Detection of Autism spectrum disorder using computer vision

B N Shrikriti, Sayantan Nandy, Hanchate Samyuktha
Vellore Institute of Technology, Chennai

Abstract

Observation plays a key role in identifying the mental disorders, especially in infants and children of the age zero to eight years. Autism Spectrum Disorder (ASD) tends to develop majorly during this period of time and parents and clinicians are expected to be keenly observant of the movements of the child. Kaggle dataset was taken and transfer learning was applied on the same. We also used VGGNet to train the model and get better accuracy.

Keywords: Autism Spectrum Disorder, VGGNet, Transfer Learning, clinicians

1 Introduction

Autism is a complex and lifelong developmental disability that mostly starts at the early ages, during childhood which impacts the ability to communicate and interact. It impacts the nervous system and affects the overall cognitive, emotional, social and physical health of the affected individual. The range and severity of symptoms can vary widely. Common symptoms include difficulty with communication, difficulty with social interactions, obsessive interests and repetitive behaviors. ASD is a set of defects, with varying degrees, in socialization, communication, expressing and understanding emotions and stereotype.

Autistic children (2 to 6 years old) might perform quite unnatural behaviors such as getting nervous by small changes, repeating words, or phrases all over again and again, avoiding eye interaction and having a tendency to be alone. Not every child has the same symptoms, they are unique and these unique symptoms can tell the severity level from high to low. Even the intelligence level varies in children with ASD. Some may be slow learners and can have low intelligence level and some may be extraordinarily intelligent and have high intelligence level but they can still be having trouble communicating and because of this unique mixture of symptoms, severity can sometimes be difficult to determine. It's generally based on the level of impairments and how they impact the ability to function.

Many researches have been performed by numerous researchers in the area of autism. Esraa T.Sadek,Noha A. Seada,Said Ghoniemy (Sadek et al., 2020) in their paper proposed medical assisting computer vision-based framework to detect observable autism symptoms. They conducted surveys and included papers of various researchers and what they worked on. Qandeel Tariq, Jena Daniels, Jessey Nicole Schwartz, Peter Washington, Haik Kalantarian1,2, Dennis Paul Wall analyzed 3-minute home videos of children with and without ASD to get a speed and an accurate autism classification. Wang et al. in their research found a robust link between specific genes and being susceptible to be diagnosed by ASD.

Transfer learning is a popular approach in deep learning. Transfer learning is said to be, will be, after supervised learning, the next driver of ML commercial success. Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. Xception is also known as "extreme" version of an Inception module.
Layer freezing means layer weights of a trained model are not changed when they are reused in a subsequent downstream task - they remain frozen. Essentially when back propagation is done during training these layers weights are untouched.

In order to detect the autism, we have applied ResNET and Xception model has been prepared for fine tuning.The dataset which we are considering has facial data, pictures of autistic and non autistic children of the age group 2-8 years.

2 Review of Literature

David Deriso, Joshua Susskind, Lauren Krieger and Marian Bartlett (Deriso et al., 2012) presented a novel intervention system to identify, improve the facial expression perception and production in children with autism spectrum disorders (ASD). They have used Emotion Mirror, a game where the facial expressions of the children with the disorders were mirrored by a cartoon character on screen that responds in real time dynamically.

According to research by F. Nazeen, F. A Boujarwah, S. Sadler, A Mogus, G.D. Abowed and R. I Arriga(Nazneen et al., 2010) hardware must not be a constraint to the environment where the test is conducted: The clinician should be free to adjust the testing conditions if necessary and children shouldn't necessarily wear any type of sensors.

Jordan Hashemi, Thiago Vallin Spina, Mariano Tepper (Hashemi et al., 2012) applied Histograms of Oriented Gradients (HOG) as descriptors to represent each of the facial features and then classify them using a Support Vector Machine. The results were taken from actual clinical recordings in which the at-risk infant/ toddler is tested by an experienced clinician following Autism Observation Scale for Infants (AOSI)

Qandeel Tariq, Jena Daniels, Jessey Nicole Schwartz, Peter Washington,Haik Kalantarian1,2, Dennis Paul Wall (Tariq et al., 2018) in their paper analyzed 3-minute home videos of children with and without ASD to get a speed and an accurate autism classification.They in their paper supported the possibility that mobile video analysis with the help of ML models may enable autism detection rapidly in order to reduce the waiting periods and get necessary treatments as early as possible.

Esraa T.Sadek,Noha A. Seada,Said Ghoniemy (Sadek et al., 2020) in their paper proposed medical assisting computer vision-based framework to detect observable autism symptoms.They conducted surveys and included papers of various researchers and what they worked on. This mostly consists of what are autism signs detection and recognition, approaches for detecting autism such as using genetics and blood analysis,using eeg,using MRI,using wearable sensors,using vision techniques.They provided information of various datasets of human activity recognition for autism detection.They also proposed model which goes through four main stages, which are dataset pre-processing, features extractions, behavior description, and behavior classification.

Halim Abbas, Ford Gaberson, Stuart Liu-Mayo,eric Glover and Dennis P. Wall (Abbas et al., 2018) in their paper presented a multimodular, machine learning-based assessment of autism comprising three complementary modules for a unified outcome of diagnostic-grade reliability: A 4-minute, parent report questionnaire delivered via a mobile app, a list of key behaviors identified from 2-minute, semi structured home videos of children, and a 2-minute questionnaire presented to the clinician at the time of clinical assessment.

3 Proposed Methodology

Initially the required libraries have been imported. The weights of the base model that has been applied in ResNet taken from Imagenet. With a rotation range of 90, the Image Data Generator has been used. Two dense layers with relu activation function has been used, followed by a flatten layer and dropout layer of 0.5. Fully connected layers are of size 1024 each. The model is compiled with learning rate 0.00001 and losse function categorical cross entropy. The number of epochs were 50 and batch size is given to be 8. The graphs have been hence plotted.
An xception model for fine tuning where the base model is of the shape (299,299,3) was done. The base layers were then freezed for no further changes. By freezing it means that the layer will not be trained. Hence the weights will not be changed.There is no enough time to train the deep neural networks.
The base layer is freezed followed by a flatten layer, this is followed by a dense layer. The activation function here is relu and this is followed by another dense layer followed by a prediction layer which is the output layer with softmax activation function.
Then a xception model is prepared to carry out finetuning. When freeze baselayers is True, the base model acts as a feature extractor that is used for classification by the latter layers. Model

summary has hence been derived.
Categorical crossentropy with varying epochs and optimizers has been tried out. The model was then compiled and Relevent graphs have been hence drawn.

3.1 Experimentation

In Xception model, we carried out experiments by changing the number of epochs and optimizers. The following are the summaries of each model.

EPOCHS=1, OPTIMIZER=Adam

The epoch value is taken as 1 and the optimizer is taken as Adam. The learning rate here taken is 0.001.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 299, 299, 3)]	0	[]
block1_conv1 (Conv2D)	(None, 149, 149, 32)	864	['input_2[0][0]']
block1_conv1_bn (BatchNormalization)	(None, 149, 149, 32)	128	['block1_conv1[0][0]']
block1_conv1_act (Activation)	(None, 149, 149, 32)	0	['block1_conv1_bn[0][0]']
block1_conv2 (Conv2D)	(None, 147, 147, 64)	18432	['block1_conv1_act[0][0]']
block1_conv2_bn (BatchNormalization)	(None, 147, 147, 64)	256	['block1_conv2[0][0]']
block1_conv2_act (Activation)	(None, 147, 147, 64)	0	['block1_conv2_bn[0][0]']
block2_sepconv1 (SeparableConv2D)	(None, 147, 147, 128)	8768	['block1_conv2_act[0][0]']
block2_sepconv1_bn (BatchNormalization)	(None, 147, 147, 128)	512	['block2_sepconv1[0][0]']
block2_sepconv2_act (Activation)	(None, 147, 147, 128)	0	['block2_sepconv1_bn[0][0]']
block2_sepconv2 (SeparableConv2D)	(None, 147, 147, 128)	17536	['block2_sepconv2_act[0][0]']
block2_sepconv2_bn (BatchNormalization)	(None, 147, 147, 128)	512	['block2_sepconv2[0][0]']
conv2d_4 (Conv2D)	(None, 74, 74, 128)	8192	['block1_conv2_act[0][0]']
block2_pool (MaxPooling2D)	(None, 74, 74, 128)	0	['block2_sepconv2_bn[0][0]']
batch_normalization_4 (BatchNormalization)	(None, 74, 74, 128)	512	['conv2d_4[0][0]']
add_12 (Add)	(None, 74, 74, 128)	0	['block2_pool[0][0]', 'batch_normalization_4[0][0]']
block3_sepconv1_act (Activation)	(None, 74, 74, 128)	0	['add_12[0][0]']
block3_sepconv1 (SeparableConv2D)	(None, 74, 74, 256)	33920	['block3_sepconv1_act[0][0]']
block14_sepconv2_bn (BatchNormalization)	(None, 10, 10, 2048)	8192	['block14_sepconv2[0][0]']
block14_sepconv2_act (Activation)	(None, 10, 10, 2048)	0	['block14_sepconv2_bn[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['block14_sepconv2_act[0][0]']
predictions (Dense)	(None, 1000)	2049000	['avg_pool[0][0]']
flatten (Flatten)	(None, 1000)	0	['predictions[0][0]']
dense (Dense)	(None, 64)	64064	['flatten[0][0]']
dense_1 (Dense)	(None, 2)	130	['dense[0][0]']
softmax (Softmax)	(None, 2)	0	['dense_1[0][0]']

Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480

EPOCHS = 2,OPTIMIZER = Adam
The epoch value is taken as 2 and the optimizer is taken as Adam. The learning rate here taken is 0.001

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 299, 299, 3)]	0	[]
block1_conv1 (Conv2D)	(None, 149, 149, 32)	864	['input_3[0][0]']
block1_conv1_bn (BatchNormalization)	(None, 149, 149, 32)	128	['block1_conv1[0][0]']
block1_conv1_act (Activation)	(None, 149, 149, 32)	0	['block1_conv1_bn[0][0]']
block1_conv2 (Conv2D)	(None, 147, 147, 64)	18432	['block1_conv1_act[0][0]']
block1_conv2_bn (BatchNormalization)	(None, 147, 147, 64)	256	['block1_conv2[0][0]']
block1_conv2_act (Activation)	(None, 147, 147, 64)	0	['block1_conv2_bn[0][0]']
block2_sepconv1 (SeparableConv2D)	(None, 147, 147, 128)	8768	['block1_conv2_act[0][0]']
block14_sepconv2_bn (BatchNormalization)	(None, 10, 10, 2048)	8192	['block14_sepconv2[0][0]']
block14_sepconv2_act (Activation)	(None, 10, 10, 2048)	0	['block14_sepconv2_bn[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['block14_sepconv2_act[0][0]']
predictions (Dense)	(None, 1000)	2049000	['avg_pool[0][0]']
flatten_1 (Flatten)	(None, 1000)	0	['predictions[0][0]']
dense_2 (Dense)	(None, 64)	64064	['flatten_1[0][0]']
dense_3 (Dense)	(None, 2)	130	['dense_2[0][0]']
softmax_1 (Softmax)	(None, 2)	0	['dense_3[0][0]']

Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480

EPOCHS = 3,OPTIMIZER = Adam
The epoch value is taken as 3 and the optimizer is taken as Adam. The learning rate here taken is 0.001

Model: "model_2"

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	[(None, 299, 299, 3)]	0	[]
block1_conv1 (Conv2D)	(None, 149, 149, 32)	864	['input_4[0][0]']
block1_conv1_bn (BatchNormalization)	(None, 149, 149, 32)	128	['block1_conv1[0][0]']
block1_conv1_act (Activation)	(None, 149, 149, 32)	0	['block1_conv1_bn[0][0]']
block1_conv2 (Conv2D)	(None, 147, 147, 64)	18432	['block1_conv1_act[0][0]']
block1_conv2_bn (BatchNormalization)	(None, 147, 147, 64)	256	['block1_conv2[0][0]']
block1_conv2_act (Activation)	(None, 147, 147, 64)	0	['block1_conv2_bn[0][0]']
block2_sepconv1 (SeparableConv2D)	(None, 147, 147, 128)	8768	['block1_conv2_act[0][0]']
block14_sepconv2 (SeparableConv2D)	(None, 10, 10, 2048)	3159552	['block14_sepconv1_act[0][0]']
block14_sepconv2_bn (BatchNormalization)	(None, 10, 10, 2048)	8192	['block14_sepconv2[0][0]']
block14_sepconv2_act (Activation)	(None, 10, 10, 2048)	0	['block14_sepconv2_bn[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['block14_sepconv2_act[0][0]']
predictions (Dense)	(None, 1000)	2049000	['avg_pool[0][0]']
flatten_2 (Flatten)	(None, 1000)	0	['predictions[0][0]']
dense_4 (Dense)	(None, 64)	64064	['flatten_2[0][0]']
dense_5 (Dense)	(None, 2)	130	['dense_4[0][0]']
softmax_2 (Softmax)	(None, 2)	0	['dense_5[0][0]']

Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480

EPOCHS = 3,OPTIMIZER = SGD
The epoch value is taken as 3 and the optimizer is taken as SGD. The learning rate here taken is 0.001

```
Model: "model_3"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_5 (InputLayer)            [(None, 299, 299, 3  0           []
                                )]

block1_conv1 (Conv2D)           (None, 149, 149, 32  864         ['input_5[0][0]']
                                )

block1_conv1_bn (BatchNormaliz  (None, 149, 149, 32  128         ['block1_conv1[0][0]']
ation)                          )

block1_conv1_act (Activation)   (None, 149, 149, 32  0           ['block1_conv1_bn[0][0]']
                                )

block1_conv2 (Conv2D)           (None, 147, 147, 64  18432       ['block1_conv1_act[0][0]']
                                )

block1_conv2_bn (BatchNormaliz  (None, 147, 147, 64  256         ['block1_conv2[0][0]']
ation)                          )

block1_conv2_act (Activation)   (None, 147, 147, 64  0           ['block1_conv2_bn[0][0]']
                                )

block2_sepconv1 (SeparableConv  (None, 147, 147, 12  8768        ['block1_conv2_act[0][0]']
2D)                             8)

                                )

block14_sepconv2 (SeparableCon  (None, 10, 10, 2048  3159552     ['block14_sepconv1_act[0][0]']
v2D)                            )

block14_sepconv2_bn (BatchNorm  (None, 10, 10, 2048  8192        ['block14_sepconv2[0][0]']
alization)                      )

block14_sepconv2_act (Activati  (None, 10, 10, 2048  0           ['block14_sepconv2_bn[0][0]']
on)                             )

avg_pool (GlobalAveragePooling  (None, 2048)         0           ['block14_sepconv2_act[0][0]']
2D)

predictions (Dense)             (None, 1000)         2049000     ['avg_pool[0][0]']

flatten_3 (Flatten)             (None, 1000)         0           ['predictions[0][0]']

dense_6 (Dense)                 (None, 64)           64064       ['flatten_3[0][0]']

dense_7 (Dense)                 (None, 2)            130         ['dense_6[0][0]']

softmax_3 (Softmax)             (None, 2)            0           ['dense_7[0][0]']

==================================================================================================
Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480
```

EPOCHS = 3,OPTIMIZER = RMSprop
The epoch value is taken as 3 and the optimizer is taken as RMSProp. The learning rate here taken is 0.001

```
Model: "model_4"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_6 (InputLayer)            [(None, 299, 299, 3  0           []
                                )]

block1_conv1 (Conv2D)           (None, 149, 149, 32  864         ['input_6[0][0]']
                                )

block1_conv1_bn (BatchNormaliz  (None, 149, 149, 32  128         ['block1_conv1[0][0]']
ation)                          )

block1_conv1_act (Activation)   (None, 149, 149, 32  0           ['block1_conv1_bn[0][0]']
                                )

block1_conv2 (Conv2D)           (None, 147, 147, 64  18432       ['block1_conv1_act[0][0]']
                                )

block1_conv2_bn (BatchNormaliz  (None, 147, 147, 64  256         ['block1_conv2[0][0]']
ation)                          )

block1_conv2_act (Activation)   (None, 147, 147, 64  0           ['block1_conv2_bn[0][0]']
                                )

block2_sepconv1 (SeparableConv  (None, 147, 147, 12  8768        ['block1_conv2_act[0][0]']
2D)                             8)

block14_sepconv2 (SeparableCon  (None, 10, 10, 2048  3159552     ['block14_sepconv1_act[0][0]']
v2D)                            )

block14_sepconv2_bn (BatchNorm  (None, 10, 10, 2048  8192        ['block14_sepconv2[0][0]']
alization)                      )

block14_sepconv2_act (Activati  (None, 10, 10, 2048  0           ['block14_sepconv2_bn[0][0]']
on)                             )

avg_pool (GlobalAveragePooling  (None, 2048)         0           ['block14_sepconv2_act[0][0]']
2D)

predictions (Dense)             (None, 1000)         2049000     ['avg_pool[0][0]']

flatten_4 (Flatten)             (None, 1000)         0           ['predictions[0][0]']

dense_8 (Dense)                 (None, 64)           64064       ['flatten_4[0][0]']

dense_9 (Dense)                 (None, 2)            130         ['dense_8[0][0]']

softmax_4 (Softmax)             (None, 2)            0           ['dense_9[0][0]']

==================================================================================================
Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480
__________________________________________________________________________________________________
```

4 Results

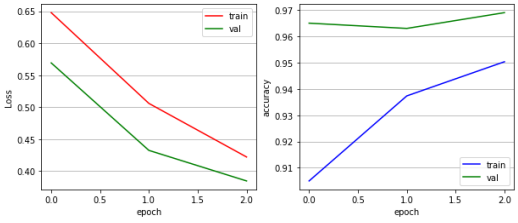
With ResNet model we got training accuracy of 0.5802 and with xception model the highest accuracy we got through various hyperparameter tuning was 0.62701.

```
dropout_6 (Dropout)             (None, 1024)         0           ['flatten_5[0][0]']

dense_16 (Dense)                (None, 1024)         1049600     ['dropout_6[0][0]']

dense_17 (Dense)                (None, 1024)         1049600     ['dense_16[0][0]']

flatten_6 (Flatten)             (None, 1024)         0           ['dense_17[0][0]']

dropout_7 (Dropout)             (None, 1024)         0           ['flatten_6[0][0]']

dense_18 (Dense)                (None, 2)            2050        ['dropout_7[0][0]']

==================================================================================================
Total params: 236,454,786
Trainable params: 212,867,074
Non-trainable params: 23,587,712
__________________________________________________________________________________________________
```

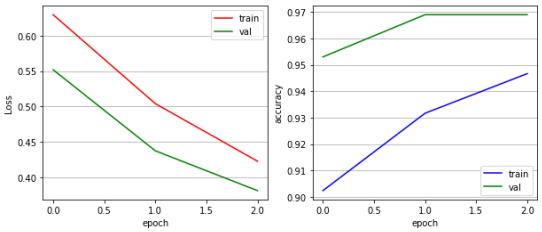
```
block14_sepconv2_act (Activati  (None, 10, 10, 2048  0           ['block14_sepconv2_bn[0][0]']
on)                             )

avg_pool (GlobalAveragePooling  (None, 2048)         0           ['block14_sepconv2_act[0][0]']
2D)

predictions (Dense)             (None, 1000)         2049000     ['avg_pool[0][0]']

flatten (Flatten)               (None, 1000)         0           ['predictions[0][0]']

dense (Dense)                   (None, 64)           64064       ['flatten[0][0]']

dense_1 (Dense)                 (None, 2)            130         ['dense[0][0]']

softmax (Softmax)               (None, 2)            0           ['dense_1[0][0]']

==================================================================================================
Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480
__________________________________________________________________________________________________
```

```
avg_pool (GlobalAveragePooling  (None, 2048)         0           ['block14_sepconv2_act[0][0]']
2D)

predictions (Dense)             (None, 1000)         2049000     ['avg_pool[0][0]']

flatten_2 (Flatten)             (None, 1000)         0           ['predictions[0][0]']

dense_4 (Dense)                 (None, 64)           64064       ['flatten_2[0][0]']

dense_5 (Dense)                 (None, 2)            130         ['dense_4[0][0]']

softmax_2 (Softmax)             (None, 2)            0           ['dense_5[0][0]']

==================================================================================================
Total params: 22,974,674
Trainable params: 64,194
Non-trainable params: 22,910,480
__________________________________________________________________________________________________
```

Graphs have been provided for reference
Optimizer=Adam, Epoch=3



Optimizer=RMSprop, Epoch=3



5 Conclusion

So as we conclude the models which we have tried are discussed above and it has to be noted that xception model gave a good accuracy and with these we were able to get a good amount of results. But still this leaves a huge gap and opportunity to

work on.

References

Halim Abbas, Ford Garberson, Eric Glover, and Dennis P Wall. 2018. Machine learning approach for early detection of autism by combining questionnaire and home video screening. *Journal of the American Medical Informatics Association*, 25(8):1000–1007.

David Deriso, Joshua Susskind, Lauren Krieger, and Marian Bartlett. 2012. Emotion mirror: A novel intervention for autism based on real-time expression recognition. In *Computer Vision – ECCV 2012*, volume 7585 of *Lecture Notes in Computer Science*, pages 671–674. Springer.

Jordan Hashemi, Thiago Vallin Spina, Mariano Tepper, Amy N Esler, Vassilios Morellas, Nikolaos P Papanikolopoulos, and Guillermo Sapiro. 2012. A computer vision approach for the assessment of autism-related behavioral markers. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics, ICDL 2012*, 2012 IEEE International Conference on Development and Learning and Epigenetic Robotics, ICDL 2012. 2012 IEEE International Conference on Development and Learning and Epigenetic Robotics, ICDL 2012 ; Conference date: 07-11-2012 Through 09-11-2012.

Fnu Nazneen, Fatima A. Boujarwah, Shone Sadler, Amha Mogus, Gregory D. Abowd, and Rosa I. Arriaga. 2010. Understanding the challenges and opportunities for richer descriptions of stereotypical behaviors of children with asd: A concept exploration and validation. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, page 67–74, New York, NY, USA. Association for Computing Machinery.

Esraa T. Sadek, Noha A. Seada, and Said Ghoniemy. 2020. A review on computer vision-based techniques for autism symptoms detection and recognition. In *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, pages 1–6.

Qandeel Tariq, Jena Daniels, Jessey Ouillon, Peter Washington, Haik Kalantarian, and Dennis Wall. 2018. Mobile detection of autism through machine learning on home video: A development and prospective validation study. *PLOS Medicine*, 15:e1002705.