Autism-Disorder Classification

Data Preprocessing:

The built-in image preprocessing function of **tflearn** was a good choice for this model in addition to using the image augmentation technique which makes several positions and varieties of each image such as rotation and flipping in different directions. These techniques helped the models to be more generalized.

1. Traditional CNN Model:

- Model Architecture:

  The traditional CNN structure is mainly composed of convolution layers, pooling layers, fully connected layers, and some activation functions. Each convolution kernel is connected to the part of feature maps. The input is connected to all the output elements in the fully connected layer. We used RGB images of size 50x50 to train the model with 100 epochs and no validation. The used activation function in the convolution layers is 'Relu' to stop forwarding negative values through the network. And 'softmax' activation is used in the fully connected layer as it will output the value between 0 and 1 based on the confidence of the model that which class the images belongs to. The used optimizer is Adam optimizer to reach to the global minima while training out model. If the model stuck in local minima while training, then the Adam optimizer will help to get out of local minima and reach global minima. The specified learning rate of the optimizer in this case is set at 0.001. Here is the followed architecture:

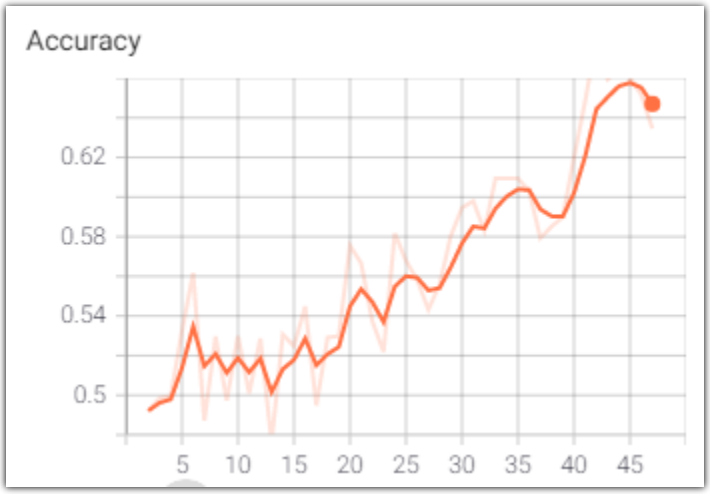
  - The input layer with shape (2536, 50, 50, 3)
  - 2D Convolution layer with 32 filters and 5 kernel size
  - 2D Max pooling layer with size 5
  - 2D Convolution layer with 64 filters and 5 kernel size
  - 2D Max pooling layer with size 5
  - 2D Convolution layer with 128 filters and 5 kernel size
  - 2D Max pooling layer with size 5
  - 2D Convolution layer with 64 filters and 5 kernel size
  - 2D Max pooling layer with size 5
  - 2D Convolution layer with 32 filters and 5 kernel size
  - 2D Max pooling layer with size 5

- Fully connected layer with size 1024

- Dropout layer by factor 0.5

- Fully Connected layer with 2 output classes

- Model Conclusion:

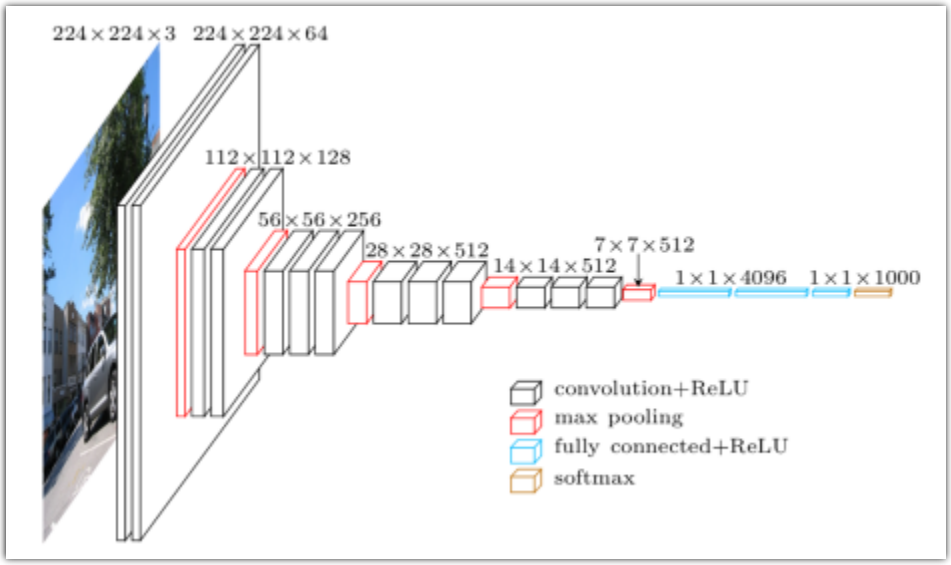


This architecture had a lot of hyperparameters tunning such as trying to train the model with grayscale images and RGB images, trying to train the model with and without validation set, trying different filter counts and sizes in the convolution layers, and the number of epochs has made a big difference in the model accuracy. Finally, the achieved accuracy of this model is **92%** training accuracy and **82%** testing accuracy.

2. VGG-16:

It is a convolution neural network architecture which is one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having many hyperparameters, they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and max pool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. All the convolution layers were with 'Relu' activation. In the end it has 2 fully connected layers followed by a 'softmax' activation for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a large network, and it has about 138 million parameters.

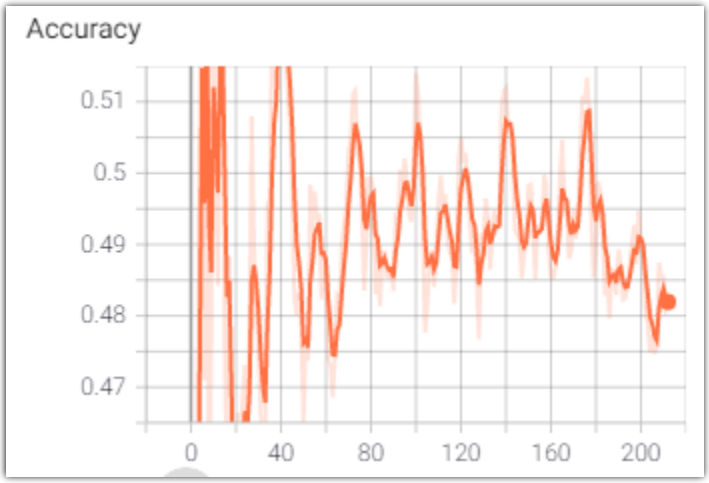
- Model Architecture:



The architecture was implemented by **tflearn** library, and it is as follows:

- 2x Convolution layer of 64 filters of 3x3 kernel and same padding
- 2D Max pooling layer with size 2 and stride 2
- 2x Convolution layer of 128 filters of 3x3 kernel and same padding
- 2D Max pooling layer with size 2 and stride 2
- 3x Convolution layer of 256 filters of 3x3 kernel and same padding
- 2D Max pooling layer with size 2 and stride 2
- 3x Convolution layer of 512 filters of 3x3 kernel and same padding
- 2D Max pooling layer with size 2 and stride 2
- 3x Convolution layer of 512 filters of 3x3 kernel and same padding
- 2D Max pooling layer with size 2 and stride 2
- Flatten layer
- 2x fully connected layers of 4096 units with 'Relu' activation followed by a dropout of 0.5
- Fully Connected layer with 2 output classes

- Model Conclusion:



We trained the model with 100 epochs and image size 50. Finally, the achieved accuracy of this model is **50%** training accuracy. It needed to be trained with more epochs and if the image size is increased, we think the accuracy will be better, but it needs high CPU specs or GPU to run this model with more than 100 epochs or with larger image size.