Journal of Information Systems Applied Research

Volume 16, No. 3
November 2023
ISSN: 1946-1836

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Journal of Information Systems Applied Research** (JISAR) is a double-blind peer reviewed academic journal published by ISCAP, Information Systems and Computing Academic Professionals. Publishing frequency is three to four issues a year. The first date of publication was December 1, 2008.

JISAR is published online (https://jisar.org) in connection with CONISAR, the Conference on Information Systems Applied Research, which is also double-blind peer reviewed. Our sister publication, the Proceedings of CONISAR, features all papers, panels, workshops, and presentations from the conference. (https://conisar.org)

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the JISAR journal. Currently the acceptance rate for the journal is approximately 35%.

Questions should be addressed to the editor at editor@jisar.org or the publisher at publisher@jisar.org. Special thanks to members of ISCAP who perform the editorial and review processes for JISAR.

Journal of Information Systems Applied Research

Editors

2023 JISAR Editorial Board

Optimizing a Convolutional Neural Network Model in Amazon SageMaker for an Autism Detection Tool, EZ Autism Screener

Catherine M. Ata
atacatherine@cityuniversity.edu

Sam Chung
chungsam@cityu.edu

Brian Maeng
maengjooyol@cityu.edu

School of Technology & Computing
City University of Seattle

Abstract

Autism is a neurological and developmental disability caused by changes in the brain's development that also affects the facial tissues. Thus, children with autism show distinct facial features that are not present in average children. Studies reveal increasing prevalence of autism; however, acquiring affordable, trouble-free, and practical early screening tools is a current concern. This impacted early detection and diagnosis of autism which also influenced effective intervention. How can we employ innovative technology, like computer vision and deep learning to build an inexpensive and universally accessible autism screener to prevent late detection and diagnosis of autism? How can we enhance public access to this screening tool and minimize the difficulties involved in the assessment process? We built a basic Convolutional Neural Network (CNN) binary image classifier with seven (7) layers including the input and output layers. This initial model produced positive outcomes with a specificity score of 90.38% and this is the most important evaluation metric for health-related problems like screening for autism. We optimized this model by performing hyperparameter tuning using a cloud machine learning platform, Amazon SageMaker. The tuning job also produced a superior and robust model as reflected in the F1 score of 94.74%. It correctly classified 95% of the images. The model's specificity indicates it correctly identified 100% of those without autism as non-autistic; the recall indicates it correctly identified 90% of those with autism as autistic while its precision indicates a 100% probability that those identified by the model as autistic have autism. Tuning this model took 6 minutes. We integrated this model into a simple iOS application for mobile devices.

Keywords: autism screener, autism detection, autism screening, autism facial recognition, AWS SageMaker image classifier, CNN image classifier

1. INTRODUCTION

The main objective of this research project is to improve access to initial screening for autism or autism spectrum disorder (ASD) and to minimize the difficulty of the overall assessment process. To accomplish this, a Convolutional Neural Network (CNN) binary image classification model is built from scratch as the backend for a simple and user-friendly iOS application for mobile devices such as iPhones and iPads. This will later be expanded to Android devices to cast an even wider net.

Problem Statement

With the increasing prevalence of autism worldwide and limited access to screening tools, especially in developing countries, countless cases are detected and diagnosed late; it is also highly likely that many children with autism are undiagnosed and untreated. The negative symptoms related to this disorder may worsen and can lead to life-long problems related to developmental, learning, communication, and social abilities, and may even cause premature death (Yin, Mostafa, & Wu, 2021). Early detection is crucial, so necessary support and treatment will be provided. Most screening and diagnostic tools for autism currently available are expensive and are normally done in the clinical setting, thus requiring regular doctors' or specialists' appointments. It is not only costly but also a tedious process. This has been an ongoing issue that impacts both individuals with autism and their families. Extensive research has been conducted to resolve the assessment and intervention issues; however, it is apparent that with the rising number of cases, there is still a large need for further research (Koegel, Koegel, Ashbaugh, & Bradshaw, 2014).

This issue needs to be addressed, and based on previous and ongoing studies, we can take advantage of advanced applied science and machine learning or deep learning.

In our research, we determined how to employ a deep learning model as an autism screening tool which can be accessed by the population to bridge the gap between the resources that are presently available to the public and have a more practical, accessible, and cheaper screening tool. Our goal is to improve early detection and diagnoses and reduce undetected cases of autism in children.

Motivation

Despite ongoing extensive research, increasing assessment and intervention facilities, as well as access to support groups, there is still an existing gap in availability of these resources (Durkin, Elsabbagh, Barbaro, Gladstone, Happe, Hoekstra, Lee, Rattazzi, Stapel-Wax, Stone, Tager-Flusberg, Thurm, Tomlinson, & Shih, 2015). In fact, those resources are not easily accessible to everyone who needs them. This is mainly because of two reasons. First is the overall access; in the United States and other developed countries, they are mostly available in urban and suburban areas and are hard to access for someone from remote areas. This inconvenience may discourage those groups from initiating and seeking proper help, starting with screening for autism; in underdeveloped countries, these resources are scarce. Second is the cost and effort involved; both informal screening and formal diagnostic testing cost money. Although informal screening costs less, it still requires time and effort, while formal diagnostic testing is approximately ten (10) times more expensive, plus there is the time and effort involved.

Inaccessibility and the cost further hinder timely identification and intervention of autism in the population. As mentioned, this is of paramount importance. It can dictate the success of interventions for improving the quality of life of those with autism as well as helping families cope with the difficulties involved (Badzis & Zaini, 2014). They highlighted the importance of early detection of young children with ASD and finding remedies to help erase or minimize the symptoms and complications of this condition. The primary advantage of early identification of young children with ASD is for parents, teachers, and other people to produce strategies on how to deal with autistic kids. Koegel et al. (2014) also emphasized the importance of early detection and intervention of ASD in young children. They also suggested that the sooner the intervention is started, the better the outcome will be. Their study also discussed the short- and long-term benefits of early diagnosis and intervention. The major benefit is that ASD does not have to be a life-long disability. With early interventions, as reported by research clinics, most children can attend regular education classrooms, and some of them have lost the diagnosis. Early intervention reduces the need for more major and expensive interventions later, thus minimizing cost.

There is well-documented knowledge that children or individuals with autism share the same distinct facial features, as mentioned in the papers by Ahmed, Aldhyani, & Jadhav, 2022; Aldridge, George, Cole, Austin, Takahashi, Duan, & Miles, 2011; Beary, Hadsell, Messersmith, Hosseini, & Soltanian-Zadeh, 2022; Lu & Perkowski, 2021; Rahman & Subashini, 2022; and Sewani & Kashef, 2020. There are also several screening methods used to detect autism. However, we can leverage the advancement in technology such as image processing tools and computer vision. These can be used to process, analyze, and understand facial images of individuals with or without autism. We can also implement deep neural networks, particularly CNNs, to infer and identify if the individual in the image has autism or not (Beary et al., 2022).

Approach
Most published research studies employed transfer learning and hybrid approaches. Between the two approaches, transfer learning, specifically with MobileNet, proved to be the most promising. However, it is highly possible to build our own CNN model for the task and improve its performance by taking advantage of cloud web services for machine learning.

This work is harnessing a unique approach to optimizing the performance of a freshly built, with a considerably basic architecture, CNN model, through Amazon SageMaker hyperparameter tuning. The model is initially built, trained, and evaluated using Python and TensorFlow. The Kaggle ASD Facial Images dataset used for training the initial model is uploaded to Amazon S3 that is reformatted to a file format accepted in Amazon SageMaker. Using a script mode approach, the same model is trained in Amazon SageMaker with the reformatted image dataset. This is optimized by creating a tuning job in Amazon SageMaker. This is again fed with the reformatted dataset and is provided with ranges of chosen hyperparameters. The best-tuned model is then deployed to a mobile application for iOS devices.

Conclusions
The initial model without retraining and hyperparameter tuning provided good evaluation results. However, after automatic tuning in Amazon SageMaker, the best-tuned model outperforms the initial model, and has the same accuracy as the MobileNet used in one of the related works. With Amazon SageMaker, the tuning job takes less amount of compute time. The tuned model can also be stored in the registry, and then this can be used for similar tasks later. This model can also be retrained and tuned further using the AWS environment with less compute time.

2. BACKGROUND

Autism or autism spectrum disorder (ASD) is a neurological and developmental disability that affects how individuals interact with others, communicate, learn, and behave due to differences in the brain, which is common in most genetic conditions. Symptoms commonly appear during the first two years of birth, but they can be diagnosed at any age (NIMH, 2022). Based on numerous studies, it is believed that ASD has multiple causes and continuous studies are done to learn more about them and their impact on those with ASD. Despite all these studies, it continues to be a challenge. There has been an increasing prevalence of ASD over the years. Based on the current data from the Centers for Disease Control and Prevention (CDC), about 1 in 44 children in the United States has ASD. It occurs in all racial, ethnic, and socioeconomic backgrounds, and is known to be more common among boys than girls (CDC, 2022). And based on the World Health Organization (WHO) records as of March 2022, 1 in every 100 children worldwide has autism.

Studies suggest that permutations of genetics, environment, or the interlinkage of both changes the embryonic developmental patterns that causes alterations in the brain, which is intimately tied to the development of facial tissues. This alteration in the embryological brain results in autism. The brain is the foundation on which the various parts of the developing face grow; thus, changes to the developing brain, as we see in autism, suggest that the development of the faces of children with autism may reflect subtle facial differences compared to typically developing children (Aldridge et al., 2011).

Children or individuals with autism or ASD do not tolerate physical contact, making it difficult to perform noninvasive but direct quantitative measurements of the body. However, those with autism present similar unusual craniofacial characteristics, or dysmorphic skull and facial features, such as an unusually broad upper face with wide-set eyes, shorter middle region of the face including cheeks and nose, and broader or wider mouth and philtrum – the divot below the nose, above the top lip (Aldridge et al., 2011; Beary et al., 2022; Rahman et al., 2022). These distinct signs of autism can be useful for early recognition and detection of the disorder with innovative technology like artificial intelligence (AI), machine learning (ML), and deep learning (DL). Deep neural networks (DNN), particularly convolutional neural network (CNN) models, are known to be highly exceptional when it comes to providing solutions that require image-based and video-based analysis as well as pattern detection or recognition. That is why several works adopted transfer learning with pre-trained CNN models like EfficientNet, Inception V3, MobileNet, VGG-16, and Xception that tackled the same problem.

3. RELATED WORK

These explorations and experimentations covered in the literature review embarked upon using different forms of AI to help detect and screen for autism at an early stage. This section covers a discussion on why and how we can apply AI and

DL models as screening tools, and the different approaches to solving the problem at hand.

Literature Review

Multiple research projects tackle a similar problem, supporting the benefits of leveraging image-based classification and/or facial recognition models, particularly convolutional neural networks (CNNs). Ahmed et al. (2022) concluded that CNN image classification models are useful for early detection and diagnosis of autism by extracting those distinctive facial features from facial images and then classifying them. According to Beary et al. (2022), facial analysis can best provide early detection and diagnosis of autism based on distinct facial features. Lu et al. (2021) concluded that DL models as viable, easy, and accurate screening solutions for autism in children. Research by Rahman et al. (2022) also supports why we can use ML or DL as a screening tool for autism; according to them, CNN models are excellent at detecting hidden patterns and extracting features from colored 2D or 3D images. We can also take advantage of hybrid approaches like in the studies by Sewani et al. (2020) and by Yin et al. (2021). Sewani's work combined standard ML and DL for analyzing and classifying images of the brain taken from functional magnetic resonance imaging (fMRI) for better output. Yin's research used a similar approach by using traditional ML and AI methods like an autoencoder (AE) for advanced feature extraction on brain images produced by fMRIs.

Looking further at these research projects, they employed different approaches like transfer learning and hybrid approach as mentioned. The work by Ahmed et al. (2022) utilizes transfer learning of pre-trained CNN models like MobileNet, Xception, and Inception V3, then retrained each model on the same Kaggle ASD Facial Image dataset to extract and analyze distinctive autism facial features. Among the pre-trained models, MobileNet outperformed the rest with 100% training and 95% validation accuracy scores at 35 epochs. Interestingly, they deployed this model in a web-based app for autism detection. Similarly, Beary et al. (2022) used MobileNet and added fully connected dense layers for facial analysis and image classification to categorize images of autism vs. non-autism from the Kaggle ASD Facial Image dataset. This produced a highly performing transfer learning model with a 94.64% accuracy score at around fifteen (15) epochs on the test data. In Lu et al. (2021) research, they made use of multiple datasets: the Kaggle ASD Facial Image and East Asia ASD Children Facial Image, and retrained VGG-16 on these datasets separately and then combined the datasets. Based on that, they found that racial factors play a significant impact on the performance of the model. The results using Kaggle dataset showed 51.3% accuracy and 66.7% F1-score; using the East Asian dataset showed 95% accuracy and 95% F1-score; lastly, using the combined dataset showed 23.9% (East Asian) FP rates. Again, in the research by Rahman et al. (2022), training was done on the Kaggle ASD dataset and extracted distinct features on facial 2D images of children. They performed several experiments, training multiple pre-trained CNN models such as MobileNet, Xception, and EfficientNet as the feature extractors, and attached a deep neural network (DNN) binary image classifier to each. This yielded results with the Xception as the best performing model with 88.46% sensitivity, 91.66% specificity, 88% NPV, 92% PPV, and 96.63% AUC. On the other hand, there is a study that resorted to automatic encoding combined with computer vision for face detection and emotion and attention analysis in children (Egger, Dawson, Hashemi, Carpenter, Espinosa, Campbell, Brotkin, Schaich-Borg, Qui, Tepper, Baker, Bloomfield, & Sapiro, 2018). The paper by Egger's group only discussed their approach at a high level and used live streamed or recorded videos collected in a clinical setting in their study. The hybrid approach done in Sewani's group experimented on standard ML models like K-nearest neighbors (KNN), Support vector machine (SVM), and Random Forest (RF), and each was attached with AEs. They also experimented on CNN they built from scratch with AEs and k-fold cross-validation. All of these were trained using ABIDE (Autism Brain Imaging Data Exchange) dataset with fMRI. Another hybrid method that also made use of the ABIDE dataset was done in the research by Yin's group. They applied an AE to extract advanced features and then trained a newly built deep neural network (DNN) on it. Then they incorporated the pre-trained AE with the DNN model and trained it with raw features found in the fMRI images. The latter produced better accuracy and ROC AUC scores, 79.2% and 82.4%, respectively.

They all reinforced the documented claims that deep learning models can provide a viable, easier, and cheaper screening solution for autism in children.

There is another research that addressed a closely similar problem but focused more on genetic syndrome in general, wherein ASD is one of them (Hong, Zheng, Xin, Sun, Yang, Lin, Liu, Li, Zhang, Zhuang, Qian & Wang, 2021). The research conveyed that many genetic syndromes have unmistakable facial dysmorphias. VGG-16, a facial recognition model, was implemented to screen for genetic syndrome in children. The model's performance and outputs were compared to the professionals' screening process, and one of these comparisons was made between the VGG-16 model and a senior pediatrician with genetics training experience, yet VGG-16 outperformed the pediatrician.

Review Conclusions

The related works included here support the theory mentioned in this research that early detection of ASD will help in determining the causes and remedies for young children with ASD, and this can also increase the success of the interventions. They underpin the hypothesis that implementing innovative technology on image-based and/or video-based facial analysis like computer vision, forms of advanced automatic algorithms, and convolutional neural networks (newly built or pre-trained) can be adopted to aid early detection and diagnosis of autism. They attest that improving the accessibility of screening and diagnostic tools is highly valued, and we are proposing that with the use of deep learning and computer vision, these tools can be available to the public, which they can access through their own mobile devices. All these previous studies boost and warrant the efforts of this research.

Thus far, transfer learning on MobileNet by Ahmed, Aldyahani, and Jadhav has been the most solid CNN model for this problem, using a similar facial images dataset, with 95% accuracy. The CNN model with autoencoder using the fMRI images dataset by Sewani and Kashef also looks promising with 84.05% accuracy, 80% sensitivity, and 75.3% specificity.

Two of the studies mentioned built a new CNN model from scratch and attached it with an under-complete autoencoder to extract key features to feed to the CNN model.

Based on the results of transfer learning and hybrid approaches, we can say that building a simple CNN model can be as effective. This can also be improved with automatic model tuning or hyperparameter tuning using Amazon SageMaker.

So far, there are no documented similar research studies that have leveraged on cloud-based ML platform. This separates our research from the ones previously mentioned. There is also not a publicly and easily available autism screening tool that everyone can use, which is also the objective of our research. In addition, the best-tuned model is as accurate as the MobileNet. The best model was evaluated against other metrics, and it outperformed all the other models mentioned in other research.

Appendix A has a synthesis matrix showing a comparison of the approaches used in the different related works and in this research study.

4. APPROACH

Requirements

Several items should be in place prior to starting development. Among them are the following:

- Google Colab Pro account for unlimited access to available GPU accelerator.
- AWS account to use Amazon S3 and Amazon SageMaker web services.
- Apple account to use XCode IDE and create an iOS developer account.
- iOS developer account to be able to run the app on actual mobile devices.
- Facial images dataset with appropriate labeling: Autistic and Non-autistic.
- CNN model as the image binary classifier.

Design and Implementation: Initial Model

The first stage of development focuses primarily on the building, training, and evaluation of the model once the appropriate and good-quality image dataset is available. The initial CNN model was built in the Google Colab environment using Python programming, necessary libraries like NumPy, Pandas, Matplotlib, and ML frameworks like TensorFlow and Keras.

During the building and training stage, some of the parameters that were explored are the different layers such as the Conv2D, MaxPool, DropOut, BatchNorm, Flatten, and Dense layers. In this model, we primarily used the Conv2D to extract the features by filtering the images, and with the help of the rectified linear unit (ReLU) activation function, patterns and specific features were detected; the MaxPool layers that follow each Conv2D layer helped in condensing the image to enhance the features extracted; the DropOut layer is used to prevent overfitting by arbitrarily removing some of the units in the neural network.

Figure 1 shows the architecture of the simple CNN model for both the initial and tuned models. The model has a total of seven (7) layers including the input and the output layers. As we can see, the first Conv2D is the input layer, in this architecture, we used 32 filters with the input shape set to 100 x 100 x 3 (height, width, and color channels – RGB), a 2 x 2 kernel size, a ReLU activation function, and the padding is set to 'same' to ensure that the convolution operation is also performed in the border values. Then we added a MaxPool layer to reduce the spatial dimension of the images. The first hidden layer is composed of a Conv2D and a MaxPool layers, while the succeeding two (2) hidden layers are composed of a combination of Conv2D, MaxPool, and Dropout layers. We increased the number of filters for each Conv2D, from 64, 128, and 256. They have the same kernel size, activation function, and padding parameter settings. The DropOut layers' setting for the last two (2) hidden layers is 20%. The last DropOut layer is followed by a Flatten layer to convert the 2D array of the image feature map into a 1D flattened matrix, which is fed to the fully connected Dense layer with 512 neurons and ReLU activation function. The flattened matrix is then passed to the output layer which is a Dense layer with Sigmoid activation function (best suited for binary classification) and with 2 units or neurons wherein each corresponds to a class name or label.

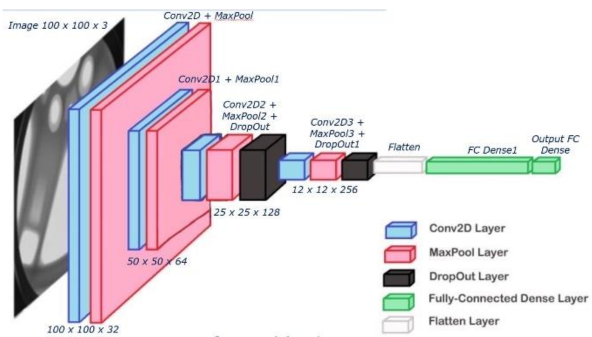


Figure 1: CNN model architecture

When training the model, to further avoid overfitting, we applied early stopping with a patience value of two (2), to ensure that after 2 epochs, if the model performance does not improve, the training is terminated. Other hyperparameters applied during training include batch size, learning rate, and the number of epochs. Although the number of epochs does not seriously affect the model's performance with early stopping, it ensures we provide enough training for the model. In this experiment, the best batch size and learning rate combination are 50 and 0.0001, respectively. We also experimented with the image size, which also impacts the model's performance. For this model, the best image size is 100 x 100.

Appendix B contains the summary of information about the CNN model, which includes the layers, the output shape of each layer, the number of weights (parameters) in each layer, the total number of parameters of the model, as well as the trainable parameters.

After training, we saved the model's weights and architecture, and then loaded the same model to perform the evaluation using a separate test data. We used several evaluation metrics to check the performance of the model. The Findings section further discussed the different evaluation metrics as well as the model's results.

Amazon SageMaker Model

Before deploying the CNN image classifier, it is necessary to ascertain that we use the best model. Thus, the second stage is geared to retraining and hyperparameter tuning which is done in a different environment; in this case, we used Amazon SageMaker, specifically, its automatic tuning or hyperparameter tuning feature.

Amazon SageMaker, as one of the cloud services offered in the AWS ecosystem, offers potential solutions to vast Data Science and machine learning projects. Some of the features it offers include access to data and pre-trained models, data analysis, building and training custom models, hyperparameter tuning, and deploying high-quality ML models at the least compute time, thus improving productivity ten (10) times more (AWS, 2022). It also offers labeling jobs for all kinds of data, including images.

Model training and hyperparameter tuning in Amazon SageMaker is known to take less time and cost, and it is highly scalable without having to manage infrastructure. The best part about hyperparameter tuning using SageMaker is it automatically adjusts thousands of algorithm parameter combinations to find the most accurate predictions. This can reduce the time and effort to get the best-performing model compared to doing it manually. It also provides built-in tools, including ML frameworks like TensorFlow, Pytorch, MxNet, and Scikit Learn, as well as open-source libraries like TensorBoard and so much more, which allows full model customization (AWS, 2022).

This is another involved process. To train the same model in SageMaker, we used a script mode approach. This is done by creating an endpoint which is the Python script that contains definitions of the model's architecture and with similar parameters used. Each parameter is initially set to lower default values. For instance, we decreased the number of epochs to ten (10), we set the batch size to thirty-two (32) and increased the learning rate to 0.001. We converted the image dataset into a compressed .npz format before using it for retraining and hyperparameter tuning of the existing model. Then, we stored the model and the compressed image dataset in Amazon S3. To further optimize the model, we performed automatic hyperparameter tuning. We added a range of values for each hyperparameter which includes the number of epochs, the learning rate, the batch size, and the optimizer. Finally, we converted the best model into a CoreML format (.mlmodel) to deploy it in an iOS mobile application.

Appendix C shows the summary of the Amazon SageMaker CNN model which is also equivalent to the initial CNN model. The total and trainable parameters are the same as the numbers found in the initial CNN model.

iOS Mobile Application
The third stage is developing the mobile application. It is a prototype with a single page that allows taking a facial image and using the integrated best-performing CNN model to identify if there is autism or not.

Appendix D shows building of the application with the deployed best autism model in XCode IDE and running it on an actual iPhone device.

Figure 2 shows a sample image taken when using the app in an actual iPhone device.

Figure 2: Taking a sample image in an actual iPhone

Technologies Used
The technologies that are used when building the entire research project include the following:

- Google Colab with GPU accelerator using Python programming and ML frameworks like TensorFlow and Keras for building, training, and evaluating the initial CNN model
- Amazon S3 for data and model storage
- Amazon SageMaker ML web service for model hyperparameter tuning
- CoreML Apple framework to access the CNN model from Amazon SageMaker and then integrate it into the iOS app.
- XCode IDE combined with Swift programming for building the iOS app

Figure 3 represents the set of the main tools used in this approach.

Figure 3: Technologies used

5. DATA COLLECTION

The dataset used for this research was originally accessible in Kaggle but can be retrieved from GitHub with the link provided under the References section. The Kaggle ASD Facial Image dataset has three (3) subsets, the training subset with 1,327 images belonging to the Autistic class and another 1,327 images belonging to the non-Autistic class; the validation subset with only forty (40) images for each class; and the test subset with 140 images for each class. However, to increase the size of the dataset, we added more images to the training subset, which we retrieved from the results of a Google search. This gave us a total of 1350 images for each class of the training subset.

Most of the images are in colored scale while the rest are in grayscale. The images also have varying sizes, facial orientation, quality, and fidelity. The images only show the faces of the children who are both boys and girls but there are more images of boys than girls with a 3:1 ratio for the autistic class and a 1:1 ratio for the non-autistic class; the children in the images belong to the age range between 2 and 14 years old, with most of them from age 2 to 8 years old. As for the race distribution, there are more Caucasian children than those of color, about a 10:1 ratio.

In this experiment, we used the updated training set, and the test set of the Kaggle ASD dataset for training and evaluating the initial and Amazon SageMaker model. We further divided the training set into training and validation sets into 80:20 split ratio, 80% for training, and 20% for validation. We did not utilize the images in the original validation subset as there are images that are duplicates of some of the images in the train subset, and this may affect the model's performance.

6. DATA ANALYSIS, VISUALIZATION, AND PREPROCESSING

Data Analysis

It is important to ensure that we have an excellent quality dataset after data collection. We performed basic analysis, like checking the count for each class to make sure there is equal distribution of the classes. We also must ensure each image has the correct shape, i.e., 100 x 100 x 3, where each value corresponds to the height, width, and color channels. We also checked that the number of labels matched the number of images for each class.

Data Visualization

Figure 4 shows random samples of images from the training dataset taken from GitHub with their original text labels while Figure 5 shows random samples of images after converting the original labels to numeric labels while training in Amazon SageMaker.

Figure 4: Sample images with original labels

Figure 5: Sample images with numeric labels used in Amazon SageMaker

Data Preprocessing

All images are set to a colored scale with three color channels: red, green, and blue (RGB). Rescaling the images is also necessary as this normalizes the RGB values of each pixel in the images to minimize the computing time required. Another important part of image preprocessing is ensuring all images have uniformity in size and are then resized to 100 x 100 pixels while considering the best resolution for this model as well as the compute power required.

Most machine learning and deep learning models are trained with a large amount of data. Since there is only one publicly available dataset for this specific research problem, the dataset size may not be sufficient. Data augmentation is convenient to use when it comes to increasing the data size. Since we are dealing with facial images, extra consideration is taken when it comes to the type of data augmentation that can be used effectively. We experimented on different data augmentation techniques like random crop, random rotation, random contrast, random flip, and random zoom. However, for this specific model, we only utilized a horizontal random flip. When using other techniques, they significantly impacted the performance of the model, making it less accurate.

7. FINDINGS

Initial Model

After multiple experiments on the different hyperparameters and among all the models built, trained, and evaluated, the performance of the CNN model that we used in this project is quite promising. Since the dataset is equally distributed, we used the accuracy score, which is the proportion of correct predictions, and the loss score, which is the prediction errors, as evaluation metrics. There are few models with better accuracy and loss scores, however, we observed overfitting and underfitting. Figures 6 and 7 show the model's training and validation accuracy and loss, respectively. Although the numbers are not magnificent, we can see that this model will generalize better with very minimal overfitting.

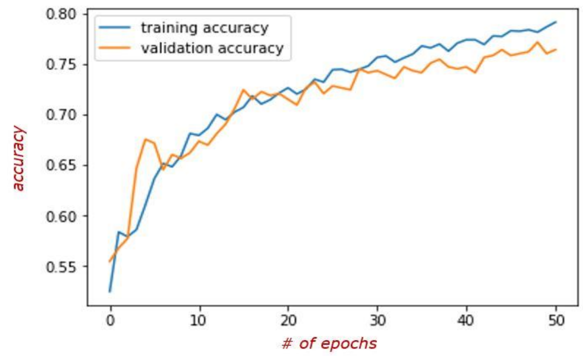


Figure 6: Training and validation accuracy scores

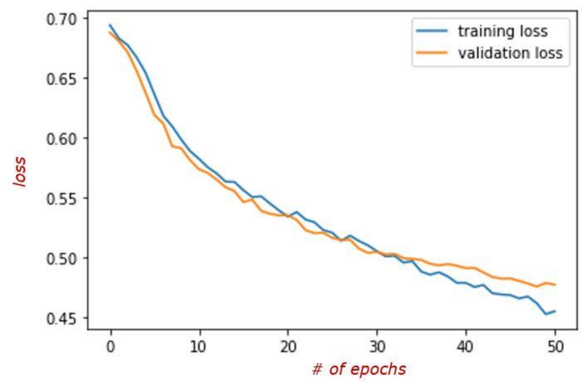


Figure 7: Training and validation loss scores

Appendix E shows the results of the prediction produced by the initial model.

When evaluating, as expected, the model's accuracy and loss scores on the test set are greater compared to the results during training, with 86.43% accuracy and 33.81% loss score, as we can see in Figure 8.

```
model_cnn.evaluate(test_set)

2/2 [==============================] - 0s 58ms/step - loss: 0.3381 - accuracy: 0.8643
[0.3381214439868927, 0.8642857074737549]
```

Figure 8: Model accuracy and loss using test data

Since we are implementing the CNN model in screening for a medical-related problem, it is recommended to use other evaluation metrics for classification tasks such as recall, precision, specificity, and F1 score. Recall or sensitivity is the true positive rate which measures the proportion of actual positives that are identified correctly by the model (Ahmed et al., 2022). This model has a recall score of 81.25%. The opposite of recall is specificity which is the true negative rate that measures the proportion of the actual negatives that are identified correctly by the model (Ahmed et al., 2022). The specificity score of this model is 90.38%. Precision is the positive predictive value that measures the proportion of positive identifications that are correct, and this model has a precision score of 88.64%. Last, the F1 score is the harmonic mean of precision and recall, which measures the preciseness and robustness of the CNN model, and it has an F1 score of 84.78.

Figure 9 shows the results of the mentioned metrics that were manually computed based on the prediction results.

```
Recall: 81.25
Precision: 88.64
Specificity: 90.38
F1 Score: 84.78
```

Figure 9: Initial model evaluation scores in percentage

Amazon SageMaker Model

We measured the same evaluation metrics on the tuned model in Amazon SageMaker. The best-tuned model's evaluation scores surpassed the initial model's scores as we can see in Figure 10. The tuned model has an accuracy of 95%, a recall score of 90%, a specificity score of 100%, a precision score of 100%, and an F1 score of 94.74%.

```
accuracy: 95.00
Recall: 90.00
Precision: 100.00
Specificity: 100.00
F1 Score: 94.74
```

Figure 10: Amazon SageMaker best model evaluation scores in percentage

Appendix F shows the predictions of the best-tuned model in SageMaker while Appendix G shows its hyperparameters based on the tuning job results.

Appendix H contains all the results of the tuning jobs in Amazon SageMaker. The first row of the data frame shows the best-tuned model's validation accuracy, which is the Final Objective Value, is 79.07% and is higher than the validation accuracy of the initial model. As expected, the compute time is shorter; it only took 388 seconds (about 6 and a half minutes) to retrain and tune this model.

iOS Application

Figure 11 is a sample output when using the EZ Autism Screener application in an actual iPhone device. The image used is from the Kaggle ASD Facial Images test subset. The iPhone camera was used to capture this image and after verifying of using this image, the application provided a result indicating "Autistic" as we can see on the top portion of the phone screen.

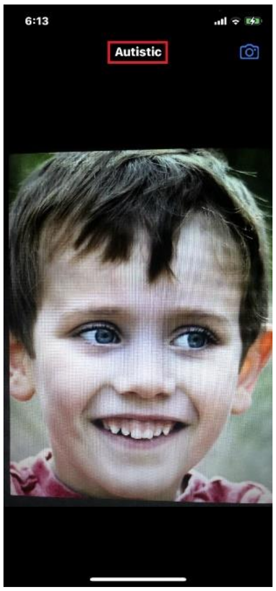


Figure 11: EZ Autism Screener iOS app result

8. CONCLUSION

Even without hyperparameter tuning, the initial model provided very promising evaluation results, most importantly the specificity score, which is an extremely critical metric in medical-related problems, like identifying autism in children. Looking at the specificity score, the model correctly identified 90.38% of those without autism as non-autistic. When diagnosing diseases or disorders, we must ensure that those predicted as negative are negative which is reflected in the specificity score. In addition, the recall score of 81.25% indicates the model correctly identified 81.25% of those with autism as autistic. The precision score of 88.64% means there is an 88.64% probability that those identified as autistic by the model have autism.

The output from Amazon SageMaker retraining and hyperparameter tuning job has exceeded our expectations and produced a superior model at a shorter compute time required for training and tuning the model, in this case, 6 minutes. Based on the accuracy score, the model correctly classified 95% of the images. With the best model's specificity score, the model correctly identified 100% of those without autism as non-autistic, thus ensuring that those predicted as negative are negative. In addition, the recall score of 90% indicates the model correctly identified 90% of those with autism as autistic. The precision score of 100% means there is a 100% probability that those identified as autistic by the model have autism. The model is also very robust and precise as the F1 score of 94.74% indicates.

A great benefit of using Amazon SageMaker is that it reduces the training time from weeks (traditional environment) to hours especially when using the correct instance type for training and tuning models. This saves hours of development. In addition, the model with the best tuning job result can be retrained and tuned further with less time and resources required. This is one of the advantages of using Amazon SageMaker.

We can leverage cloud-based machine learning platform and deep learning models with computer vision, specifically CNNs, to build a universally accessible, user-friendly, and inexpensive tool to screen for autism in children. This is done by integrating the superior model produced from Amazon SageMaker hyperparameter tuning to a mobile app. This broadens access to early detection of autism thereby improving chances of early and effective intervention.

9. FUTURE WORK

We need to improve the application to allow for storing images into Amazon S3 to collect more data with users' consent and approval and if permitted by regulations on collecting sensitive data. This will create a quality dataset of facial images to be available and help with future research on the specific problem. In addition to iOS devices, we need to expand the application to Android devices.

10. REFERENCES

Ahmed, Z., Aldhyani, T., & Jadhav, M. (2022). Facial Features Detection System to Identify Children with Autism Spectrum Disorder: Deep Learning Models. *Hindawi Computational and Mathematical Methods in Medicine*. https://doi.org/10.1155/2022/3941049

Aldridge, K., George, I., Cole, K., Austin, J., Takahashi, N., Duan, Y., & Miles, J. (2011). Facial Phenotypes in Subgroups of Prepubertal Boys with Autism Spectrum

Disorders are Correlated with Clinical Phenotypes. *Molecular Autism, 2*(15). https://doi.org/10.1186/2040-2392-2-15

Amazon SageMaker Model Training. (n.d.). AWS. https://aws.amazon.com/sagemaker/train/

Autism. (2022, March 30). World Health Organization (WHO). Retrieved July 30, 2022, from https://www.who.int/news-room/fact-sheets/detail/autism-spectrum-disorders#:~:text=It%20is%20estimated%20that%20worldwide,prevalence%20varies%20substantially%20across%20studies

Autism Spectrum Disorder. (2022, March). National Institute of Mental Health [NIMH]. Retrieved July 6, 2022, from https://www.nimh.nih.gov/health/topics/autism-spectrum-disorders-asd#:~:text=Autism%20spectrum%20disorder%20(ASD)%20is,first%20two%20years%20of%20life

Badzis, M., & Zaini, M. F. (2014). Early Identification and Intervention of Autism Spectrum Disorder Among Young Children. *IIUM Journal of Educational Studies, 2(1), 67-89*. https://doi.org/10.31436/ijes.v2i1.25

Bauer, K., Morin, K. L., Renz, T. E. III, & Zungu, S. (2022). Autism Assessment in Low- and Middle-Income Countries: Feasibility and Usability of Western Tools. *Sage Journals*. https://doi.org/10.1177/10883576211073691

Beary, M., Hadsell, A., Messersmith, R., Hosseini, M. P., & Soltanian-Zadeh, H. (2022). Diagnosis of Autism in Children Using Facial Analysis and Deep Learning*. Frontiers in Computational Neuroscience*. https://doi.org/10.3389/fncom.2021.789998

Durkin, M. S., Elsabbagh, M., Barbaro, J., Gladstone, M., Happe, F., Hoekstra, R. A., Lee, L. C., Rattazzi, A., Stapel-Wax, J., Stone, W. L., Tager-Flusberg, H., Thurm, A., Tomlinson, M., & Shih, A. (2015). Autism screening and diagnosis in low resource settings: Challenges and opportunities to enhance research and services worldwide. *Autism Res.* *8*(5). https://pubmed.ncbi.nlm.nih.gov/26437907/

Data & Statistics on Autism Spectrum Disorder. (2022, March 02). Centers for Disease Control and Prevention (CDC). Retrieved July 5, 2022, from https://www.cdc.gov/ncbddd/autism/data.html

Egger, H. L., Dawson, G., Hashemi, J., Carpenter, K. L. H., Espinosa, S., Campbell, K., Brotkin, S., Schaich-Borg, J., Qui, Q., Tepper, M., Baker, J. P., Bloomfield, R. A. J., & Sapiro, G. (2018). Automatic emotion and attention analysis of young children at home: a ResearchKit autism feasibility study. *Npj Digital Med*, *1*(20). https://doi.org/10.1038/s41746-018-0024-6

Mm909. (2020). Kaggle-Autism [Data]. GitHub, https://github.com/mm909/Kaggle-Autism

Hong, D., Zheng, Y. Y., Xin, Y., Sun, L., Yang, H., Lin, M. Y., Liu, C., Li, B. N., Zhang, Z. W., Zhuang, J., Qian, M. Y., & Wang, S. S. (2021). Genetic syndromes screening by facial recognition technology: VGG-16 screening model construction and evaluation. *Orphanet Journal of Rare Diseases*, *16*(344). https://doi.org/10.1186/s13023-021-01979-y

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ScienceDirect*. https://doi.org/10.1016/j.icte.2020.04.010

Koegel, L. K., Koegel, R. L., Ashbaugh, K., & Bradshaw, J. (2014, December 11). *The Importance of Early Identification and Intervention for Children with Or at Risk for Autism Spectrum Disorders.* Taylor & Francis Online. Retrieved July 6, 2022, from https://doi.org/10.3109/17549507.2013.861511

Lu, A., & Perkowski, M. (2021). Deep Learning Approach for Screening Autism Spectrum Disorder in Children with Facial Images and Analysis of Ethnoracial Factors in Model Development and Application. *Brain Science*, *11*, 1446. https://doi.org/10.3390/brainsci11111446

Mussar, M. (2020, April 27). Kaggle - Autism [electronic resource: dataset]. https://github.com/mm909/Kaggle-Autism

Rahman, K. K. M., & Subashini, M. M. (2022). Identification of Autism in Children Using Static Facial Features and Deep Neural Networks. *Brain Science*, *12*(94). https://doi.org/10.3390/brainsci12010094

Sewani, H., & Kashef, R. (2020). An Autoencoder-Based Deep Learning Classifier for Efficient Diagnosis of Autism. *Children*, *7*(0182). https://doi.org/10.3390/children7100182

Yin, W., Mostafa, S., & Wu, F. X. (2021). Diagnosis of Autism Spectrum Disorder Based on Functional Brain Networks with Deep Learning. *Journal of Computational Biology*, *28(2),* 146-165. https://doi.org/10.1089/cmb.2020.0252

APPENDIX A
Synthesis Matrix of Related Work

	Ahmed, et al., (2022)	Beary, et al., (2022)	Egger, et al., (2018)	Lu, A., & Perkowski, M. (2021)	Rahman, K. K. M., & Subashini, M. M. (2022)	Sewani, H., & Kashef, R. (2020)	Yin, W., Mostafa, S., & Wu, F.X. (2021)	Ata, C. (2022)
Dataset	Kaggle ASD Facial Image	Kaggle ASD Facial Image	Videos collected from participants.	Kaggle ASD Facial Image and East Asia ASD Children Facial Image	Kaggle ASD Facial Image	ABIDE	ABIDE	Kaggle ASD Facial Image
Model	Transfer learning: MobileNet, Xception, and Inception V3.	Transfer learning: MobileNet	Autoencoding with computer vision for analysis.	Transfer learning: VGG-16 trained it in 2 different datasets.	Transfer learning: MobileNet, Xception, and EfficientNet.	Hybrid approach: Autoencoder-KNN, Autoencoder-SVM, Autoencoder-Random Forest, & Autoencoder-CNN with k-fold cross-validation	Hybrid approach: DNN trained on the advance features extracted by the AE. DNN with pretrained AE on raw features found in the MRI results.	Simple CNN model with data augmentation and AWS SageMaker hyperparameter tuning
Evaluation	Best model: MobileNet Training: 100% accuracy at 35 epochs Validation: 95% accuracy at 35 epochs	Test: 94.64 accuracy score at around 15 epochs.		***Kaggle dataset:*** 51.3% accuracy, 66.7% F1-score, 75% (African American) and 86.7% (East Asian) FP rates ***East Asian dataset:*** 95% accuracy, 95% F1-score, 6.67% FP rate ***Combined:*** 23.9% (East Asian) FP rates	Best model: Xception Sensitivity - 88.46% Specificity - 91.66% NPV - 88% PPV - 92% AUC - 96.63%	Best performing model: Autoencoder-CNN 84.05% Accuracy 80% Sensitivity 75.3% Specificity	DNN with AE-extracted advance features: 76.2% accuracy 79.7 ROC AUC score. DNN with pre-trained AE on raw features from fMRI 79.2% accuracy and 82.4% ROC AUC score.	No hyperparameter tuning Accuracy: 86.43% Recall: 81.25% Precision: 88.64% Specificity: 90.38% F1 Score: 84.75% With hyperparameter tuning: Accuracy: 95% Recall: 90% Precision: 100% Specificity: 100% F1 Score: 94.74%
Deployment	MobileNet deployment to web app.		iOS Research Kit for iOS used in clinical setting.					iOS mobile app for public use.

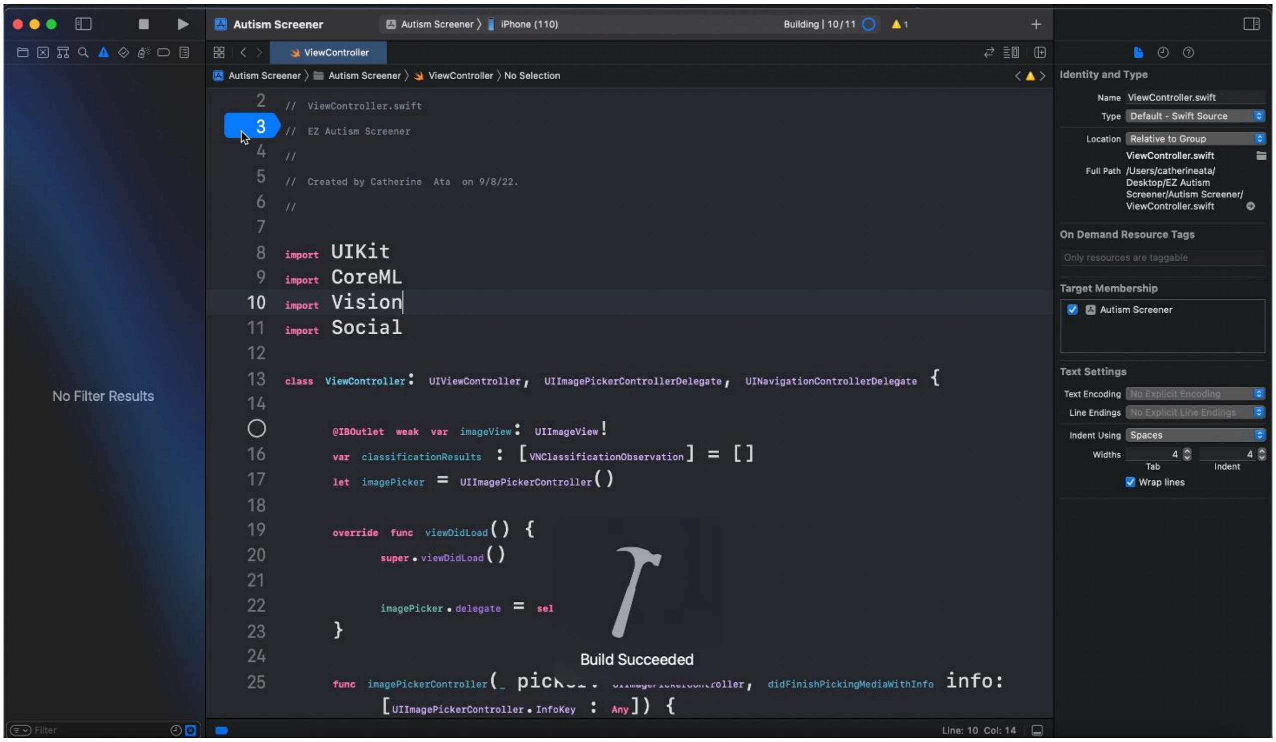
APPENDIX B
Initial CNN Model Summary

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, 100, 100, 3)       0

 rescaling (Rescaling)       (None, 100, 100, 3)       0

 conv2d (Conv2D)             (None, 100, 100, 32)      416

 max_pooling2d (MaxPooling2D  (None, 50, 50, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 50, 50, 64)        8256

 max_pooling2d_1 (MaxPooling  (None, 25, 25, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 25, 25, 128)       32896

 max_pooling2d_2 (MaxPooling  (None, 12, 12, 128)      0
 2D)

 dropout (Dropout)           (None, 12, 12, 128)       0

 conv2d_3 (Conv2D)           (None, 12, 12, 256)       131328

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 dropout_1 (Dropout)         (None, 6, 6, 256)         0

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 512)               4719104

 dense_1 (Dense)             (None, 2)                 1026

=================================================================
Total params: 4,893,026
Trainable params: 4,893,026
Non-trainable params: 0
```

APPENDIX C
Amazon SageMaker CNN Model Summary

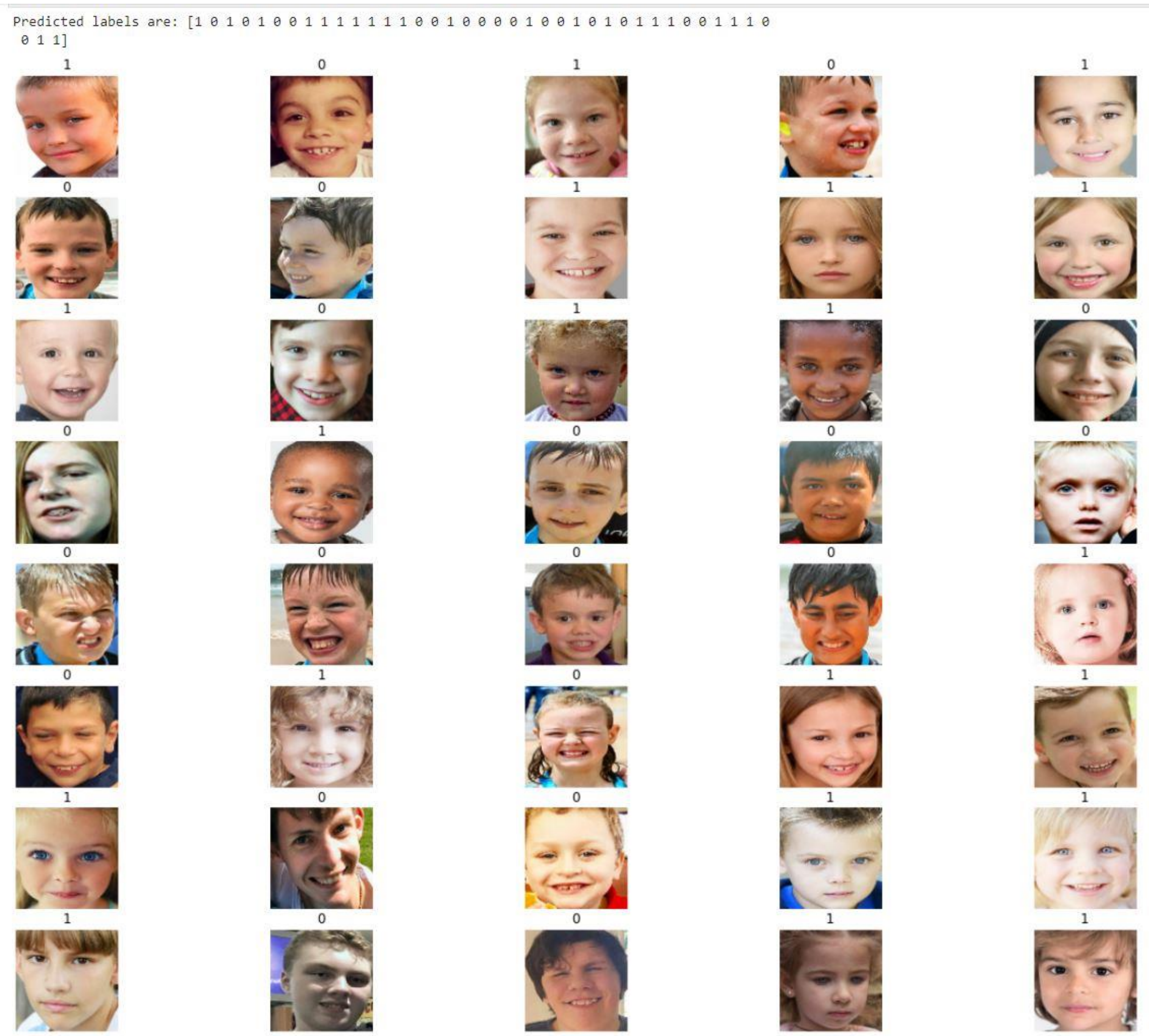
```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
conv2d (Conv2D)             (None, 100, 100, 32)      416
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 50, 50, 32)       0
_________________________________________________________________
conv2d_1 (Conv2D)           (None, 50, 50, 64)        8256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 25, 25, 64)       0
_________________________________________________________________
dropout (Dropout)           (None, 25, 25, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)           (None, 25, 25, 128)       32896
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 128)      0
_________________________________________________________________
dropout_1 (Dropout)         (None, 12, 12, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)           (None, 12, 12, 256)       131328
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 6, 6, 256)        0
_________________________________________________________________
dropout_2 (Dropout)         (None, 6, 6, 256)         0
_________________________________________________________________
flatten (Flatten)           (None, 9216)              0
_________________________________________________________________
dense (Dense)               (None, 512)               4719104
_________________________________________________________________
dense_1 (Dense)             (None, 2)                 1026
=================================================================
Total params: 4,893,026
Trainable params: 4,893,026
Non-trainable params: 0
```

APPENDIX D
Building the Application with the Best Autism Model in XCode



APPENDIX E
Initial Model Predictions vs Labels

APPENDIX F
Amazon SageMaker Best-Tuned Model Sample Predictions



APPENDIX G
Hyperparameters of the Best-Tuned Model in Amazon SageMaker

Hyperparameters

Key	Value
_tuning_objective_metric	val_acc
batch-size	32
epochs	50
learning-rate	0.0010000000000000002
model_dir	"s3:/ /tensorflow-training-2022-08-30-10-03-16-889/model"
optimizer	"nag"
sagemaker_container_log_level	20
sagemaker_estimator_class_name	"TensorFlow"
sagemaker_estimator_module	"sagemaker.tensorflow.estimator"
sagemaker_job_name	"tensorflow-training-2022-08-30-10-03-16-889"
sagemaker_program	"train-cnn.py"
sagemaker_region	"us-east-1"
sagemaker_submit_directory	"s3:// /tensorflow-training-2022-08-30-10-03-16-889/source/sourcedir.tar.gz"

APPENDIX H
AmAmazon SageMaker Tuning Job Results

	batch-size	epochs	learning-rate	optimizer	TrainingJobName	TrainingJobStatus	FinalObjectiveValue	TrainingStartTime	TrainingEndTime	TrainingElapsedTimeSeconds
2	32.0	50.0	0.001000	"nag"	tensorflow-training-220830-1003-028-e05ae715	Completed	0.7907	2022-08-30 10:51:47+00:00	2022-08-30 10:58:15+00:00	388.0
3	85.0	25.0	0.001405	"sgd"	tensorflow-training-220830-1003-027-909a63d9	Completed	0.7796	2022-08-30 10:48:43+00:00	2022-08-30 10:51:15+00:00	152.0
1	84.0	19.0	0.001722	"sgd"	tensorflow-training-220830-1003-029-dceb1e33	Completed	0.7796	2022-08-30 10:51:49+00:00	2022-08-30 10:53:56+00:00	127.0
21	32.0	28.0	0.001000	"sgd"	tensorflow-training-220830-1003-009-1f348e2b	Completed	0.7722	2022-08-30 10:17:23+00:00	2022-08-30 10:21:10+00:00	227.0
13	37.0	23.0	0.001122	"sgd"	tensorflow-training-220830-1003-017-8f73bbbb	Completed	0.7704	2022-08-30 10:32:26+00:00	2022-08-30 10:35:44+00:00	198.0
18	32.0	16.0	0.001263	"sgd"	tensorflow-training-220830-1003-012-993df826	Completed	0.7648	2022-08-30 10:23:34+00:00	2022-08-30 10:26:11+00:00	157.0
5	84.0	45.0	0.001403	"sgd"	tensorflow-training-220830-1003-025-3787c11b	Completed	0.7630	2022-08-30 10:44:07+00:00	2022-08-30 10:48:14+00:00	247.0
0	34.0	37.0	0.001016	"adam"	tensorflow-training-220830-1003-030-43d1b4c1	Completed	0.7611	2022-08-30 10:54:25+00:00	2022-08-30 10:59:17+00:00	292.0
22	96.0	42.0	0.001367	"rmsprop"	tensorflow-training-220830-1003-008-208734dc	Completed	0.7611	2022-08-30 10:17:06+00:00	2022-08-30 10:20:53+00:00	227.0
8	100.0	34.0	0.001000	"adam"	tensorflow-training-220830-1003-022-fd6129f5	Completed	0.7593	2022-08-30 10:40:25+00:00	2022-08-30 10:43:32+00:00	187.0
11	46.0	25.0	0.001458	"sgd"	tensorflow-training-220830-1003-019-b3b0c931	Completed	0.7593	2022-08-30 10:35:33+00:00	2022-08-30 10:38:35+00:00	182.0
12	83.0	26.0	0.001154	"sgd"	tensorflow-training-220830-1003-018-e1ba2dbc	Completed	0.7593	2022-08-30 10:32:27+00:00	2022-08-30 10:35:15+00:00	168.0
20	100.0	12.0	0.001000	"sgd"	tensorflow-training-220830-1003-010-ccbb1979	Completed	0.7537	2022-08-30 10:21:40+00:00	2022-08-30 10:23:17+00:00	97.0
17	32.0	13.0	0.001203	"sgd"	tensorflow-training-220830-1003-013-a3dd9f0f	Completed	0.7444	2022-08-30 10:24:43+00:00	2022-08-30 10:26:55+00:00	132.0
19	32.0	19.0	0.001059	"nag"	tensorflow-training-220830-1003-011-a4654592	Completed	0.7426	2022-08-30 10:21:42+00:00	2022-08-30 10:24:30+00:00	168.0
24	86.0	46.0	0.001804	"sgd"	tensorflow-training-220830-1003-006-6541ddeb	Completed	0.7389	2022-08-30 10:13:05+00:00	2022-08-30 10:17:07+00:00	242.0
15	32.0	35.0	0.001470	"sgd"	tensorflow-training-220830-1003-015-816f559e	Completed	0.7315	2022-08-30 10:27:10+00:00	2022-08-30 10:31:47+00:00	277.0
10	44.0	50.0	0.001057	"sgd"	tensorflow-training-220830-1003-020-99c8ea66	Completed	0.7315	2022-08-30 10:35:59+00:00	2022-08-30 10:42:27+00:00	388.0
4	45.0	34.0	0.001293	"sgd"	tensorflow-training-220830-1003-026-92fbe283	Completed	0.7296	2022-08-30 10:46:24+00:00	2022-08-30 10:51:22+00:00	298.0
9	64.0	5.0	0.001389	"sgd"	tensorflow-training-220830-1003-021-f78d51ec	Completed	0.7241	2022-08-30 10:39:02+00:00	2022-08-30 10:40:11+00:00	69.0
14	100.0	19.0	0.001664	"rmsprop"	tensorflow-training-220830-1003-016-f65b970b	Completed	0.7204	2022-08-30 10:29:34+00:00	2022-08-30 10:31:57+00:00	143.0
6	100.0	50.0	0.001557	"rmsprop"	tensorflow-training-220830-1003-024-1d396641	Stopped	0.5093	2022-08-30 10:43:48+00:00	2022-08-30 10:44:55+00:00	67.0
23	96.0	25.0	0.008969	"rmsprop"	tensorflow-training-220830-1003-007-b633d3e9	Completed	0.5093	2022-08-30 10:14:01+00:00	2022-08-30 10:16:39+00:00	158.0
25	96.0	30.0	0.008933	"rmsprop"	tensorflow-training-220830-1003-005-126798da	Completed	0.5093	2022-08-30 10:10:31+00:00	2022-08-30 10:13:28+00:00	177.0
16	32.0	50.0	0.001000	"rmsprop"	tensorflow-training-220830-1003-014-418cc33c	Stopped	0.4907	2022-08-30 10:26:34+00:00	2022-08-30 10:27:57+00:00	83.0
26	66.0	44.0	0.003749	"adam"	tensorflow-training-220830-1003-004-4f193d33	Completed	0.4907	2022-08-30 10:08:42+00:00	2022-08-30 10:12:50+00:00	248.0
27	33.0	17.0	0.008287	"rmsprop"	tensorflow-training-220830-1003-003-5f9c0753	Completed	0.4907	2022-08-30 10:07:38+00:00	2022-08-30 10:10:10+00:00	152.0
28	92.0	33.0	0.006509	"rmsprop"	tensorflow-training-220830-1003-002-8e0680c7	Completed	0.4907	2022-08-30 10:05:02+00:00	2022-08-30 10:08:29+00:00	207.0
29	39.0	10.0	0.003516	"adam"	tensorflow-training-220830-1003-001-c6a3bb88	Completed	0.4907	2022-08-30 10:04:44+00:00	2022-08-30 10:07:13+00:00	149.0
7	49.0	32.0	0.001465	"sgd"	tensorflow-training-220830-1003-023-94c757e5	Stopped	0.4815	2022-08-30 10:43:02+00:00	2022-08-30 10:43:55+00:00	53.0