extracted using an algorithm. [We used velocity threshold (I-VT) algorithm [23] for fixation extraction]

List of (x,y,dur) datasets:

1. OSIE [24]
2. Toronto [25]
3. DOVES [26]
4. EMoD [27]
5. FiWI [28]

List of (x,y) datasets:

1. SALICON [4]
2. CAT2000 [6]
3. VIU [29]

Raw gaze point datasets:

MIT1003 [30]
=> can be turned into (x,y,dur) using I-VT

Data Preprocessing

1. MIT1003:
   - 1003 images
   - Original data was in matlab files, EDA was done using MATLAB to understand the structure of the original dataset, and then loaded into python for processing using SciPy io library function loadmat.
   - Created an ivt_mod() function which taken in raw gaze points captured at a particular frequency and computes fixation coordinates as well as duration of each fixation based on the velocity-thresholding algorithm. Another heuristic was applied such that any sequential fixations closer than 20 px were merged together and considered as one fixation.
   - Annotation dictionary created which maps for each image name as key to a dictionary containing fixations and durations list as observed by different viewers. The dictionary was then saved as a JSON file.

Data Preprocessing

2. CAT2000:
   - 2000 images
   - Original data was in matlab files in the form of a MATLAB Map container, which couldn't be accessed using loadmat function of SciPy. Hence the map container was encoded into JSON using MATLAB's jsonencode function.
   - The JSON file was loaded and for each image name, fixations were extracted and cleaned using the 20 px heuristic. Annotation dictionary created and then saved as json file.
   - (x,y) data

3. SALICON:
   - 10000 images
   - A single sequence of fixation already available per image. Extracted and annotation dictionary created with the same structure as previous datasets and saved as json file.
   - (x,y) data

4. VIU:
   - 700 images
   - Per image, we have multiple fixations by 17-22 people
   - Annotation dictionary made and saved as json file. (x,y) data

Posing a sufficiently useful Self-Supervised learning task

- Since we have 2 kinds of datasets i.e with and without duration as a feature, we should first train our network without using duration first. And then continue training while using duration datasets. Thus our network first gets a grasp of how to represent fixation sequences usefully for any downstream task. We have 12800 images for no-duration data.

- Some ideas for self-supervised learning task consists of:

  1. Masking last part of the sequence, and letting the model predict the last part using the first half of sequences.

  2. Divide sequence into two. Mask some fixations of both the halves. Also have distractor sequences generated using noise or from another image. Give the first half and another half either from same sequence or from another/noise and ask the model to predict whether the two given sequences are of the same one or not.

- We might need to explore transformers aswell compared to LSTMs. A thorough study of the field of SSL is required.

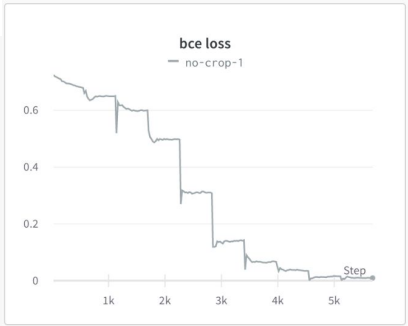
Week 4 Progress
3/7/22 - 8/7/22

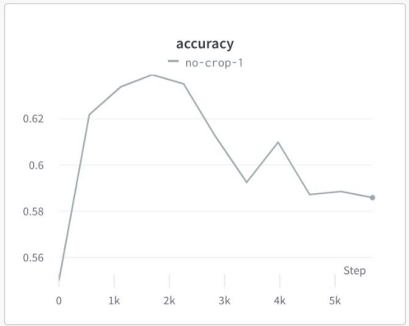
1. Experimenting with LSTM
2. PImplementing Transformer and experimentation
3. Self supervised task

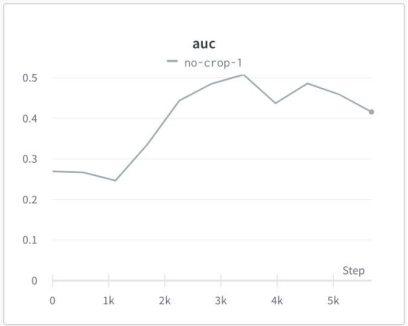
Experiments on LSTM

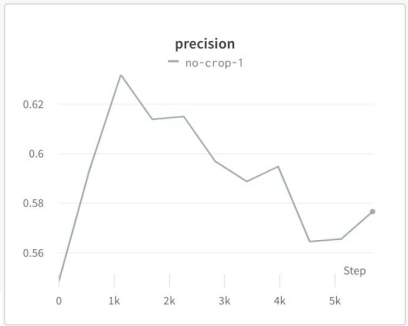
1. Last hidden state w/o crop: [acc: 0.5859 | auc: 0.4157 | f1: 0.571 | pre: 0.5766 | rec: 0.5656]

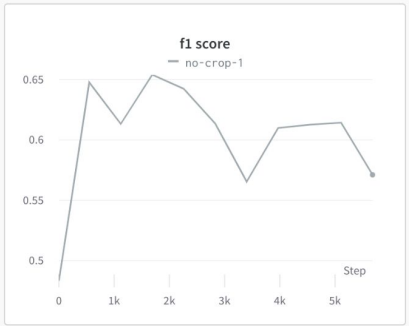
```
#Sal_seq
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
clip = 10
img_height = 600
img_width = 800
hidden_size = 512
mask = False
joint = False
crop_seq = False
all_lstm = False
time_proj_dim = 256
```







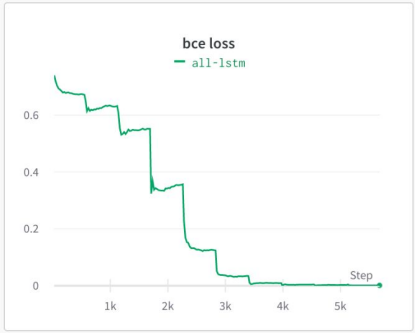


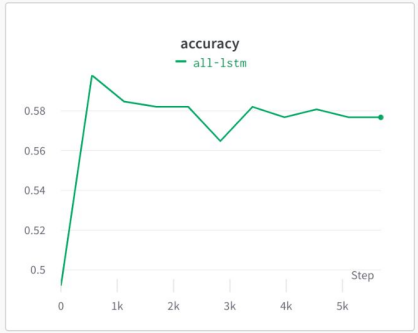


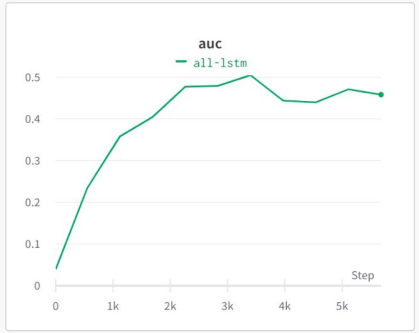
Experiments on LSTM

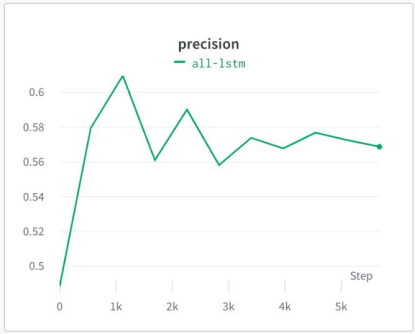
2. All hidden states: [acc: 0.5767 | auc: 0.4582 | f1: 0.6 | pre: 0.5687 | rec: 0.6349]

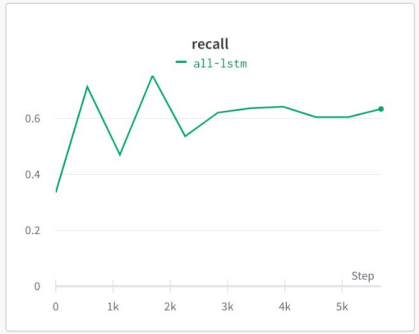
```
#Sal_seq
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
clip = 10
img_height = 600
img_width = 800
hidden_size = 512
mask = False
joint = False
crop_seq = False
all_lstm = True
time_proj_dim = 256
```











Experiments on Transformer

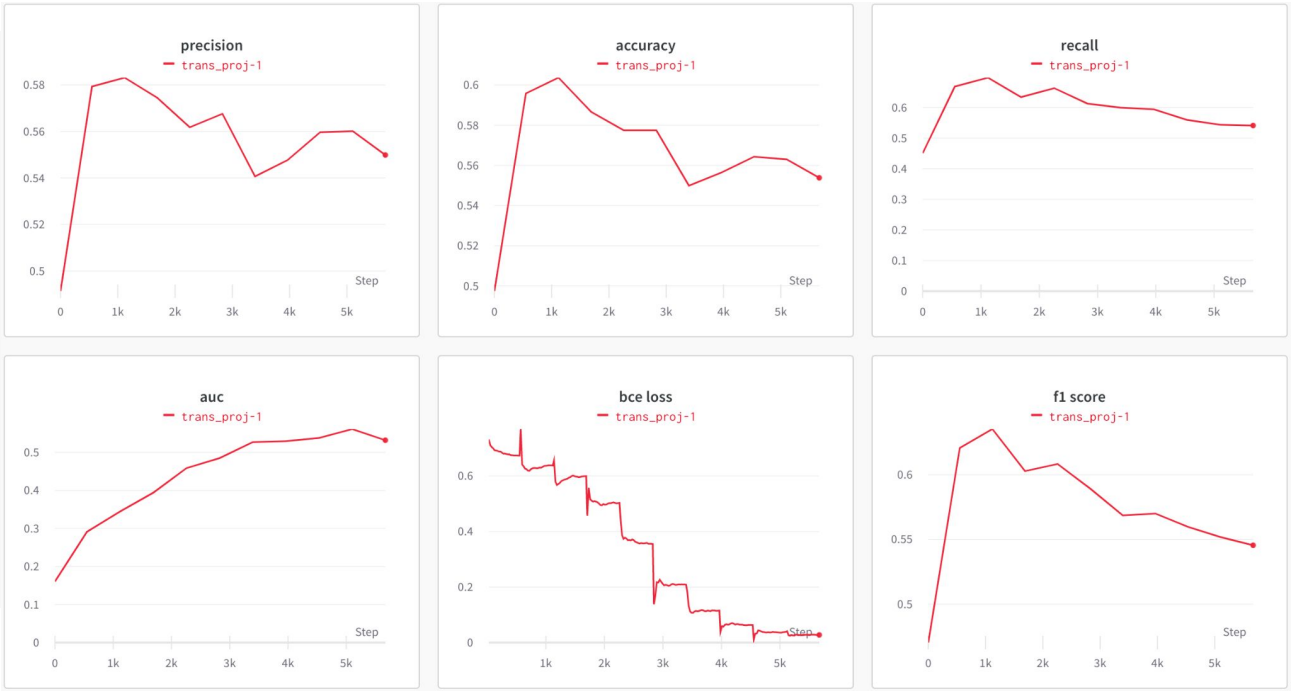
1. 2048 embedding dimension: [acc: 0.5773 | auc: 0.4775 | f1: 0.604 | pre: 0.5701 | rec: 0.6421]

```
#Sal_transformer
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
clip = 10
img_height = 600
img_width = 800
num_blocks=1
heads=4
device='gpu'
expansion=4
drop_prob=0.4
emb_dim=2048
```

Experiments on Transformer

2. 512 embedding dimension: [acc: 0.5538 | auc: 0.532 | f1: 0.5455 | pre: 0.5499 | rec: 0.5411]

```
#Sal_transformer
LR = 1e-4
img_dir = 'Dataset/TrainingData/Images'
anno_dir = 'Dataset/TrainingData'
backend = 'resnet'
checkpoint_path= 'checkpoints'
num_epochs = 10
val_ratio = 0.1
batch_size = 12
max_len = 14
clip = 10
img_height = 600
img_width = 800
num_blocks=1
heads=4
device='gpu'
expansion=4
drop_prob=0.4
emb_dim=512
```



Experiments

	Accuracy	AUC	F1 score	Precision	Recall
baseline	0.5829	0.5038	0.5532	0.5846	0.5249
mask	0.6064	**0.5628**	0.6038	0.6087	0.5989
joint	0.5762	0.4815	0.5859	0.58	0.5918
mask + joint	**0.624**	0.3748	**0.6166**	**0.6376**	0.5969
no crop	0.5859	0.4157	0.571	0.5766	0.5656
all states	0.5767	0.4582	0.6	0.5687	0.6349
trans-2048	0.5773	0.4775	0.604	0.5701	**0.6421**
trans-512	0.5538	0.532	0.5455	0.5499	0.5411

The mask+joint+crop architecture seems to give the best results yet. It is still too early to rule out transformers. We should see which of these architectures work well on the pretraining task.

Self-Supervised Learning Task Idea & Plans

- During train time, mask out any of the fixations in the sequence, then make the model predict the missing fixation's coordinates (and duration).

- Will have to preprocess all Saliency datasets and form a single large dataset to train on.

- Train on the dataset without predicting duration. Then train on the portion of the dataset for which duration is present to also predict duration.

- Experiments will have to be done to determine which architecture can best handle the pre-training task so as to have a more successfully transfer learning to the ASD dataset.

References:

1. de Belen, R.J., Bednarz, T.P., Sowmya, A., & Del Favero, D. (2020). Computer vision in autism spectrum disorder research: a systematic review of published studies from 2009 to 2019. Translational Psychiatry, 10.

2. H. Duan, G. Zhai, X. Min, Z. Che, Y. Fang, X. Yang, J. Gutiérrez, P. Le Callet, “A Dataset of Eye Movements for the Children with Autism Spectrum Disorder”, ACM Multimedia Systems Conference (MMSys’19), Jun. 2019.

3. O. Le Meur, A. Nebout, M. Chérel & E. Etchamendy, From Kanner autism to Asperger syndromes, the difficult task to predict where ASD people look at, IEEE Access, 2020. DOI 10.1109/ACCESS.2020.3020251

4. Jiang, Ming, Shengsheng Huang, Juanyong Duan and Qi Zhao. “SALICON: Saliency in Context.” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 1072-1080.

5. Judd, Tilke, Frédo Durand and Antonio Torralba. “A Benchmark of Computational Models of Saliency to Predict Human Fixations.” (2012).

6. Ali Borji, Laurent Itti. CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research [CVPR 2015 workshop on "Future of Datasets"]

7. Chen, Shi and Qi Zhao. “Attention-Based Autism Spectrum Disorder Screening With Privileged Modality.” 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019): 1181-1190.

8. Duan, Huiyu, Xiongkuo Min, Yi Fang, Lei Fan, Xiaokang Yang and Guangtao Zhai. “Visual Attention Analysis and Prediction on Human Faces for Children with Autism Spectrum Disorder.” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 15 (2020): 1 - 23.

9. Banire, Bilikis, Dena Al-Thani, Marwa Khalid Qaraqe and Bilal Mansoor. “Face-Based Attention Recognition Model for Children with Autism Spectrum Disorder.” Journal of Healthcare Informatics Research 5 (2021): 420 - 445.

10. Hosseini, Mohammad-Parsa, Madison Beary, Alex Hadsell, Ryan Messersmith and Hamid Soltanian-Zadeh. “Deep Learning for Autism Diagnosis and Facial Analysis in Children.” Frontiers in Computational Neuroscience 15 (2021): n. Pag.

References:

11. Arumugam, Sajeev Ram, Rebecca Balakrishna, Rashmita Khilar, Oswalt Manoj and C. Shylaja. “Prediction of Autism Spectrum Disorder in Children using Face Recognition.” 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (2021): 1246-1250.

12. Gutiérrez, Jesús, Zhaohui Che, Guangtao Zhai and Patrick Le Callet. “Saliency4ASD: Challenge, dataset and tools for visual attention modeling for autism spectrum disorder.” Signal Process. Image Commun. 92 (2021): 116092.

13. Wei, Weijie, Zhi Liu, Lijin Huang, Alexis Nebout and Olivier Le Meur. “Saliency Prediction via Multi-Level Features and Deep Supervision for Children with Autism Spectrum Disorder.” 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (2019): 621-624.

14. Nebout, Alexis, Weijie Wei, Zhi Liu, Lijin Huang and Olivier Le Meur. “Predicting Saliency Maps for ASD People.” 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (2019): 629-632.

15. Fang, Yuming, Haiyan Zhang, Y. Zuo, Wenhui Jiang, Hanqin Huang and Jiebin Yan. “Visual attention prediction for Autism Spectrum Disorder with hierarchical semantic fusion.” Signal Process. Image Commun. 93 (2021): 116186.

16. M. Startsev, M. Dorr, Classifying autism spectrum disorder based on scanpaths and saliency, in: IEEE International Conference on Multimedia & Expo Workshops, ICMEW, Shanghai, China, 2019, pp. 633–636

17. G. Arru, P. Mazumdar, F. Battisti, Exploiting visual behaviour for autism spectrum disorder identification, in: IEEE International Conference on Multimedia & Expo Workshops, ICMEW, Shanghai, China, 2019

18. Y. Tao, M.-L. Shyu, SP-ASDNet: CNN-LSTM Based ASD classification model using observer ScanPaths, in: IEEE International Conference on Multimedia & Expo Workshops, ICMEW, Shanghai, China, 2019, pp. 641–646

19. C. Wu, S. Liaqat, S.-C.S. Cheung, C.-N. Chuah, S. Ozonoff, Predicting autism diagnosis using image with fixations and synthetic saccade patterns, in: IEEE International Conference on Multimedia & Expo Workshops, ICMEW, Shanghai, China, 2019, pp. 647–650

References:

20. S. Xu, J. Yan, M. Hu, A new bio-inspired metric based on eye movement data for classifying ASD and typically developing children, Signal Process. (2021)

21. Pan, Junting, Cristian Canton-Ferrer, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol and Xavier Giro-i-Nieto. “SalGAN: Visual Saliency Prediction with Generative Adversarial Networks.” ArXiv abs/1701.01081 (2017): n. Pag.

22. Li, Yang, Nan Du and Samy Bengio. “Time-Dependent Representation for Neural Event Sequence Prediction.” ArXiv abs/1708.00065 (2018): n. Pag.

23. Salvucci, Dario D. and Joseph H. Goldberg. “Identifying fixations and saccades in eye-tracking protocols.” ETRA (2000).

24. Juan Xu, Ming Jiang, Shuo Wang, Mohan Kankanhalli, Qi Zhao. Predicting Human Gaze Beyond Pixels [JoV 2014]

25. Neil Bruce, John K. Tsotsos. Attention based on information maximization [JoV 2007]

26. Ian van der Linde, Umesh Rajashekar, Alan C. Bovik, Lawrence K. Cormack. DOVES: A database of visual eye movements [Spatial Vision 2009]

27. Shaojing Fan, Zhiqi Shen, Ming Jiang, Bryan L. Koenig, Juan Xu, Mohan Kankanhali, Qi Zhao. Emotional Attention: A Study of Image Sentiment and Visual Attention [CVPR 2018] (Spotlight)

28. Chengyao Shen, Qi Zhao. Webpage Saliency [ECCV 2014]

29. Kathryn Koehler, Fei Guo, Sheng Zhang, Miguel P. Eckstein. What Do Saliency Models Predict? [JoV 2014]

30. Tilke Judd, Krista Ehinger, Fredo Durand, Antonio Torralba. Learning to Predict where Humans Look [ICCV 2009]