INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD
VI Semester B.Tech in Information Technology
Report - C3

Autism detection in children via Image Processing

Under the guidance of:

Dr Vrijendra Singh

Submitted By :-

Archisha Baranwal (IIT2018018)

Palak Mittal (IIT2018117)

Nehal Singh (IIT2018119)

Puja Kumari(IIT2018191)

Prabha Kumari (IIT2018195)

Tables Of Content

1. Abstract

Autism Spectrum Disorder (ASD) is a neuro-disorder in which a child or a person has a lifelong effect on interaction and communication with others. It is diagnosed at any stage in life It is called “behavioral disease” . The ASD problem starts with childhood and continues to keep going on into adolescence and adulthood . We have used Convolutional Neural Network (CNN) for predicting and analysis of ASD problems in a child, adolescents, and adults. Dataset is consist of Autistic and Non-Autistic face images of children. We are using two models of CNN (Convolutional Neural Network) methods for prediction which are VGG16 and VGG19. It is giving accuracy of almost 84.333 % and 84.0% respectively .

2. Introduction & Motivation

A. Introduction

Autism spectrum disorder (ASD) is a neurodevelopmental disorder with onset in early childhood that is associated with a wide range of symptoms and ability levels. Although no cure exists for **autism** spectrum disorder, and there is no one-size-fits-all treatment. The only treatment is to maximize your child's ability to function by reducing **autism** spectrum disorder symptoms and supporting development and learning. In this project we are going to detect autism disease by analysing images to make such a model which will help two determine ASD and help people gain confidence in their recovery rate.

B. Motivation

- While surfing on the web, one day one of us came across an advertisement of a child suffering from ASD which took attention to look into the term.
- On detailed research we analysed The Centers for Disease Control's Autism and Developmental Disabilities Monitoring (ADDM) Network reports that in 2014, approximately 1 in 59 children in the United States (1 in 37 boys, and 1 in 151 girls), has been identified with an autism spectrum disorder (ASD).
- The biggest challenge we see coming is to detect the ASD with high accuracy for our testing dataset so as to ensure to get the best output from our project.
- Thus, we decided to work on the project and propose NGOs to lead it further, maybe our little contribution can create a little effect.

3. Problem Definition & Objectives

A. Problem definition

Topic -Image-based Autism detection using eye movements and fixation time
We have a dataset of facial images of autistic and non-autistic children.
We have to predict autism from facial image.

B. Objectives

Our aim is to develop a binary classifier for discriminating the two classes of facial image . We have to train the classifier that can predict the class of participants into autistic (1) and non-autistic (0) using CNN classifiers of machine learning using various models of CNN and comparing their results.

4. Literature Review and Dataset Description

A. Literature Review

S No.	Paper Title	Name of the Conference/journal (Year)	Purpose	Methodology	Dataset	Results	Future Scope
1.	Visualization of Eye-Tracking Patterns in Autism Spectrum Disorder: Method and Dataset [1]	Thirteenth International Conference on Digital Information Management (ICDIM) Date of Conference: 24-26 Sept. 2018 Conference Location: Berlin, Germany, Germany	Aim of this paper is to visualize the eye-tracking patterns of ASD-diagnosed individuals with particular focus on children at early stages of development.	This paper use the methodology of transforming the dynamics of eye motion into a visual representation, and hence diagnosis-related tasks could be approached using image-based techniques.	It uses 59 participants(children) for an eye tracking experiment. Then the psychologist labelled participants into two categories as: i) ASD-Diagnosed, or ii) Non-ASD.	The promising prediction accuracy achieved by a simple logistic regression model evidently demonstrated the validity and applicability of our methodology .	In future, the dataset can allow for developing further useful applications or discovering interesting insights using Machine Learning or data mining techniques.

2.	Learning Clusters in Autism Spectrum Disorder: Image-Based Clustering of Eye-Tracking Scanpaths with Deep Autoencoder [2]	41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Date of conference - 23-27 July 2019 Conference Location: Berlin, Germany, Germany	Aim of this conference was to apply unsupervised machine learning to discover clusters in Autism Spectrum Disorder (ASD).	The clustering model was trained using compressed representations learned by a deep autoencoder. The key idea is to learn clusters based on the visual representation of eye-tracking scanpaths.	It uses the dataset of images of retina for clustering. Eye-tracking scanpaths could be grouped into coherent clusters	Its results provided a set of implications to be considered. First, the clustering experiments empirically confirmed that eye-tracking scanpaths could be grouped into coherent clusters, which largely resembled the original grouping of samples (i.e. ASD or non-ASD).	The dataset can allow for developing further useful applications or discovering interesting insights using image based clustering of eye-tracking.
3.	Predict Autism Spectrum Disorder Using Machine Learning and Eye-Tracking Scanpaths [3]	12th International Conference on Health Informatics Date of Conference- February 2019 Conference Location - Prague, Czech republic	This paper describes the visualization of eye tracking with the focusing on Autism Spectrum Disorder.	It uses the techniques of image based diagnosis	This dataset includes visualizations of eye-tracking scanpaths with a particular focus Autism Spectrum Disorder m(ASD).	Results evidently demonstrated that such visual representations could simplify the prediction problem, and attained a high accuracy as well	We can use simple neural network models and a relatively limited dataset).

4.	Early detection for autism spectrum disorder in young children [4]	Article : Paediatrics & Child Health, Volume 24, Canada, Issue 7, November 2019, Pages 424–432 Published Date - 24 Oct 2019	This paper statement provides clear, comprehensive, evidence-informed re-commendations and tools to help community paediatricians and other primary care providers .	This paper describes the result on canadian children. It provides the general information of ASD.	The dataset has the figure of canadian child and the suggestion for ASD detection.it describes the sign of ASD , it will be very helpful to diagnose it.	This paper gave an overview of data with the help of canadian children and described the information of research in a beautiful manner.	In future, this dataset can be used for data analysing of future model It can be used as a dataset in the coming days.
5.	Evaluating the EEG and Eye Movements for Autism Spectrum Disorder . [5]	2018 IEEE International Conference on Big Data (Big Data) Date of Conference: 10-13 Dec. 2018 Conference location:Seattle, WA, USA	This paper presents an analysis and comparison between EEG, Eye and combined data.	They have compared four models,two models were created for each model with only EEG and combined data by using PCA and without using PCA, Like SVM with PCA and without PCA. .	The dataset has the figure of canadian child and the suggestion for ASD detection.it describes the sign of ASD , it will be very helpful to diagnose it.	This paper gave an overview of data with the help of canadian children and described the information of research in a beautiful manner.	In future, this dataset can be used for data analysing of future model It can be used as a dataset in the coming days.
6.	Automatic Detection of Autism Spectrum Disorder by Tracking the Disorder Co-morbidities [6]	2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON) Date of Conference: 13-15 March	This main aim of this paper provides insights into the mechanism underlying risk perception in ASD.	In this paper we use the SVM classifier as it's more robust and accurate in comparison to other classifiers. The SSD was calculated	The data were collected , using Biopac MP150 system, from three electrodes placed on the midline(Fz,Pz,and Cz) electrodes placed according to	The overall accuracy has a value of 99% for automatic detection of ASD.	ERP is a very good technique for neuropsychiatric research and holds great promise for the future work.

		2019 Conference Location: Jaipur, India		by computing the difference between the median risky and medium neutral response.	10-20 international systems.		
7.	A Review of Early Detection of Autism Based on Eye-Tracking and Sensing Technology [7]	2020 International Conference on Inventive Computation Technologies (ICICT) Date of Conference: 26-28 Feb. 2020 Conference Location: Coimbatore, India	The main aim of this paper is to detect and monitor ASD in the early stage of life using eye-tracking and sensing technologies.	The eye tracking technique allows psychologists to diagnose ASD by monitoring eye movement for a short time and analyze the eye fixation of ASD.	The data was collected from 65 participants, 34 with ASD children and 31 TD children. The average age was 8 years old. They collected the data by Tobi X2 eye tracking.	The result showed the children in the training group significantly increased the percentage of engagements to faces relative to objects after training. And also showed the ASD children have less fixation on the face.	In the future, Studies collected the eye fixation and eye-tracking maps while watching the visual dynamic stimuli or static stimuli.
8.	Predicting Autism Diagnosis using Image with Fixations and Synthetic Saccade Patterns [8]	2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) Date of Conference: 8-12 July 2019 Conference Location: Shanghai, China	This paper describes the predictions of ASD diagnosis using gaze data.	In this paper, we propose two machine learning methods, synthetic saccade approach and image based approach, to automatically classify ASD given the scanpath data from	The data provided contains 300 images. There are 6050 sample from TD and ASD subjects. We split them into training and validation dataset (80% vs 20%).	In this paper an impressive results of 92% accuracy on 20 high-functio-ning ASD and 19 typically developed adults, which may not be directly applicable to gaze patterns from children.	In the future One of the goals of the challenge is to propose ML models to classify ASD and typically developed (TD) viewers using gaze data.

				children on free viewing of natural images			

B. Dataset Description-

Dataset (Name/ Description)	Characteristics of dataset	Publicly available (if yes then mention URL)
Learning to predict autism spectrum disorder based on the facial image	This dataset includes visualizations of facial images with a particular focus Autism Spectrum Disorder (ASD).	Detect Autism from a facial image

We will be using this (dataset) as our dataset. This dataset includes visualizations of facial image with a particular focus on Autism Spectrum Disorder (ASD). This dataset is also publicly available and has been used to give successful experimental results in context to ASD. In the dataset 0 is representation for Autisitic and 1 is representation for Non_Autisitic. The data is provided in the consolidated directory. This directory has the two sub directories of Autistic and Non_Autistic. It represents the consolidation of the files from the train, test and valid directories into a single set. The training set is labeled as a train. All categories consist of two sub directories, Autistic and Non_Autistic.

The dataset descriptions is as follows:-

- Train: Train dataset images used to prepare our model, to train it.
- Test: Test dataset images used to test our model and then we measure performance of our model.
- Valid: Valid dataset used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters.

Activities of children during which images were clicked:

- Repeat words,phrases from television or movies
- Repetitive activities with objectives like pencils ,toys figures
- Repetitive body , arm,hand of fingers movement.
- During sharing of interests , achievements or emotions.
- During eye contact to communicate.

- Unusual reactions with smells , sounds ,taste.
- During playing with peers.
- There are also many more activities during which those dataset images are clicked.

Dataset described above is in the public domain, and can be easily downloaded. We have done some modifications for sorting the dataset, and the final one we have uploaded and worked from https://drive.google.com/drive/folders/1ubuJjj2Xubajey_d2vmp-hcpR2tWk926

5. Background

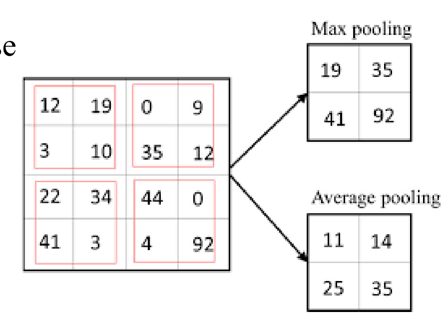
a. *CNN (Convolutional Neural Network)*

Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner.
Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

The components of a CNN are as detailed below:

1. Convolution Layer : In convolution layer we take a small window size (typically of length 3*3) that extends to the depth of the input matrix. During every iteration we slid the window by stride size,, and computed the dot product of filter entries and input values at a given position. That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color.

2. Pooling Layer : We use a pooling layer to decrease the size of the activation matrix and ultimately reduce the learnable parameters.
There are two type of pooling :
a) Max Pooling
b) Average Pooling



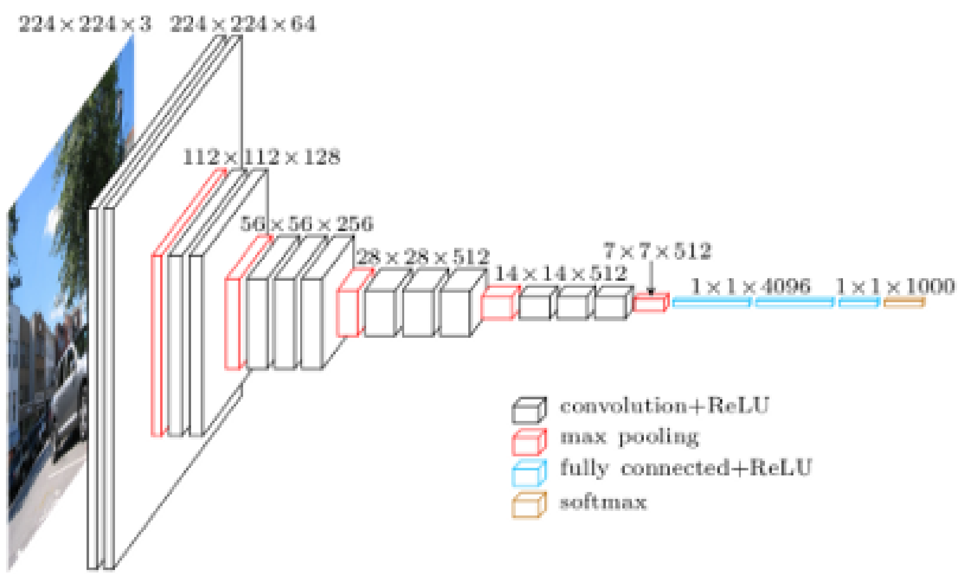
3. Fully Connected Layer :
In convolution layers neurons are connected only to a local region, while in a fully connected region, we connect all the inputs to neurons.

4. Final Output Layer :
After getting values from the fully connected layer, we connect them to the final layer of neurons (having count equal to total number of classes), that will predict the probability of each image to be in different classes.

Model of CNN which we are using in this project are :

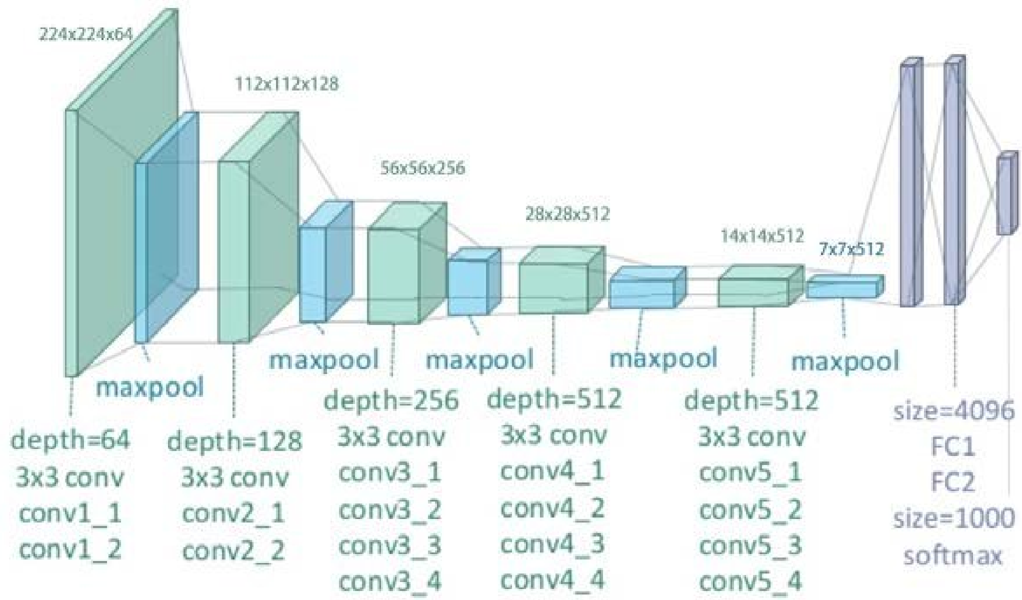
a. VGG16:

VGG16 is a convolution neural net (CNN) architecture which we would be using. It is considered to be one of the excellent vision model architectures till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having Convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride2 It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully Connected layers) followed by a softmax for
output. The 16 in VGG16 refers to it having 16 layers that have weights.
This network is a pretty large network and it has about 138 million (approx) parameters.



b. VGG19:

VGG19 is a variant of the VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layers, 5 MaxPool layers and 1 SoftMax layer).
The VGG19 network uses only five types of layers. All convolution layers use 3 × 3 kernels and all the max pool layers use 2 × 2 kernels. 19 counts only the convolution and fully connected layers. Due to the presence of fully connected layers, VGG-19 is not fully convolutional and therefore, will only be able to accept 224 × 224 × 3 inputs without retraining.



6. Methodology

- Load Train dataset: First, we will start by training our dataset. Our dataset contains the file with file names as “Non_Autistic.135.jpg , Autistic.391.jpg”, where the word is splitted by ‘.’, initial word describes the kind (whether the person is autistic or not and next word describes the image no.).
- Load Test dataset: Our dataset contains the file with file names as “Non_Autistic.135.jpg , Autistic.391.jpg”
- We will be using Multi layer neural networks as deep neural networks.
- Since the dataset consists of images ,Convolutional neural networks will be used.
- Preparing Model: Then, we will prepare the model following the steps given below:

We will define hyperparameters

Then Flatten the output layer to 1 dimension

Further we will add a fully connected layer with 512 hidden units and ReLU activation

Next a dropout rate of 0.5 will be added

Lastly a final sigmoid layer for classification will be added

- Preprocessing of Data: We will be preprocessing the data and extracting data from images in the form of matrix.
- Preparing the training dataset:

Training Generator: Here we will be preparing and generating the train data.

Validation Generator: We will be creating a validation generator to filter out the data to ensure the quality of performance by using the validate data.

validation_data: Data on which to evaluate the loss and any model metrics at the end of each epoch. The model will not be trained on this data.

- Model Fitting: Now we will train the data for a fixed number of epochs and batch size.
- Preparing the testing dataset: We will create a testing generator for generating the test dataset using the previously loaded test images.
- Prediction: Now, we will be predicting the result for the test dataset and using the model evaluated earlier.

  Output will be as follows :

  Image name as :- actual_name(image name{0 or 1}) , eg. autistic.127.jpg(1) , here 0/1 is prediction.
- Our predictions will next go to submission_13010030.csv.
- This file contains the predicted result, and further we will calculate its accuracy , precision etc.

7. Software and Hardware Requirements:

A. Software Requirements

- Google Collab

Languages

- Python 3
- Matlab

Tools:

- Keras
- Tensorflow
- PIL

Libraries :

- Numpy
- Pandas
- Matplotlib
- SKlearn
- Optimizer
- Layers
- Seaborn
- Os
- VGG16
- VGG19
- Model

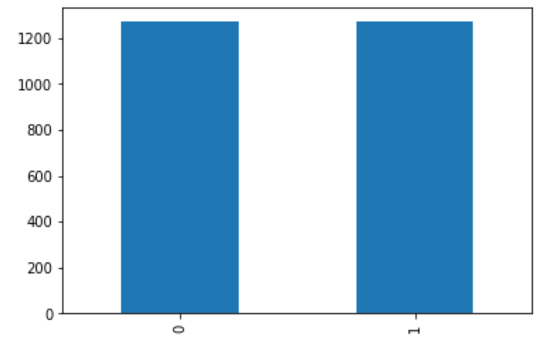
B. Hardware Requirements

- Laptop
- 8 GB RAM, GPU
- System with fast processor and a graphics card.

8. Implementation

- Train dataset: We have classified the dataset as category (0,1) as (Non Austistic , Austistic) respectively.

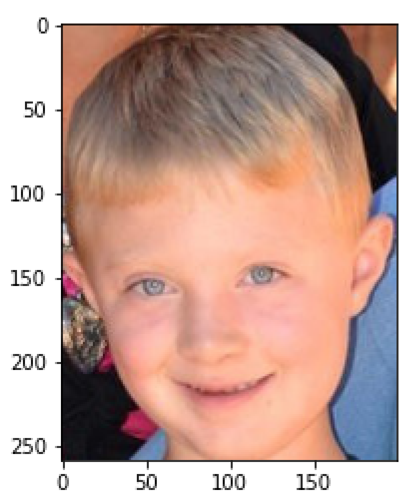
	filename	category
0	Non_Autistic.135.jpg	0
1	Non_Autistic.1211.jpg	0
2	Non_Autistic.261.jpg	0
3	Autistic.999.jpg	1
4	Non_Autistic.1227.jpg	0
2539	Autistic.303.jpg	1



- Test dataset: Here we will load the test dataset which will consist of only filename.

	Filename
0	Autistic.135.jpg
1	Autistic.27.jpg
2	Autistic.14.jpg
3	Autistic.88.jpg
4	Autistic.17.jpg

- Sample Image: Now, we will load the sample image as given below:



- Model was prepared with hyperparameters as epochs = 20 and batch_size = 12.

  Model was compiled with :

```
model.compile(loss='binary_crossentropy',
optimizer=optimizers.SGD(lr=1e-3, momentum=0.9),
metrics=['accuracy'])
```

Now Model is prepared with output as:

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0         
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792      
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928     
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0         
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856     
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584    
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0         
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168    
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080    
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080    
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0         
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160   
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808   
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808   
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0         
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808   
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808   
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808   
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0         
_________________________________________________________________
global_max_pooling2d (Global (None, 512)               0         
_________________________________________________________________
dense (Dense)                (None, 512)               262656    
_________________________________________________________________
dropout (Dropout)            (None, 512)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513       
=================================================================
Total params: 14,977,857
Trainable params: 14,977,857
Non-trainable params: 0
_________________________________________________________________
```

- Next data was trained and preprocessed with using parameters as :

  rotation_range=15,
  rescale=1./255,
  shear_range=0.2,
  zoom_range=0.2,
  horizontal_flip=True,
  fill_mode='nearest',
  width_shift_range=0.1,

height_shift_range=0.1

- Now data will validated and stored in “validation_datagen”
- Model fitting was carried out , and for each epoch accuracy and loss was calculated.

```
Epoch 1/20
  2/190 [..............................] - ETA: 9s - loss: 0.6298 - accuracy: 0.7083WARNING:tensorflow:Callbacks method `on_train_batch
190/190 [==============================] - 40s 209ms/step - loss: 0.7220 - accuracy: 0.5259 - val_loss: 0.6818 - val_accuracy: 0.5198
Epoch 2/20
190/190 [==============================] - 39s 205ms/step - loss: 0.6717 - accuracy: 0.5792 - val_loss: 0.5425 - val_accuracy: 0.7302
Epoch 3/20
190/190 [==============================] - 39s 205ms/step - loss: 0.5789 - accuracy: 0.6970 - val_loss: 0.4283 - val_accuracy: 0.8095
Epoch 4/20
190/190 [==============================] - 39s 204ms/step - loss: 0.5247 - accuracy: 0.7423 - val_loss: 0.4196 - val_accuracy: 0.8016
Epoch 5/20
190/190 [==============================] - 39s 204ms/step - loss: 0.4916 - accuracy: 0.7744 - val_loss: 0.3452 - val_accuracy: 0.8532
Epoch 6/20
190/190 [==============================] - 39s 205ms/step - loss: 0.4441 - accuracy: 0.7880 - val_loss: 0.3388 - val_accuracy: 0.8532
Epoch 7/20
190/190 [==============================] - 39s 206ms/step - loss: 0.4373 - accuracy: 0.8043 - val_loss: 0.3159 - val_accuracy: 0.8770
Epoch 8/20
190/190 [==============================] - 39s 203ms/step - loss: 0.4386 - accuracy: 0.7982 - val_loss: 0.3591 - val_accuracy: 0.8492
Epoch 9/20
190/190 [==============================] - 39s 204ms/step - loss: 0.4074 - accuracy: 0.8162 - val_loss: 0.3490 - val_accuracy: 0.8690
Epoch 10/20
190/190 [==============================] - 39s 203ms/step - loss: 0.3970 - accuracy: 0.8245 - val_loss: 0.3810 - val_accuracy: 0.8333
Epoch 11/20
190/190 [==============================] - 39s 203ms/step - loss: 0.3818 - accuracy: 0.8377 - val_loss: 0.3046 - val_accuracy: 0.8690
Epoch 12/20
190/190 [==============================] - 38s 202ms/step - loss: 0.3441 - accuracy: 0.8522 - val_loss: 0.3209 - val_accuracy: 0.8690
Epoch 13/20
190/190 [==============================] - 38s 203ms/step - loss: 0.3457 - accuracy: 0.8474 - val_loss: 0.3000 - val_accuracy: 0.8929
Epoch 14/20
190/190 [==============================] - 38s 202ms/step - loss: 0.3483 - accuracy: 0.8514 - val_loss: 0.3646 - val_accuracy: 0.8373
Epoch 15/20
190/190 [==============================] - 39s 205ms/step - loss: 0.3124 - accuracy: 0.8729 - val_loss: 0.2924 - val_accuracy: 0.8730
Epoch 16/20
190/190 [==============================] - 39s 204ms/step - loss: 0.3263 - accuracy: 0.8580 - val_loss: 0.4080 - val_accuracy: 0.8214
Epoch 17/20
190/190 [==============================] - 38s 203ms/step - loss: 0.2898 - accuracy: 0.8804 - val_loss: 0.3349 - val_accuracy: 0.8651
Epoch 18/20
190/190 [==============================] - 38s 202ms/step - loss: 0.2952 - accuracy: 0.8738 - val_loss: 0.3018 - val_accuracy: 0.8929
Epoch 19/20
190/190 [==============================] - 38s 202ms/step - loss: 0.2850 - accuracy: 0.8817 - val_loss: 0.3119 - val_accuracy: 0.8651
Epoch 20/20
190/190 [==============================] - 38s 201ms/step - loss: 0.2446 - accuracy: 0.8975 - val_loss: 0.3313 - val_accuracy: 0.8651
```

- Loss and accuracy for model will be : accuracy = 0.865079 ; loss = 0.331409
- Next prediction was made with a threshold as 0.5 .
  (Refer image5.jpeg).
  This image contains 9 images with actual and predicted results as filename.
- Our predictions will next go to submission_13010030.csv, which will be as follows:

id	**label**
Autistic	1
Autistic	1
Autistic	1
Autistic	1
Autistic	1
Autistic	1

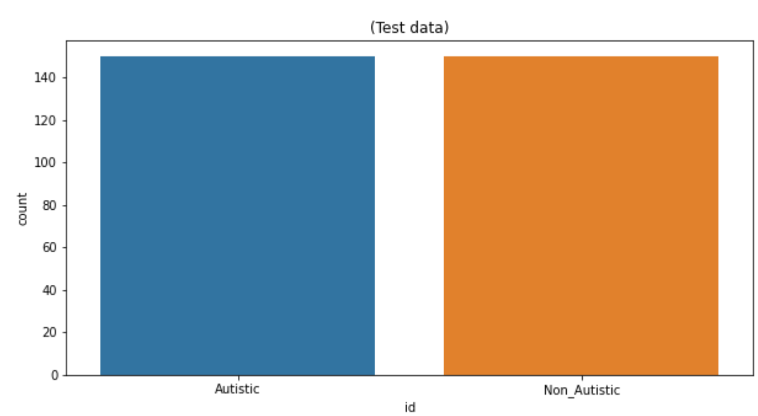
For **VGG19,**

model was as follows :

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv4 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv4 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv4 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
global_max_pooling2d (Global (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 512)               262656
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 20,287,553
Trainable params: 20,287,553
Non-trainable params: 0
_________________________________________________________________
```

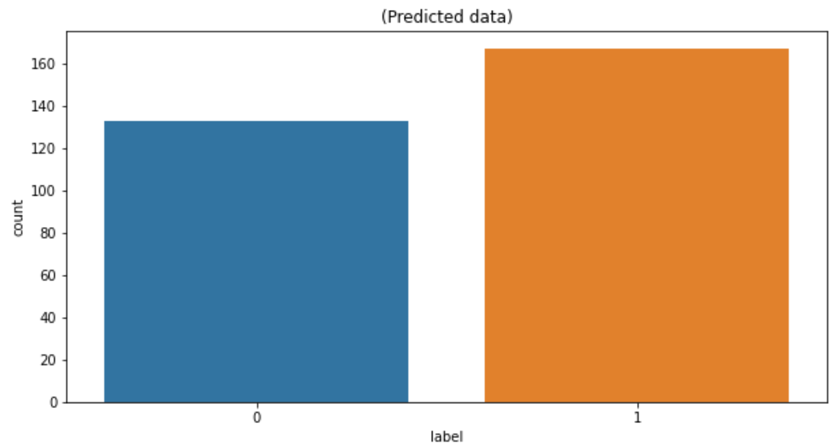
9. Results

- Graph for Actual Test Data:



For VGG16:

- Graph for Predicted Data :



- Number of images which were classified as :
    - Predicted Autistic : 149
    - Predicted Non Autistic : 151
    - Actual Autistic : 150
    - Actual Non Autistic : 150

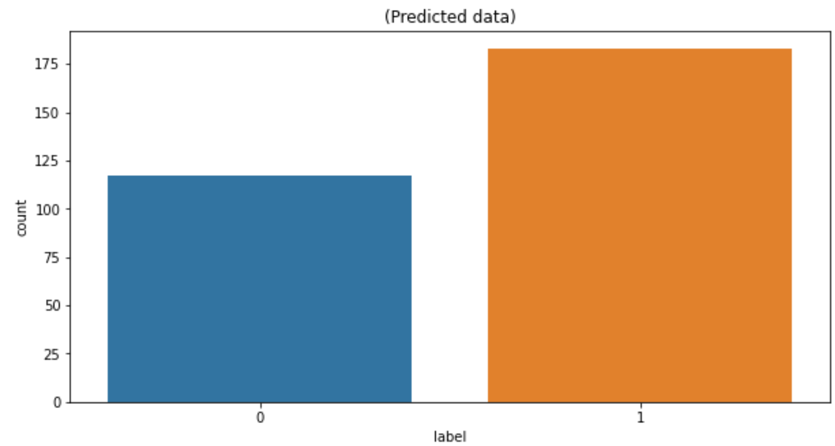
- Percentage of images classified as :
    - `Actual Non Autistic percentage in total test data:  50.0 %`
    - `Predicted Non Autistic percentage in total test data: 50.33333333333333 %`
    - `Actual Autistic percentage in total test data:  50.0 %`
    - `Predicted Autistic percentage in total test data: 49.666666666666664 %`

- Confusion matrix was used for predicting accuracy & other results:
        - `True positive :  126`
        - `True Negative :  127`
        - `false Positive :  24`
        - `false Negative :  23`

    - Accuracy is: 84.33333333333334 %
    - Precision is: 84.0 %
    - Sensitivity is: 84.56375838926175 %
    - Specificity is: 84.10596026490066 %

For VGG19:

- Graph for Predicted Data :



- The number of images which were classified as :
    - predicted Autistic : 158
    - predicted Non Autistic : 142
    - Actual Autistic : 150
    - Actual Non Autistic : 150

- Percentage of images classified as :
    - Actual Non Autistic percentage in total test data: 50.0 %
    - Predicted Non Autistic percentage in total test data: 47.333333333333336 %
    - Actual Autistic percentage in total test data: 50.0 %
    - Predicted Autistic percentage in total test data: 52.666666666666664 %

- Confusion matrix was used for predicting accuracy & other results:
        - True positive : 130
        - True Negative : 122
        - false Positive : 20
        - false Negative : 28

    - Accuracy is: Accuracy is: 84.0 %
    - Precision is: 86.66666666666667 %
    - Sensitivity is: 82.27848101265823 %
    - Specificity is: 85.91549295774648 %

10. Result (Taking real dataset)

Here we took the images for the prediction and result calculation from a real dataset that is from the Internet and our own family members. THese images are in the “test2” folder in the Dataset folder.

Dataset :

```
                  filename
0      real-autistic.2.jpg
1  real_non_autistic.3.jpg
2      real_autistic.1.jpg
3  real_non_autistic.4.jpg
```

Results :

	id	label
0	real-autistic	1
1	real_non_autistic	0
2	real_autistic	1
3	real_non_autistic	0

Dataset Link :
https://drive.google.com/drive/folders/1uoPtK_HyqIpLv150eFupSUFimTZVxoh8?usp=sharing

11. Comparison

Base paper has a dataset of 59 participants labelled as ASD-Diagnosed and Non-ASD.
In this base paper , a key idea in the methodology is to construct a visual representation of eye motion. They have transformed the eye-tracking records into visual patterns.It appears that the ASD- diagnosed participant had a tendency of looking at the bottom of the screen ,where the eye-tracking device was placed . The data transformation process was fully implemented using python. The visualizations were produced by the Matplotlib library .

We have a dataset of 1470 autistic facial images and 1470 non-autistic facial images . Out of those we are using 1269 images comprising both autistic and non-autistic for training purposes. We have used 149 images for testing the model. We have used VGG16 and VGG19 models of CNN (Convolutional Neural Network) methodology to train the model and test the model .Accuracy of this model is approx 84.33% and 84% respectively and precision is about 84.6% and 86.3% .

12. Contribution

We have used base code from ref no. 9 for training and fitting the model. Later we improved the accuracy from 77% to 84.3%, precision, specificity and sensitivity. By making significant changes like epoch number, batch size and various other factors, we have also improved training and testing as compared to base code.

Also, we have calculated various factors to judge the results in a better way such as calculating the true positive, true negative, false positive and false negative followed by the representation of the confusion matrix, calculating the accuracy, precision, sensitivity and specificity of the prediction result in the submission.csv.

We have also implemented the VGG19 model of CNN and compared the results of both the models.

13. Answer to C2 questions

- Number of parameters:
  - Input layer: Zero as input layer has nothing to learn . It just provides the input image's shape so there are no learnable parameters here
  - CONV layer : Number ((m * n * d)+1)* k), added 1 because of the bias term for each filter.
    - m: height of filter
    - n: width of filter
    - d: no of filter in prev layer
    - k: no of filter in present layer
    - So, for example , for conv layer 1 in block 1:
      - m=3 n=3 d=3 k=64
      - Hence, no of params = ((3*3*3)+1)*64 = 1792
  - Max pooling layer : Zero because there are no parameters we could learn in the pooling layer. This layer is just used to reduce the image dimension size.
  - Fully connected layer : ((c * p)+1*c) , here 1*c is added as a bias term.
    - c : current layer neurons
    - p : previous layer neurons

14. Conclusion

In this project, we proposed a CNN (Convolutional Neural Network) architecture to identify and classify ASD patients. Our proposed CNN architecture is able to obtain higher classification performance with fewer parameters, which will reduce the training time. Therefore, our proposed model is less complex and faster as compared to other similar models.

The result shows that the accuracy of our model using test data by the VGG16 model is 84.33% and the VGG19 model is 84% meaning that it outperformed the best accuracy obtained on this dataset so far and **we get better accuracy in the VGG16 model.** But **precision in the VGG16 model is 84% whereas in the VGG19 model is 86.6%**, here **VGG19 outperforms the VGG16 results in terms of precision.** Our model **Sensitivity and specificity is approximately 84.56 and 84.10% for the VGG16 model and, 82.27 % and 85.915% for the VGG19 model**, so we can say that our project determines actual positive cases and negative cases (in case of Autistic and Non_Autistic predicate true) very efficiently.

15. References

[1.] Romuald Carette, Mahmoud Elbattah, Gilles Dequen, Jean-Luc Guérin, Federica Cilia, “Visualization of Eye-Tracking Patterns in Autism Spectrum Disorder: Method and Dataset”, 2018 Thirteenth International Conference on Digital Information Management (ICDIM)
Visualization of Eye-Tracking Patterns in Autism Spectrum Disorder: Method and Dataset

[2]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," 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).
Learning Clusters in Autism Spectrum Disorder: Image-Based Clustering of Eye-Tracking Scanpaths with Deep Autoencoder

[3.] Romuald Carette, Mahmoud Elbattah, Federica Cilia, Gilles Dequen, ”Predicting Autism Spectrum Disorder Using Machine Learning and Eye-Tracking”, 12th International Conference on Health Informatics At: Prague, Czech Republic
Learning to Predict Autism Spectrum Disorder based on the Visual Patterns of Eye-tracking Scanpaths

[4] Lonnie Zwaigenbaum, Jessica A Brian, Angie Ip, Early detection for autism spectrum disorder in young children, Paediatrics & Child Health, Volume 24, Issue 7, November 2019, Pages 424–432,
Early detection for autism spectrum disorder in young children

[5.] Sashi Thapaliya, Sampath Jayarathna, Mark Jaime, ”Evaluating the EEG and Eye Movements for Autism Spectrum Disorder” , 2018 IEEE International Conference on Big Data (Big Data)
Evaluating the EEG and Eye Movements for Autism Spectrum Disorder - IEEE Conference Publication

[6.] Tanu and D. Kakkar, "Automatic Detection of Autism Spectrum Disorder by Tracing the Disorder Co-morbidities," 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)
Automatic Detection of Autism Spectrum Disorder by Tracing the Disorder Co-morbidities

[7.] Zeyad Abdulhameed Taha Ahmed, Mukti E. Jadhav, “A Review of Early Detection of Autism Based on Eye-Tracking and Sensing Technology” , 2020 International Conference on Inventive Computation Technologies (ICICT)
A Review of Early Detection of Autism Based on Eye-Tracking and Sensing Technology

[8.] Chongruo Wu, Sidrah Liaqat, Sen-ching Cheung, Chen-Nee Chuah, Sally Ozonoff, “Predicting Autism Diagnosis using Image with Fixations and Synthetic Saccade Patterns”, 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)
Predicting Autism Diagnosis using Image with Fixations and Synthetic Saccade Patterns

[9]. Cihan Senol, Student at Gaziosmanpasa University, Kielce, Świętokrzyskie, Poland
https://www.kaggle.com/cihan063/autism-cnn-vgg16