CLASS ACTIVATION MAPPING OF CNN FOR AUTISM DETECTION

Project Report

Priyanka Gujar — gujar.p@northeastern.edu
Muskan Khandelwal — khandelwal.mu@northeastern.edu
Himanshu Tahelyani — tahelyani.h@northeastern.edu

December 2023

1 Objective and Significance

Our project's aim is to create a reliable machine learning model for forecasting an individual's risk of developing autism spectrum disorder (ASD), a pressing issue given the increasing global prevalence of ASD. The global prevalence of ASD and the necessity for early and precise diagnosis make this initiative crucial. For those with ASD, early intervention and support can enhance their quality of life and long-term results. In order to build trust in deployed models, we need to understand how they are making the decision, and make the model more 'transparent'. To achieve this, we employ advanced techniques like image analysis with Grad-CAM and SHAP, which offer insights into the regions of input images that influence our model's decisions. Despite challenges in utilizing image datasets, our commitment is to enhance the effectiveness of ASD screening, particularly when using images that capture behavioral traits and demographic variables. Research

to enhance the effectiveness, sensitivity, specificity, and predictive accuracy of the autism spectrum disorder (ASD) screening procedure is still hampered, particularly when using picture datasets. Image databases that capture behavioral features coupled with demographic variables like gender, age, and ethnicity are very uncommon and harder to access than genetic datasets.

2 Background

Autism spectrum disorder (ASD) is a developmental disorder characterized by behavioral and communicative abnormalities. Recent studies have suggested potential facial markers associated with autism, making facial imaging an area of interest for early diagnosis. Recent advancements in machine learning have seen the application of Convolutional Neural Networks (CNNs) in ASD detection using facial images. CNNs are deep learning models capable of automatically learning and extracting features from images, making them well-suited for image-based classification tasks.

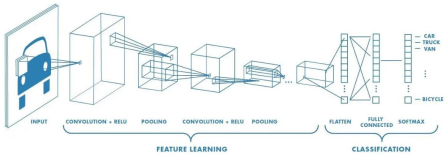


Figure 1: Architecture of CNN [1]

However, CNNs are often viewed as "black box" models, as it can be challenging to understand how and why they make specific predictions. This lack of transparency is a significant concern in applications where interpretability is crucial, such as medical diagnosis. The inspiration for model interpretability is drawn from the research done by Sami Azam, Sidratul Montaha, Kayes Uddin Fahim, A.K.M. Rakibul Haque Rafid, Md. Saddam Hossain Mukta, and Mirjam Jonkman [2]. Our approach seeks to address this challenge

by combining the power of CNNs for facial image analysis with Grad-CAM, a technique that provides insights into which regions of the input image are most influential in the model's decision. This integration not only aims for accurate ASD diagnosis but also ensures the interpretability of the model's decisions.

For 'visual explanations' of decisions from the Convolutional Neural Network (CNN) based models, we are implementing two methods: Grad-CAM and SHAP. Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept ('autism' in our case) flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept [3]. SHapley Additive exPlanations (SHAP) shows the contribution or the importance of each feature on the prediction of the model.

3 Methods

3.1 Data Acquisition

Our project begins with the acquisition and preparation of the dataset. We have collected image data for Autism detection from two distinct sources, one from Kaggle [4] containing 1470 autistic and 1470 non-autistic facial images, and second from Github [5] containing 1500 autistic and 1500 non-autistic facial images. This image data is preprocessed using various methods.

3.2 Image Resizing

As a critical step in data preprocessing, we have resizde all facial images to a consistent size, of 150 x 150 pixels. Consistency in image dimensions is paramount when working with Convolutional Neural Networks (CNNs). These uniform dimensions ensure that the model can effectively process and extract features from the images without distortion.

3.3 Image Normalization

Normalization of pixel values to a common scale is a crucial step in image data preprocessing. For our dataset containing 8-bit images, we have divided each pixel value by 255, resulting in the overall image range being [0, 1]. Normalization is crucial for several reasons. First, it ensures all input values are within a similar range, preventing any individual pixel value from dominating the learning process. Moreover, it aids in stabilizing the model's weights and biases during training, contributing to more efficient convergence and improved performance. Normalization also helps in mitigating potential issues related to differences in brightness and contrast among the input images, making the model more robust to variations in the dataset.

3.4 Data Augmentation

As our input dataset has a meager size, this might lead to the model not learning the concept well and performing poorly. To mitigate this, we have augmented our dataset by performing the following data augmentation operations: horizontal flip, zoom-in, and brightness change. This leads to the generation of three new images from each original image. These operations were chosen keeping in mind the the nature of data that the model will encounter in real life. This model works on face image data. When deployed, it is highly unlikely that the model will receive an image which is vertically flipped because of the sensitive nature of its appication. Training the model on this kind of data may lead to a reduction in model accuracy.

For achieving this, we have used the tensorflow. keras. preprocessing. image. ImageDataGenerator library.

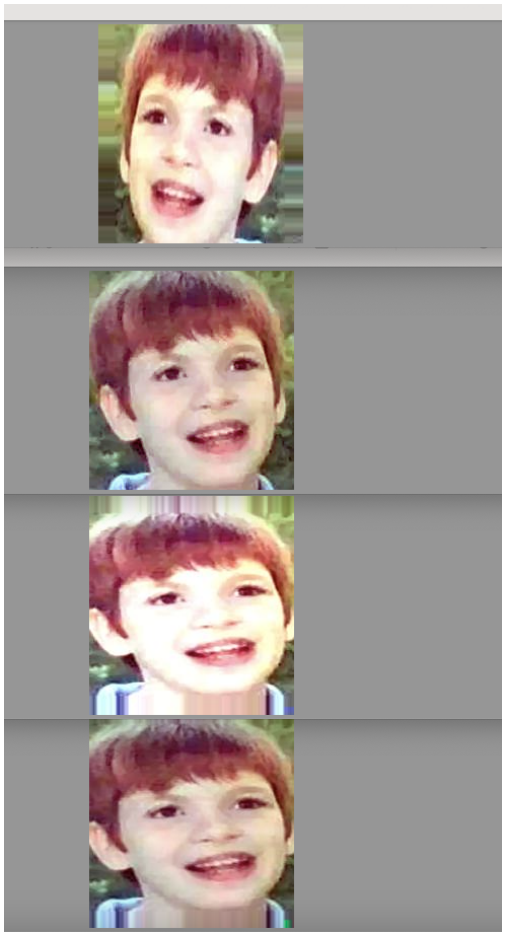


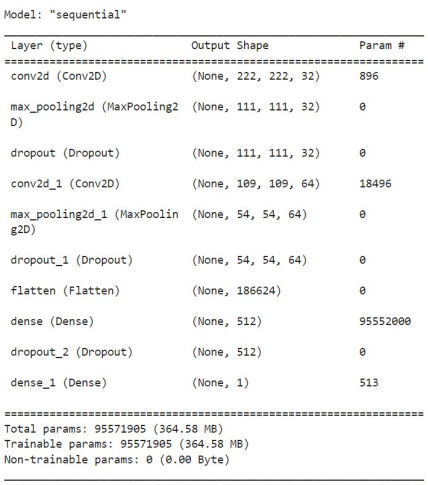
Figure 2: Data Augmentation

3.5 Model Building

In this project we have implemented three different models: CNN, VGG16 and ResNet50. CNNs are well-suited for image classification tasks due to their ability to automatically learn hierarchical features from visual data. We have created a custom implementation of CNN and utilized pretrained VGG16 and ResNe50t models for comparing the performance and understanding the models better.

3.5.1 CNN original

We have implemented a CNN model having two convoluted layers with relu activation, two pooling layers, one dense layer and one output layer with sigmoid activation. This architecture was finalized by experimenting with the number of layers as well as the number of neurons in each layer and evaluating the performance of each of those models. We observed that this architecture gave the best accuracy, precision and recall and was trained in reasonable amount of time.

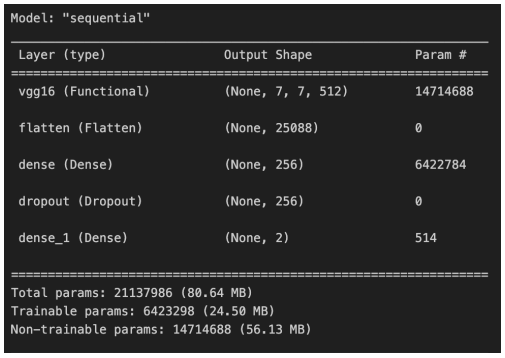


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 max_pooling2d (MaxPooling2  (None, 111, 111, 32)      0
 D)

 dropout (Dropout)           (None, 111, 111, 32)      0

 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0
 g2D)

 dropout_1 (Dropout)         (None, 54, 54, 64)        0

 flatten (Flatten)           (None, 186624)            0

 dense (Dense)               (None, 512)               95552000

 dropout_2 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 1)                 513

=================================================================
Total params: 95571905 (364.58 MB)
Trainable params: 95571905 (364.58 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3: CNN model architecture

3.5.2 VGG16

VGG16 is Visual Geometry Group pretrained CNN model which is 16 layers deep. We deployed this model to understand the performance of transfer learning on our data. We utilized the keras implementation of VGG16 without the top (fully connected) layers and created a new model on top of VGG16. We added a dense layer with 256 neurons and relu activation function, and an output layer with softmax activation.

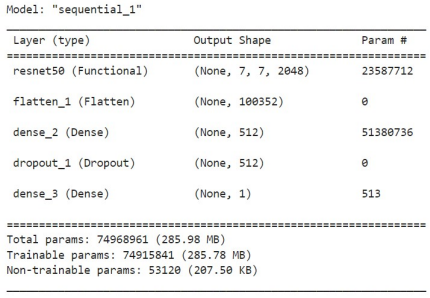


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 256)               6422784

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 2)                 514

=================================================================
Total params: 21137986 (80.64 MB)
Trainable params: 6423298 (24.50 MB)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

Figure 4: VGG16 model architecture

3.5.3 ResNet50

ResNet50, or Residual Network model is a 50 layers deep pre-trained CNN model. This is also a part of transfer learning. We utilized the keras implementation of ResNet50 without the top layers and added a dense layer with relu activation function, and an output layer with sigmoid activation.



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 flatten_1 (Flatten)         (None, 100352)            0

 dense_2 (Dense)             (None, 512)               51380736

 dropout_1 (Dropout)         (None, 512)               0

 dense_3 (Dense)             (None, 1)                 513

=================================================================
Total params: 74968961 (285.98 MB)
Trainable params: 74915841 (285.78 MB)
Non-trainable params: 53120 (207.50 KB)
_________________________________________________________________
```

Figure 5: ResNet50 model architecture

3.6 GradCAM

Grad-CAM (Gradient weighted Class Activation Mapping) is a model interpretability technique that creates heatmaps depicting which portions of the data are considered as 'important' by the model.

This technique finds the gradient of the output of desired model layer with respect to the model loss. Then, it takes sections of the gradient which contribute to the prediction, later reducing, resizing, and rescaling to overlay the heatmap with the original image

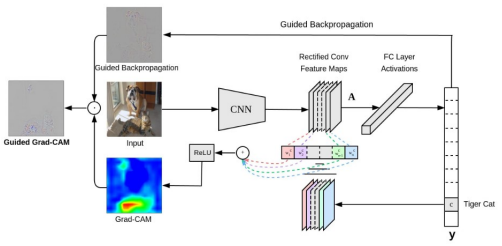


Figure 6: Grad CAM Technique

3.7 SHAP

SHAP, or SHapley Additive exPlanations, is another method for understanding models. SHAP is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions [6]. It allows local and global analysis for the dataset and problem at hand.

Figure 7: SHAP [6]

4 Results

The data was split in training, validation and test sets, and we implemented each model on the data separately. For evaluating the model, precision, recall, and f1 score are used as metrics in addition to accuracy as accuracy is often misleading.

The results on test data of the models are as follows:

4.1 CNN original

```
Model Accuracy on Test Set: 0.906547285954113

 Classification report :
              precision    recall  f1-score   support

           0       0.90      0.92      0.91       906
           1       0.91      0.90      0.90       881

    accuracy                           0.91      1787
   macro avg       0.91      0.91      0.91      1787
weighted avg       0.91      0.91      0.91      1787

Accuracy   Score :  0.906547285954113
F1 Score:  0.9042979942693411
```

Figure 8: CNN model evaluation

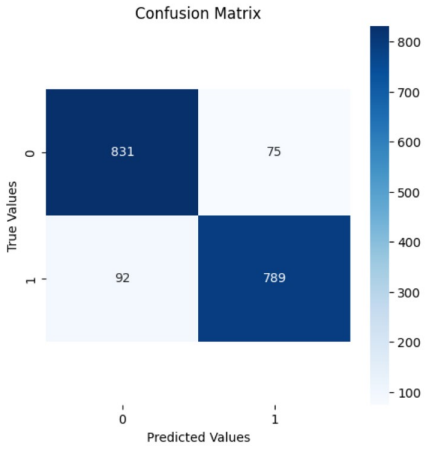


Figure 9: CNN confusion matrix

4.2 VGG16

```
Model Accuracy on Test Set: 0.8746502518186905

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.83      0.87       906
           1       0.84      0.92      0.88       881

   micro avg       0.87      0.87      0.87      1787
   macro avg       0.88      0.88      0.87      1787
weighted avg       0.88      0.87      0.87      1787
 samples avg       0.87      0.87      0.87      1787
```

Figure 10: VGG16 model evaluation

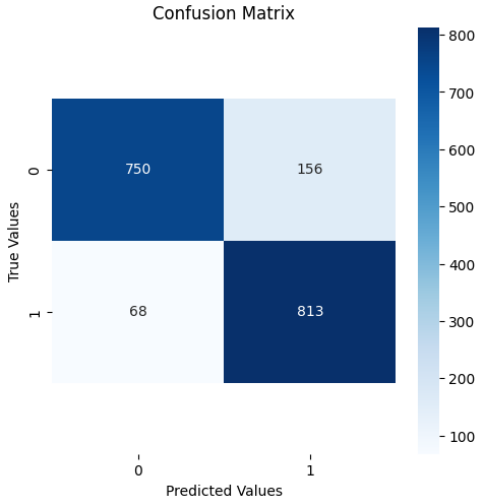


Figure 11: VGG16 confusion matrix

4.3 ResNet50

```
Model Accuracy on Test Set: 0.846670397313934

 Classification report :
              precision    recall  f1-score   support

           0       0.79      0.94      0.86       906
           1       0.92      0.75      0.83       881

    accuracy                           0.85      1787
   macro avg       0.86      0.85      0.84      1787
weighted avg       0.86      0.85      0.85      1787

Accuracy   Score :  0.846670397313934
F1 Score:  0.8283208020050125
```

Figure 12: ResNet50 model evaluation

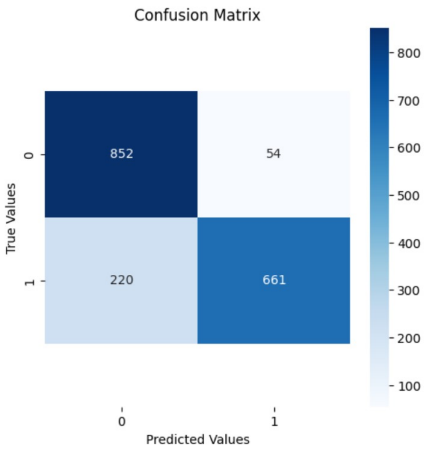


Figure 13: ResNet50 confusion matrix

4.4 Grad CAM

The implementation of Grad CAM produces the following heatmaps:

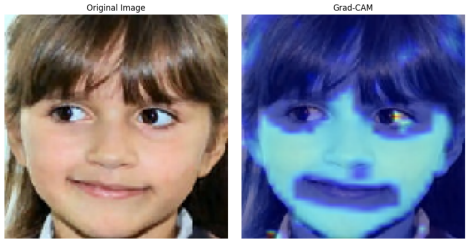


Figure 14: Grad CAM 1

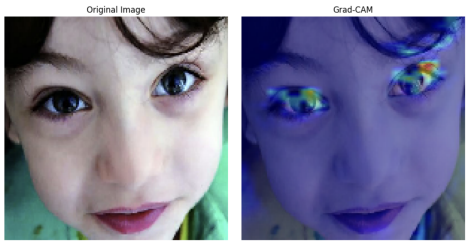


Figure 15: Grad CAM 2

4.5 SHAP

When applied with different activation functions in the output layer, SHAP gives the following results:

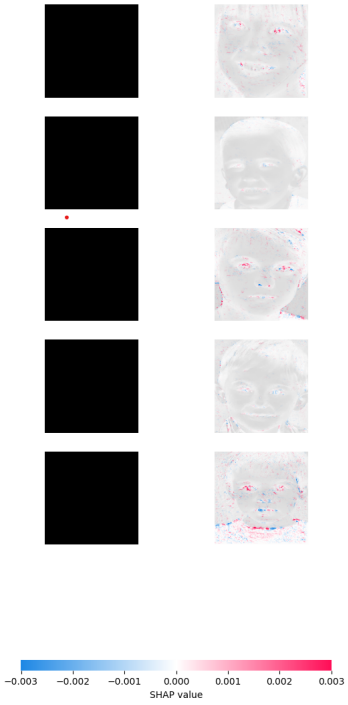


Figure 16: SHAP with sigmoid activation

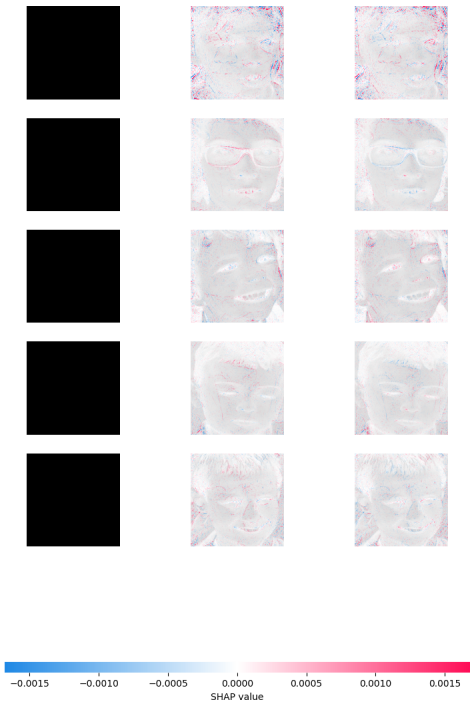


Figure 17: SHAP with softmax activation

5 Conclusion

From the evaluation, we can observe that the implemented CNN model has better values of evaluation parameters including accuracy, precision, recall and f1 score than the pre-trained VGG16 and ResNet50 models.

One possible reason might be that the VGG16 and ResNet50 models are trained on a huge dataset that includes many classes whereas the CNN model was developed and trained for this specific application.

In addition, due to their complex nature, the pre-trained models take a long time to train.

Grad CAM shows that the eyes and mouth play the most important role in the decision of the model. The SHAP plot also shows higher values in the same region.

Hence, model interpretability methods like Grad CAM and SHAP do help in undertanding the model's decision making process. They aid in uncovering the 'black box' of the model and making the model transparent.

CNNs can be used in addition to other autism detection tests for early and accurate detection of ASD.

6 Future Scope

Future developments in the field of autism prediction from images appear promising when SHAP and Grad-CAM approaches are combined.

Using an ensemble of models in

conjunction with CNNs and SHAP and Grad-CAM insights may offer a variety of viewpoints for predictions that are more reliable. Incorporating domain-specific knowledge and working with experts in neurology or psychology for a multimodal approach could strengthen the study.

Feature engineering to extract more subtle information from images can be done.

Furthermore, understanding the correlation between Grad-CAM heatmaps and SHAP values would provide more profound understanding of the models' decision-making processes.

7 Individual Tasks

7.1 Priyanka Gujar

Responsible for implementing the CNN, and training, evaluating, conducting experiments, and getting the results from the data specifically for interpretation from CNN, and creating the report. Studied the specific literature for CNN.

7.2 Muskan Khandelwal

Responsible for the data extraction and preprocessing through DataGenerators. Worked on the VGG-16 model in addition along with interpretation with GRAD-CAM as displayed in the images. Also contributed to writing the report.

7.3 Himanshu Tahelyani

Responsible for constructing the ResNet-50 model for binary feature extraction. Used Tensorflow to create the ResNet-50 architecture as mentioned. Also responsible for creating the SHAP for Autism interpretation. Also contributed to writing the report.

8 References

[1] https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/

[2] Sami Azam, Sidratul Montaha, Kayes Uddin Fahim, A.K.M. Rakibul Haque Rafid, Md. Saddam Hossain Mukta, Mirjam Jonkman. Using feature maps to unpack the CNN

'Black box' theory with two medical datasets of different modality.

[3] https://towardsdatascience.com/understand-your-algorithm-with-grad-cam-d3b62fce353

[4] https://www.kaggle.com/datasets/cihan063/autism-image-data

[5] https://github.com/mm909/Kaggle-Autism/tree/master/data/consolidated/Autistic

[6] https://shap.readthedocs.io/en/latest/

[7] https://mrsalehi.medium.com/a-review-of-different-interpretation-methods-in-deep-learning-part-1-saliency-map-cam-grad-cam-3a34476bc24d

[8] https://medium.com/ @mohamedchetoui/grad-cam-gradient-weighted-class-activation-mapping-ffd72742243a

[9] https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137

[10] https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918

[11] https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at- the-model-architecture-and-code-implementation-d8d8fa67e46f