technologies

MDPI

Article

Efficient Deep Learning-Based Data-Centric Approach for Autism Spectrum Disorder Diagnosis from Facial Images Using Explainable AI

Mohammad Shafiul Alam [1], Muhammad Mahbubur Rashid [1], Ahmed Rimaz Faizabadi [1], Hasan Firdaus Mohd Zaki [1], Tasfiq E. Alam [2], Md Shahin Ali [3], Kishor Datta Gupta [4] and Md Manjurul Ahsan [2,*]

[1] Department of Mechatronics Engineering, International Islamic University, Jln Gomak, Kuala Lumpur 53100, Malaysia; alam.s@live.iium.edu.my (M.S.A.); mahbub@iium.edu.my (M.M.R.); ahmed.rimaz@live.iium.edu.my (A.R.F.); hasanzaki@iium.edu.my (H.F.M.Z.)
[2] Industrial and Systems Engineering, University of Oklahoma, Norman, OK 73019, USA; tasfiq@ou.edu
[3] Department of Biomedical Engineering, Islamic University, Kushtia 7003, Bangladesh; shahinbme.iu@gmail.com
[4] Computer and Information Science, Clark Atlanta University, Atlanta, GA 30314, USA; kgupta@cau.edu
* Correspondence: ahsan@ou.edu

Abstract: The research describes an effective deep learning-based, data-centric approach for diagnosing autism spectrum disorder from facial images. To classify ASD and non-ASD subjects, this method requires training a convolutional neural network using the facial image dataset. As a part of the data-centric approach, this research applies pre-processing and synthesizing of the training dataset. The trained model is subsequently evaluated on an independent test set in order to assess the performance matrices of various data-centric approaches. The results reveal that the proposed method that simultaneously applies the pre-processing and augmentation approach on the training dataset outperforms the recent works, achieving excellent 98.9% prediction accuracy, sensitivity, and specificity while having 99.9% AUC. This work enhances the clarity and comprehensibility of the algorithm by integrating explainable AI techniques, providing clinicians with valuable and interpretable insights into the decision-making process of the ASD diagnosis model.

Keywords: deep learning; convolutional neural network; ASD diagnosis; augmentation; facial image

Citation: Alam, M.S.; Rashid, M.M.; Faizabadi, A.R.; Mohd Zaki, H.F.; Alam, T.E.; Ali, M.S.; Gupta, K.D.; Ahsan, M.M. Efficient Deep Learning-Based Data-Centric Approach for Autism Spectrum Disorder Diagnosis from Facial Images Using Explainable AI. *Technologies* **2023**, *11*, 115. https://doi.org/10.3390/technologies11050115

Academic Editors: Yudong Zhang and Zhengchao Dong

Received: 28 June 2023
Revised: 13 August 2023
Accepted: 21 August 2023
Published: 29 August 2023

1. Introduction

Autism spectrum disorder (ASD) is a neurological disorder that severely impairs the communication skills necessary for regular living. Most people with autism have mild difficulties but occasionally severe ones that necessitate specialized care. As a result of their difficulties communicating with others, people with ASD often struggle in social situations. Most of the neurophysiological symptoms of ASD are known to medical professionals, but no definitive biosignature or pathological technique can diagnose autism at any time [1]. Despite the absence of a specific treatment protocol, receiving a diagnosis at a very early age can improve outcomes significantly. Children with ASD may have a better chance of improving their socializing skills in early childhood with proper intervention due to greater flexibility in brain development at this age. Scientific evidence suggests that children who receive medical care before age four have a higher average IQ than those who wait until they are older [2]. Despite these efforts, a new study estimates that only 34% of children with ASD are identified by the age of three in the United States. However, the proportion is substantially lower in underdeveloped nations [3]. Currently, there is no particular treatment protocol for ASD. However, specialists have carefully explored several intervention techniques to minimize symptoms, enhance cognitive capacity, and improve daily living skills. Early and precise identification of ASD is essential for successfully implementing

various intervention modalities. The conventional interview-based diagnosis methods, the Autism Diagnostic Observation Schedule (ADOS) and the Autism Diagnostic Interview-Revised (ADI-R), have been considered a golden standard in this regard [4]. These methods primarily depend on the skilled physicians and the precision of the information provided by patients' attendants or the parents. Although highly dependable, human bias may reduce the accuracy of these procedures. Recent advances in artificial intelligence have prompted the desire to implement it in this advanced medical diagnosis system. AI can improve the accuracy and efficiency of medical diagnoses by providing doctors with valuable information and insights that can aid in their decision-making processes [5].

1.1. Literature Review

The accurate and early diagnosis of autism spectrum disorder (ASD) is crucial for facilitating timely intervention and providing individualized care for affected individuals. Rapid advances in deep learning techniques in recent years have ushered in a new era of medical image analysis, particularly in the context of ASD detection using facial images. This section offers a comprehensive analysis of modern deep-learning approaches employed in diagnosing autism spectrum disorder (ASD) through facial imaging. The focus is particularly placed on the crucial significance of data pre-processing in the domain of medical image analysis.

1.1.1. Deep Learning-Based Method for ASD Detection

Recent works in the literature have demonstrated that methods based on deep learning can play a significant role in the diagnosis of ASD. The use of neuroimaging data is one of the most investigated methods for diagnosing ASD in recent studies, as compared to interview-based methods, which are considered the gold standard. Structural MRI is one modality of neuroimaging data, while functional neuroimaging consists of electroencephalography (EEG). Both images are used to train various deep neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), autoencoders (AEs), generative adversarial networks (GANs), etc. [6]. The fusion of neuroimaging data from both modalities with algorithmic deep features makes the detection of ASD more robust and accurate [7]. While neuroimaging offers higher specificity and relevance for autism spectrum disorder (ASD) detection, it is expensive and time-consuming for patients to acquire the necessary images. A second method for diagnosing ASD is based on a dataset of behavioral characteristics, including capturing special activity (video) [8], eye gaze pattern [9], subsequently analyzing speech pattern [10] and handwriting [11] and so on. All of these methods initiated by behavioral datasets necessitate a substantial amount of time and extensive pre-processing steps. Rather, another very promising ASD diagnosis technique is analyzing facial features [12] using deep learning. This approach avoids causing any discomfort to children as a result of lengthy medical protocols, is devoid of human prejudice, and is inexpensive, which could potentially provide a more objective and efficient method compared to current diagnostic practices. However, the accuracy of this method is still under investigation, and more research is needed to validate its efficacy.

1.1.2. Deep Methods for ASD Diagnosis by Facial Image

Early screening for ASD from facial images can greatly benefit using convolutional neural network (CNN) models with a transfer learning approach [13]. The advantage of transfer learning is that it allows a machine learning model to leverage knowledge from a pre-trained model and apply it to a different but related problem. This can save time and resources and improve performance compared to training a model from scratch [14]. This approach avoids causing unwarranted injury to children due to lengthy medical protocols, is devoid of human prejudice, and is inexpensive. This method aims to automate the diagnosis process by analyzing facial features from individuals' images or videos, which could provide a more objective and efficient method compared to current diagnostic practices. However, the accuracy of this method is still under investigation, and more

research is needed to validate its efficacy. Recently, excellent progress has been made in screening ASD from facial images. Mohammad-Parsa et al. (2022) demonstrated the first amazing result by using the same MobileNet model to obtain 94.6% prediction accuracy in autism identification with a cleaned dataset [15]. Later, Zayed A. T. Ahmed et al. (2022) concentrated on the same study and obtained 95% accuracy with the MobileNet model [16]. B. R. G. Elshoky et al. (2022) analyzed their performance using shallow ML and deep neural networks before implementing the automated program TPOT AutoML to achieve a classification accuracy of 96.6% [17]. Taher M. Ghazal et al. (2022) used a modified version of Alexnet to create their own ASD detection model, ASDDTLA, which showed an accuracy of just 87.7% [18]. M. S. Alam et al. (2022) conducted a systematic ablation study to tune the optimizers and hyperparameters and, utilizing Xception and the optimal parameter set, reported a maximum accuracy of 95% [12]. In 2023, both Narinder Kaur et al. (2023) and M. Ikermane et al. (2023) conducted identical research with accuracies of 70% and 98%, respectively [19,20]. In every research, CNN-based models were used to extract features from the photos in the Kaggle ASD [21,22] dataset. They were pre-trained on the ImageNet dataset containing 14 million images divided into 1000 categories.

Table 1 illustrates the relatively latest research on the diagnosis of ASD using the transfer learning approach by the Kaggle ASD dataset, which consists of facial photos of youngsters. All the prior researchers concentrated principally on the model-centric approach. They focused mostly on fine-tuning CNN models with an optimal set of hyper-parameters. Not a single study could be explained in terms of particular features of facial traits causing ASD or other observational factors. However, the success of AI relies heavily on optimal training [23], and the quantity of high-quality, categorized datasets is a crucial factor in this regard. Industry experts expect that the most significant restriction of AI, the lack of high-quality data, will become increasingly apparent [24]. In order for machine learning to be effective, there must be a vast and varied dataset to analyze, and there comes the need for a data-centric approach.

Table 1. Recent research on CNN-based transfer learning algorithms for diagnosing autism spectrum disorders.

Ref.	Method	Sample Size	Accuracy (%)	Explainability	Dataset Pre-Processing	Data-Centric Approach
[15]	MobileNet	3014	94.64	none	not done	cleaning only
[16]	MobileNet	3014	95	none	not done	cleaning only
[17]	AutoML	2936	96.6	none	not done	no
[18]	ASDDTLA	2940	87.7	none	not done	no
[12]	Xception	3014	95	none	not done	cleaning only
[20]	VGG16	2940	70	none	not done	no
[19]	Densenet	2897	98	none	not done	no

1.1.3. Data Augmentation in Medical Image Analysis

To work with medical image-based datasets, researchers face a few obstacles when training deep neural network models. The availability of annotated medical images is limited, and collecting these data is expensive and time consuming. In contrast, the images from different sources vary in terms of acquisition protocols, image modalities, and image resolutions, making it challenging to standardize the data [25]. Class imbalance is another challenge, where one class dominates over the other, leading to a bias in the model [26]. Thus, deep learning (DL) models are prone to overfitting when trained on small datasets [27]. In this context, data pre-processing can be highly useful for minimizing noise and achieving a uniform image dataset size. On the other hand, augmentation techniques can assist in overcoming practically all of the aforementioned obstacles by adding new samples to the dataset [28]. Medical data augmentation refers to the process of synthesizing additional training samples from existing data to increase the size of the dataset. This technique is widely used in medical imaging to overcome the limitations of small, unbalanced, and annotated datasets [29]. One popular method of data augmentation

in medical imaging is image transformations. This involves applying various geometric and intensity transformations to the original image to generate new samples. Some commonly used image transformations include rotations, translations, flips, and scaling [30]. Another approach to medical data augmentation is data synthesization, which involves creating new data samples by combining or altering existing data. These methods can increase the dataset's size and improve the model's robustness by exposing it to variations in the data [31].

A growing body of literature demonstrates the effectiveness of dataset pre-processing and augmentation. In medical image classification, for example, researchers have found that augmenting the training dataset with random transformations can substantially improve accuracy and stability. For example, Deepak et al. 2020 [32] applied data augmentation to MRI images to detect brain tumors; after augmentation, the CNN classifier's detection accuracy increased by 6.7%. Ju et al. 2021 [33] utilized the generative adversarial network (CycleGAN) model on the UWF fundus image dataset. It demonstrated an improvement of 2.87% for precision and 4.85% for F1-score on diabetic retinopathy (DR) classification, lesion detection, and tessellated fundus segmentation after augmentation. By synthetic image augmentation technique using X-rays, D. Srivastav et al. (2021) improved the prediction accuracy of COVID-19 pneumonia detection by 3.2% [34]. No substantial literature to date has adopted the data-centric approach for ASD diagnosis with facial features, but recent research reported a 3% increment in prediction accuracy using augmented eye-tracking data [35]; consequently, we wish to investigate the benefits of pre-processing and augmentation to synthesize this dataset and evaluate the likelihood of improving performance matrices. In this study, we employ a data-centric method to screen for ASD, utilizing a facial image dataset and CNN algorithms that have been pre-trained. With a data-centric approach, we develop studies based on data manipulation. Instead of focusing on models and hyperparameter tuning, the performance evaluation is based on various data pre-processing and augmentation strategies. The major contributions of our work are as follows:

1. We developed a novel data-centric protocol using a robust data pre-processing pipeline and state-of-the-art augmentation techniques to synthesize the Kaggle ASD dataset, increasing data diversity and model generalization.
2. This study marks the pioneering effort to incorporate essential observational explainable AI factors for face-based ASD diagnosis, providing clinicians with interpretable insights into the decision-making process.
3. This study introduces an innovative image visualization method to investigate the attention of deep models on various ASD facial features for gender and pose variation, thereby substantially advancing the field of ASD diagnosis using faces.

The upcoming section of this paper is structured as follows, where Section 2 discusses the materials and methods used to create pre-trained DL models. Sections 3 and 4 contain the results and discussion on different models. Section 5 concludes the paper with the contribution of this research and future works.

2. Materials and Methods

Implementing deep neural networks in autism spectrum disorder diagnosis can involve the following steps as shown in Figure 1:

1. Data collection: The first step in using AI for ASD diagnosis is to collect a large amount of relevant data, such as patient symptoms, medical history, test results, and diagnosis information.

2. Data pre-processing: This step involves cleaning and transforming the data into a format suitable for models to process.

3. Model selection and training: Select an appropriate model and train it on the pre-processed data. The goal is to train the model to predict the diagnosis based on the input data accurately.

4. Model evaluation: Evaluate the performance of the model by testing it on a separate set of data and comparing the results to actual diagnoses. This step helps to determine the accuracy and reliability of the AI-based diagnostic system.

5. Explainability: Explainable AI concerns the ability to understand how a model arrived at a particular output and to what extent the output is trustworthy, resulting in increasing the transparency and accountability of machine learning models. This section discusses the above-mentioned steps employed to execute the experiment.

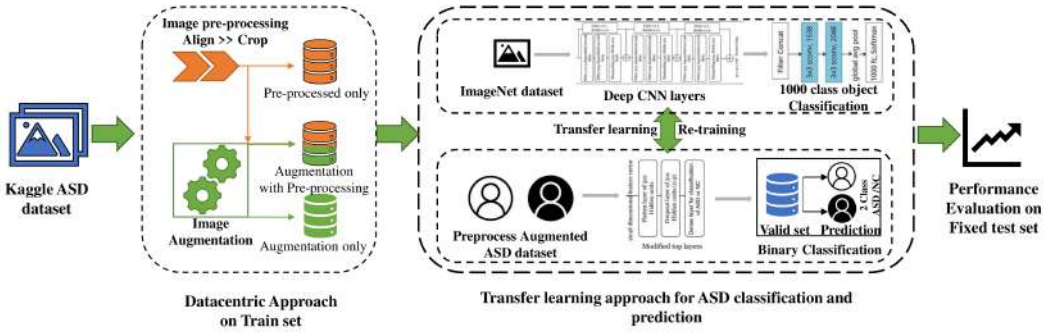

Figure 1. The workflow of the study.

2.1. Dataset

DL models require a very large training dataset to reach a high-performance level. If the model is trained for almost all possible scenarios, the model's accuracy will increase dramatically [36]. In this experiment, the autistic children dataset from the Kaggle repository [21,22] was employed to train Deep CNNs, which is named Kaggle ASD in Table 2. Kaggle ASD consists of a total of 3014 photos of children between the ages of 2 and 14, where most of them are 2 to 8 years of age. Though the number of facial images of males is three times that of the number of female populations, the ratio of the autistic and normal control class is 1:1. The contributor, Gerry Piosenka, gathered the photographs from different online sources. The dataset does not contain any medical profiling, the severity of illness, or the children's nationality. A few images are subpar in terms of facial alignment, brightness, and image size.

Table 2. Autistic children dataset from the Kaggle repository.

Dataset	Class	Number	Label
Train	Normal Control (NC)	1327	
	Autistic (ASD)	1327	
Test	Normal Control (NC)	140	NC-0
	Autistic (ASD)	140	ASD-1
Valid	Normal Control (NC)	40	
	Autistic (ASD)	40	

2.2. Transfer Learning for ASD Diagnosis

2.2.1. Dataset Pre-Processing

To maintain the accuracy and consistency of results, the dataset used for model training should include an all-inclusive group of images that depicts all conceivable scenarios for extracting ASD diagnostic features. The prior literature taught us that the photos in the dataset have noisy backgrounds and alignment issues, affecting the DL models' accuracy. To solve this issue, our dataset required cleansing and alignment [37].

2.2.2. Align Dataset

A few steps are taken to align the images, and CNN is employed. Face recognition is performed as an auxiliary task using multi-task cascaded convolutional networks (MTCNNs), a deep CNN designed as both a face detection and alignment solution. The

MTCNN consists of three stages of CNNs that can identify faces and landmarks, such as the eyes, nose, and mouth [38]. A fully convolutional network (FCN) is in the initial stage of MTCNN, called P-Net. This FCN differs from ordinary CNNs in lacking dense layers at all stages of its architecture. Bounding box vectors are created around the desired face objects, and the overlapping regions are excluded to reduce the number of boxes. The CNN layer is required to further reduce the number of bounding boxes by merging the overlapped region employing non-maximum suppression (NMS). This CNN is called R-Net, which defines whether the input image is a face and returns ten element vectors to locate the landmarks of a face. The last stage, called O-Net, is very similar to R-Net, which returns the five landmarks of the face—the left eye, right eye, nose, left corner of the mouth, and right corner of the mouth. The first task of this process is face identification, where the cross-entropy loss for each sample is given by

$$L_i^{\text{det}} = -(y_i^{\text{det}} \log(p_i) + (1 - y_i^{\text{det}})(1 - \log(p_i))) \quad (1)$$

where p_i gives the probability of sample $i = 0, 1, \ldots, n$ being a face which the P-Net decides, and the y_i^{det} is the ground truth level.

For R-Net to create a bounding box, four corners of the box must be located, which is treated as a regression problem, and the Euclidean loss for each sample is calculated by

$$L_i^{\text{box}} = ||\hat{y}_i^{\text{box}} - y_i^{\text{box}}||_2^2 \quad (2)$$

where $\hat{y}_i^{\text{box}}$ is the desired level obtained from the neural network and y_i^{box} is the ground level coordinate. For making the bounding box, four co-ordinate-like top, width, and height are required, so $y_i^{\text{box}} \in \mathbb{R}^4$.

In the last steps, the Euclidean loss is again minimized as per the below equation to formulate the task of face landmarks detection:

$$L_i^{\text{landmark}} = ||\hat{y}_i^{\text{landmark}} - y_i^{\text{landmark}}||_2^2 \quad (3)$$

where $\hat{y}_i^{\text{landmark}}$ is the co-ordinates of facial landmarks—left eye, right eye, nose, left corner of the mouth, and right corner of the mouth—and y_i^{landmark} is the ground truth co-ordinate for the ith number of images, and thus, $y_i^{\text{landmark}} \in R^{10}$.

After detecting the left and right eye coordinates, we can obtain the angle θ from the length of the triangle's three sides. The length of each edge can be found from the Euclidean distance [39]. Now the image has to be rotated anti-clockwise at an angle θ as shown in Figure 2a.

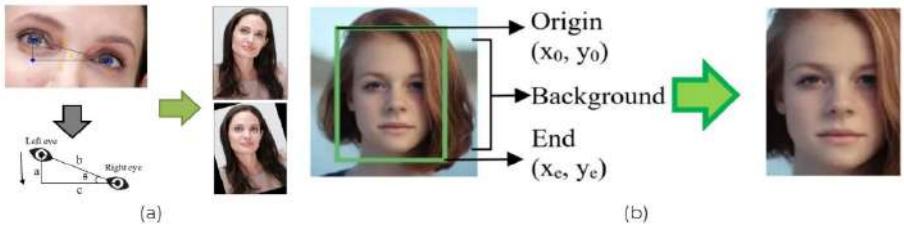

Figure 2. Facial image pre-processing. (**a**) Alignment. (**b**) Crop.

Algorithm 1 presents the pseudocode for image rotation, which takes an input path for images and an output path for rotated images. The algorithm proceeds for each image in the input path by detecting the face and landmarks using the detect_face and detect_landmarks functions, respectively. The x and y coordinates of the left and right eye landmarks are stored as xl,

yl, xr, and yr. The rotation angle θ is then calculated using the arctan function with the expression $\theta = \tan^{-1}((yr - yl)/(xr - xl))$. Finally, the image is rotated by the angle θ using the rotate_image function and saved to the output path using the save_image function.

Algorithm 1 Image rotation algorithm

Input: img_path, out_path
Output: img_a
for *img in img_path* **do**
  read_image(img);
  face[] ← detect_face(img);
  landmarks[] ← detect_landmarks(face);
  xl, yl ← landmarks['left eye'];
  xr, yr ← landmarks['right eye'];
  $\theta \leftarrow \tan^{-1}((yr - yl)/(xr - xl))$; // rotation angle
  img_a ← rotate_image(img, θ);
  save_image(img_a, out_path);
end

2.2.3. Crop Dataset

Cropping is an essential procedure for enhancing the aesthetic quality of digital photographs, as it removes undesired regions outside of a rectangular selection. The dataset includes facial images in the training set with a noisy texture and superfluous patterns in the background, which can impair the model's training. Cropping is the process of removing a portion of an image to reframe it. Similar to the alignment procedure, MTCNN is used for face recognition, creating a bounding box around the face region; for cropping, the bounding box must be precise and tightly confined to the face region only. The upper left point of the bounding box is called the origin, and the lower right corner is called the end. The pixels inside the box are copied to a new image; thus, we obtain the cropped [40] one as shown in Figure 2b.

Algorithm 2 represents a face-cropping algorithm that takes the path for the input images and the output path where cropped images are stored. The algorithm reads each image from the input path and detects the face in each image using the function detect_face. Then, it calculates the x and y coordinates of the top left corner and the bottom right corner of the face's bounding box using the function convert_xywh. The image is then cropped using the function crop_face, with the width and height calculated as the difference between the x and y coordinates. Finally, the cropped image is saved to the output path using the function save_image. The algorithm iterates through all images in the input path and applies the same process to each image.

Algorithm 2 Face-cropping algorithm

Input: img_path, out_path
Output: img_c
for *img in img_path* **do**
  read_image(img);
  face ← detect_face(img);
  xo, yo, xe, ye ← convert_xywh(face);
  img_c ← crop_face(xe − xo, ye − yo);
  save_image(img_c, out_path);
end

2.3. Dataset Augmentation

Augmentation is a technique used to enhance the amount of an existing dataset by modifying and manipulating the original data. This strategy enables the model to discover or anticipate all possible real-time data patterns. Both pre-processed and unprocessed photos from the Kaggle ASD dataset are augmented. Different types of augmentations include horizontal flip, grey cycle, resize, rotation, shear, zooming, addition and removal of noise, changing brightness, hue, saturation, etc.

2.3.1. Horizontal Flip

When working with facial images, the horizontal flip is the most typical technique for augmentation. The human face is highly symmetrical, so a single-sided feature can frequently cause confusion during training. A complete horizontal flip allows learning the features from both sides of the face. Horizontal flip is obtained from the Transform module of the Torchvision library [41]. The input images are fed from the PIL rather than tensors. The image's width, height, and pixels are obtained from the PIL library using image functions, and then the image is transposed, as shown in Figure 3.

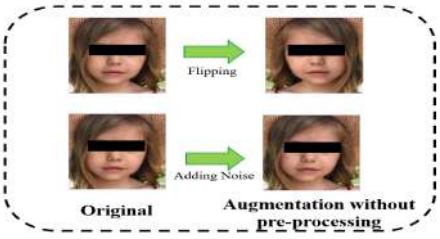

Figure 3. Facial image only augmentation pipeline.

Algorithm 3 is an image-flipping algorithm that takes an image path, an output path, and a probability as input. For each image in the input path, the algorithm reads the image and creates a new image by copying the original image into a new variable, img_f. The algorithm then loops through the width and height of the original image and checks if a random number is less than the given probability. If the condition is satisfied, the pixel value at position (x, y) in the new image is replaced by the original image's pixel value at position (width − x − 1, y).

Finally, the new image is saved to the output path.

Algorithm 3 Image-flipping algorithm

```
Input: img_path, out_path, probability
Output: img_f
for img in img_path do
    read_image(img);
     img_f ← img;
     for x ← range (0 − img[width]) do
         for y ← range (0 − img[height]) do
             if random() < probability then
                 img_f[x, y] ← copy_image_pixel(img[width] − x − 1, y);
             end
         end
    end
    save_image(img_f, out_path);
end
```

2.3.2. Add Pepper–Salt Noise

Noisy face images were identified as one of the primary causes of poor accuracy in previous investigations [12]. Some image quality is so poor that the class cannot be predicted during testing. If noise can be added manually to the training set, the model will be able to learn and extract their features. Therefore, the PIL and NumPy libraries are utilized to alter the photos. The PIL opens the input facial images, which are converted into NumPy arrays. Additionally, the user must provide the probability or density of noise imposition. Random probabilities are compared to a threshold value for every image pixel in order to place noise in the required pixels [42].

The salt-and-pepper noise algorithm, presented in Algorithm 4, takes an input path for images, an output path for the noisy images, and a probability parameter. The algorithm randomly assigns black and white pixels to generate salt-and-pepper noise on the input images. The probability parameter controls the degree of noise to be added to the input images. For each image in the input path, the algorithm reads the image, creates a new image, and then walks through each pixel. The random number between 0 and 1 is generated to compare with the threshold, which is set to be the complement of the probability. If the generated number is less than the threshold, the pixel is black (pepper noise); otherwise, if the number is less than the probability, the pixel is white (salt noise). Finally, the noisy image is saved to the output path.

Algorithm 4 Salt-and-pepper noise algorithm

```
Input: img_path, out_path, probability
Output: img_n
Threshold ← 1 − probability
 for img in img_path do
    read_image(img);
     img_n ← img;
     for x ← range (0 − img[height]) do
        for y ← range (0 − img[width]) do
            r ← random number between 0–1;
             if r < Threshold then
                img_n(x, y) ← 0; ;                    // pepper pixel (black dot)
            end
            else if r < probability then
                img_n[x, y] ← 255; ;                    // salt pixel (white dot)
            end
        end
    end
    save_image(img_n, out_path);
end
```

2.4. Convolutional Neural Networks

Since the 1980s, convolutional neural networks (CNNs) have been utilized in image classification and recognition [43]. CNNs were initiated from the research of the brain's visual cortex. CNNs have achieved superhuman performance on some challenging visual tasks in recent years because of the development in computer power, the quantity of data samples accessible for training deep neural networks, and transfer learning for user-modified classification [44,45]. The main function of the face-recognition or object-classification models is to extract an entity's features, making it feasible for the binary classification of two classes, autistic and non-autistic faces, by acquiring knowledge from a vast collection of pictures by transfer learning approach [46]. A machine learning method can be used for similar kinds of work by adapting changes in the pre-trained models' top layers. The core convolutional layers of CNN-based models, which were previously trained with the ImageNet dataset, can be used to extract the features of autistic and normal faces. The classification layers are modified for binary classification. This study is based on three pre-trained deep CNN models: MobileNetV2 [47], ResNet50V2 [48], and Xception [49]. These models were determined to perform best among contemporary works [12].

2.4.1. MobileNetV2 Model

MobileNetV2 is a lightweight deep CNN model that is thus perceived to develop mobile phone applications to implement classification tasks. The basics of this MobileNetV2 model are to establish the connection from one bottleneck layer to another [50]. The inverted residual architecture consists of 19 residual bottleneck layers; 32 full convolution layers

exist before these layers. These convolution layers perform depth-based convolutions, utilizing non-linear filter characteristics.

2.4.2. ResNet50V2 Model

ResNet50V2 comprises several units that promulgate both forwards and backward direction mapping identities and are residual in nature. Through block-to-block propagation, classification accuracy is maintained at a high level. With the assistance of these residual mappings, training will be substantially easier and more generalized. In ImageNet or COC contests, ResNet models frequently have more than 100 layers and have outstanding accuracy.

2.4.3. Xception Model

This model has a very simple modular structure based on Google's Inception model. The model comprises three primary blocks, entry, center, and exit, with separate convolutional layers and Relu activation functions for each block. The input image size is $299 \times 299 \times 3$. The input is processed by the entry flow, which extracts features of $19 \times 19 \times 728$ dimensions. The residual connections take the maximum value of each layer as output after every block. In the middle block of the feature map, the feature map remains preserved despite being passed nine times through convolution layers. The output of the final component for a standard-size input image has 2048 features. Finally, the prediction layer receives the features via an FC layer, and the modifications are made to the final layers for binary classification.

2.4.4. Regularization

Some deep neural networks can have millions of parameters, although most have thousands. This allows DL networks unprecedented flexibility and the ability to accommodate a wide range of complex datasets. However, high adaptability increases the risk of model overfitting while training the dataset. Regularization is a process that can be implemented to avoid overfitting. Some regularization techniques are early stopping, batch normalization, $\ell 1$ and $\ell 2$ regularization, dropout, and max-norm regularization [51]. Also, choosing the best optimizer is another factor that will help prevent overfitting.

In this paper, we use AdaGrad, which we obtained from the ablation study of the previous literature [12]. Gradient descent first rushes down the sharpest slope, which does not lead directly to the global optimum, before slowly proceeding down to the valley floor. If the algorithm could rightly change the direction earlier, heading more directly toward the global optimum, that would be great. In order to make this adjustment, the AdaGrad algorithm scales the gradient vector according to the equation below:

$$s \leftarrow s + \nabla_\theta J(\theta) \otimes \nabla_\theta J(\theta) \tag{4}$$

$$\theta \leftarrow \theta - \eta \nabla_\theta J(\theta) \oslash \sqrt{s + \epsilon} \tag{5}$$

Here, s is a vector whose i-th element is s_i and adds all the partial derivatives of the cost function in a square according to θ_i. Thus, after each iteration, s_i becomes larger. The next equation almost refers to the gradient descent function, but here, the factor is kept less by factor $\sqrt{s + \epsilon}$. This θ is a vector that represents the i-th element as θ_i. Thus, it is evident that this algorithm can alter the learning rate for steep slopes considerably more rapidly than for gentle slopes. Consequently, this adaptive learning rate facilitates the model's propagation towards the global optimal. So, minimal adjustment of the learning rate hyperparameter is required [52].

Dropout is another regularization technique that can be used to address the overfitting problem [53]. The dropout method is quite straightforward in that, at each layer, some neurons are disregarded so that they can be reactivated in the subsequent step. The maximum dropout rate is normally between 10% and 50%, with the probability p controlling this rate.

This is limited to 40% to 50% for CNN, whereas 50% dropout is employed in our transfer learning models just before the topmost decision layer. It considerably lessens the training load and prevents overfitting.

2.5. Explainable AI

Explainable AI is a branch of artificial intelligence that focuses on developing systems that can produce accurate predictions and provide human-understandable explanations of their decisions. Explainable AI aims to increase the transparency, accountability, and interpretability of machine learning models and algorithms [54,55].

Grad-CAM (gradient-weighted class activation mapping) is a visualization technique that can explain the predictions made by deep neural networks [56]. Grad-CAM generates a heatmap that highlights the regions of the input image that are most important for the prediction made by the network. The heatmap is generated by computing the gradients of the output class scores with respect to the feature maps in the last convolutional layer of the network. These gradients are then used to weigh the feature maps, and the resulting weighted feature maps are averaged to produce the final heatmap. Grad-CAM can be used to visualize the internal workings of deep neural networks and explain the predictions made by the network. This can be particularly useful in medical imaging, where accurate predictions are important, but it is also important to understand why the network made a certain prediction.

When working with a large number of identical image samples for recognizing or extracting a given pattern, the mean image might play a crucial role [57]. The mean image is simply the average of all images in the dataset. It is computed by summing up all pixel values of all images and dividing by the total number of images. The resulting image represents the average intensity of each pixel in the dataset.

The simple equation to calculate the mean image is

Input: *A set of N images I_1, I_2, . . ., I_N*

Output: *The mean image M*

$M = (I\_1 + I\_2 + \ldots + I\_N) / N$

The importance of mean images in explainable AI is increasingly recognized in the recent literature. One of the key benefits of the mean image in explainability is its ability to facilitate feature visualization [58]. Feature visualization refers to the process of visualizing the features that the AI system learned during training. By subtracting the mean image, we can visualize the features that the AI system has learned, which can help interpret its decision-making processes. These visualizations can also be used to identify features important for classification, which can help improve the accuracy of the AI system.

2.6. Experimental Setup

The three primary phases of this research are (1) pre-processing and augmentation of the dataset; (2) optimizing the deep CNN models with hyperparameters; and (3) assessing the models with appropriate performance metrics. The success of artificial intelligence is highly dependent on optimal training and the quantity of high quality; thus, categorized datasets are a critical aspect in this regard. The more data algorithms have to work with, the faster they may learn and enhance their judgment of future outcomes. In order for machine learning to be effective, a large and diverse data collection must be analyzed. When sufficient high-quality data are available, AI systems can readily outperform baseline methods. Thus, in this study, the experiments are mostly data-centric, and the dataset is the main focus. It is well recognized that medical datasets are very difficult to acquire, making the number of datasets required to train DL models difficult. Here, the autistic children dataset from the Kaggle repository is used, named ASD Kaggle as stated in Table 2. With a view to enhancing the training, the training set consisting of the facial images of a total of 2654 children (1327 ASD and 1327 normal) was pre-processed as described in Table 3.

Table 3. The details of the training dataset after pre-processing and augmentation.

Sl No	Processing Status	Name of the Dataset	Training Size
1	Without Pre- processing	Kaggle ASD	2654
2		Flippedk	2654
3		Noisyk	2654
4	Pre- processed	ASDp	2654
5		Flippedkp	2654
6		Noisykp	2654

The pre-processing algorithms and techniques are explained at the start of this section. The images were first aligned and then cropped at the time of processing, as indicated in Figure 4. The resulting set is named ASDp. This ASDp was fed through an algorithm that flips and adds noise to 2654 facial images, termed Flippedkp and Noisykp. Two additional datasets, Flippedk and Noisyk, were subjected to the flip and noise addition augmentations of Kaggle ASD, respectively as shown in Figure 5, for the purpose of comparison. The details of all datasets are shown in Table 3.

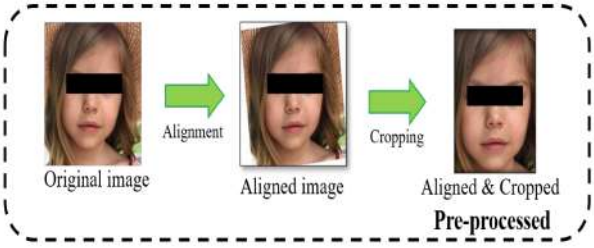

Figure 4. Facial image pre-processing pipeline.

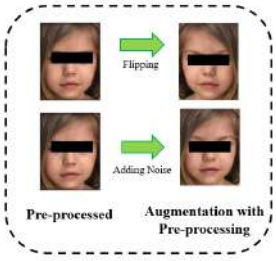

Figure 5. Facial image pre-processed augmentation pipeline.

During training, we merged the Kaggle ASD with the full training set since the Kaggle ASD is the original uncured set we obtained (Table 2), resulting in a twofold increase in the number of training sets (Table 4). As an illustration, the flipped training set for the augmentation-only approach combines the Kaggle ASD and Flippedk datasets presented in Table 3. Thus, the train column in Table 4 gives the number of training samples we obtained after pre-processing or augmentation. The subscripts in datasets labeled as All for both the augmentation-only and augmentation with pre-processing approaches represent the combination of datasets listed in Table 3 according to the provided Sl No. The test set and valid set used for testing and validation are the same across all experimental setups, as we wish to compare our results to those of contemporary research.

We employed the Deep CNN-based MobileNetV2, ResNet50V2, and Xception pre-trained using the ImageNet dataset of 4.2 billion photos of 1000 classes. The models were modified so that the prediction layer receives the features via an FC layer, and the modifications for binary classification were made to the final layers. The main reason for choosing these models is that among the recent research, only one performed a complete ablation investigation on five different models, and it was found that these three models

performed the best. The hyperparameters listed below were also fine-tuned as part of the same research, and the same values were retained for this study.

Table 4. The details of the dataset for the experiments.

Approach	Training Set	Training	Test	Valid	Labels
Augmentation Only	Flipped	5308	280	80	ASD-1
	Noisy	5308	280	80	
	All T3 (1 + 2 + 3)	7962	280	80	
Pre-processed Only	ASDP	5308	280	80	NC-0
Augmentation with Pre-processing	Flipped	5308	280	80	Ratio-1:1
	Noisy	5308	280	80	
	All T3 (1 + 4 + 5 + 6)	10,616	280	80	

We employed a handful of performance matrices to evaluate the model's effectiveness. The most evident one is the binary classification accuracy stated as "accuracy". One assessment matrix includes the area under the curve (AUC), used in some earlier studies to measure how well a model predicts outcomes. Since it is based on the ROC curve, this AUC is more convincing evidence of the model's efficacy than accuracy alone. Precision and recall, the other two matrices, reflect the accuracy with which the desired classes could be predicted. Accuracy, precision, recall, and F1-score can be expressed mathematically as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

3. Result

The codes were developed in Python and run on the Kaggle platform. The results were analyzed using several tools for data analysis, including matplotlib, sklearn, and Pandas. We trained the model using deep transfer learning from the Keras API Library [59]. The performance of three distinct DLs (MobileNetV2, ResNet50V2, and Xception) was evaluated in terms of accuracy, precision, recall, and F1-score in this work using Equations (6)–(9). The DL networks were selected based on the ablation study conducted by M. S. Alam et al. (2022) while retaining their optimal hyperparameters and optimizer [12] as shown in Table 5. The batch size was set to 32, and Adagrade was utilized as an optimizer. The convolutional neural network (CNN) was trained for a maximum of 50 epochs, utilizing a learning rate of 0.001, in order to facilitate the effective learning and accurate prediction of samples associated with autism spectrum disorder (ASD). In the context of binary classification, the loss function selected is BinaryCrossentropy. This loss function is accompanied by the use of the ReLU activation and the sigmoid function in the final layer. The best values of several performance matrices derived from various data-centric approaches are displayed with bold fonts in tables. Our experiments are primarily data-driven. The initial part of this study is titled the augmentation-only approach because the training sets were generated by applying two fundamental augmentations to the Kaggle ASD dataset: flip and noise addition. As shown in Table 4, the image was labeled as '0' for normal control (NC) children and '1' for ASD children while producing a data frame, and no dataset pre-processing was

applied. Table 6 summarizes the comparative training and test evaluation matrices of the deep learning models using the set of hyperparameters stated in Table 5.

Table 5. The list of essential parameters for model training.

Hyperparameters and Other Settings	
Number of Batch	32
Maximum Epoch	50
Optimizer	AdaGrad
Activation function	Relu
Learning rate	0.001
Loss function	BinaryCrossentropy
Classification layer	Sigmoid

Table 6. Performance of Deep CNN models for augmentation-only approach.

		Training					Test				
Augmentation	**DCNN**	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1-Score**
Flip	Xception	0.998	1.000	0.998	0.998	0.998	**0.925**	**0.979**	0.925	0.925	0.925
	ResNet50V2	0.999	1.000	0.999	0.999	0.999	0.882	0.943	0.882	0.882	0.882
	MobileNetV2	0.996	1.000	0.996	0.996	0.996	0.896	0.950	0.896	0.896	0.896
Noise	Xception	0.996	1.000	0.996	0.996	0.996	0.918	0.959	0.918	0.918	0.918
	ResNet50V2	0.998	1.000	0.998	0.998	0.998	0.900	0.952	0.900	0.900	0.900
	MobileNetV2	0.982	0.998	0.982	0.982	0.982	0.861	0.938	0.861	0.861	0.861
All	Xception	0.997	1.000	0.997	0.997	0.997	0.904	0.964	0.904	0.904	0.904
	ResNet50V2	0.999	1.000	0.999	0.999	0.999	0.900	0.955	0.900	0.900	0.900
	MobileNetV2	0.994	1.000	0.994	0.994	0.994	0.914	0.967	0.914	0.914	0.914

Flip augmentation was used for the Kaggle ASD training dataset, and the best training and testing performance was achieved. The ResNet50V2 model performed the best for training with a 99.9% accuracy and 100% AUC value, while the Xception model ranked first in testing with 92.5% accuracy and 97.9% AUC. Figure 6 displays the training and validation accuracy along with the loss graphs for all three models. The training and validation curves clearly demonstrate that the model began to overfit beyond a certain point because the validation curve is ascending and descending while the training loss is constantly dropping. Clearly, the model cannot extract features for every potential scenario from these training and validation sets.

Table 7 demonstrates the performance of the evaluation matrices after the pre-processing of the Kaggle ASD dataset. The pre-processing flow is shown in Figure 3, where the facial image is first aligned and then cropped, keeping only the facial region. The training accuracy is maximal when using ResNet50V2 with a value of 99.5% and an AUC of 100%. Similarly, for these approaches also, the testing performance is better while predicting ASD with the Xception algorithm. Accuracy is 97.9%, AUC is 99%, and the precision, recall, and F1-score are all reported to be 97.9%. Figure 7 depicts the training and validation accuracy as well as the loss graphs for the three models with a pre-processing-only approach. Unlike Figure 6, Figure 7 displays a highly orderly increase in training accuracy over time. The graph of validation accuracy is parallel to the graph of the training accuracy. For the Xception model, these two lines tend to overlap to illustrate the consistency of the training and validation trend. For the training and validation loss graphs, the pre-processing-only approach yields symmetric plots, indicating that the prior approach's overfitting is minimized.

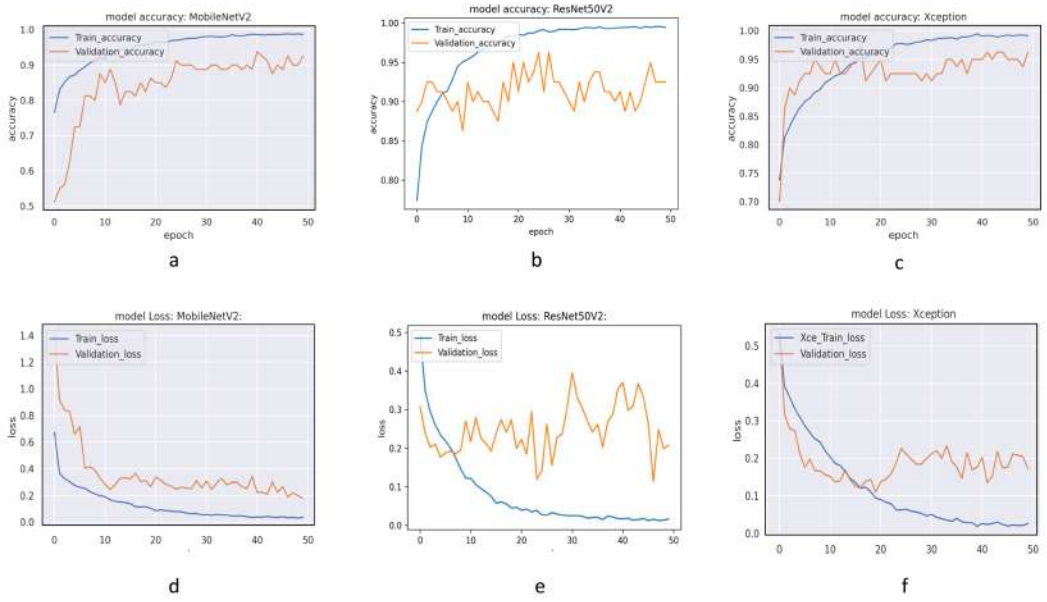

Figure 6. Training and validation accuracy plot for (**a**) MobileNetV2, (**b**) ResNet50V2, and (**c**) Xception and training and validation loss graph for (**d**) MobileNetV2, (**e**) ResNet50V2, and (**f**) Xception for the flip augmentation approach.

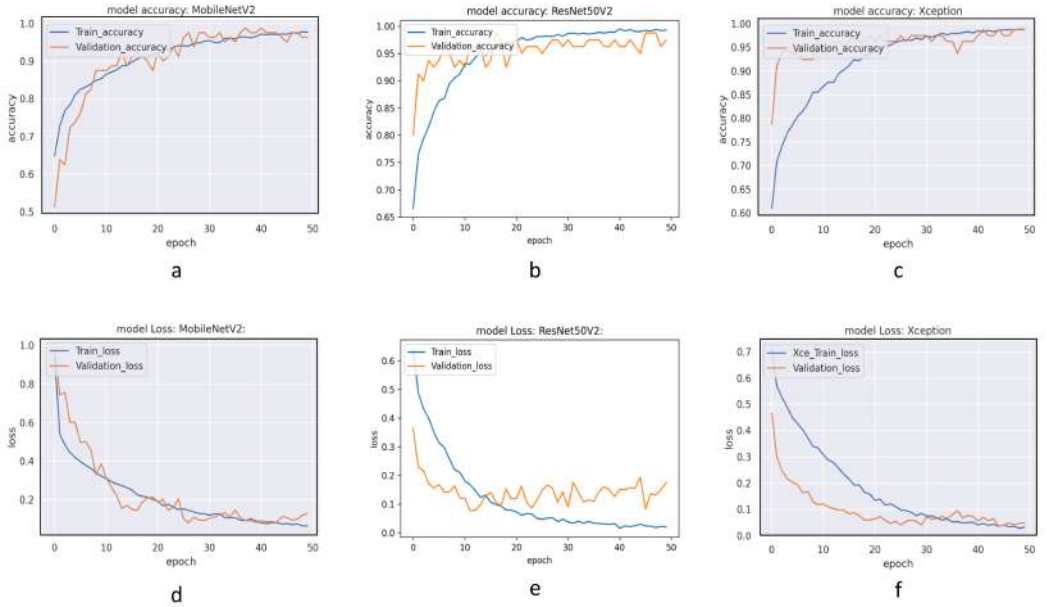

Figure 7. Training and validation accuracy plot for (**a**) MobileNetV2, (**b**) ResNet50V2, and (**c**) Xception and Training and validation loss graph for (**d**) MobileNetV2, (**e**) ResNet50V2, and (**f**) Xception for the pre-processing-only approach.

Table 7. Performance of Deep CNN models for pre-processing-only approach.

	Train					Test				
DCNN	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1-Score**
Xception	0.995	1.000	0.995	0.995	0.995	0.979	0.990	0.979	0.979	0.978571
ResNet50V2	0.997	1.000	0.997	0.997	0.997	0.975	0.987	0.975	0.975	0.975
MobileNetV2	0.990	0.999	0.990	0.990	0.990	0.943	0.983	0.943	0.943	0.942857

After applying augmentation to a pre-processed picture dataset, the performance of models is outlined in Table 8. The ResNet50V2 achieves the highest training accuracy, precision, recall, and F1-Score, while the AUC is reported to be 100%. The Xception demonstrates the highest testing accuracy, 98.9%, with a 99.9% AUC. The assessment matrices for this approach yielded the highest values among these three data-centric approaches. Figure 8 depicts the accuracy and loss performance of the training and validation sets. For the Xception model, the growth of the training and validation graphs is highly similar, with a minor variation, and the curves are a perfect match. The ResNet50V2 model demonstrates superior performance when comparing training and validation loss while maintaining minimal overfitting. Overall, this experimental setup performs the best.

Table 8. Performance of Deep CNN models for augmentation with pre-processing approaches.

		Train					Test				
Augmentation	**DCNN**	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1-Score**
Flip	Xception	0.996	1.000	0.996	0.996	0.996	0.979	0.998	0.979	0.979	0.979
	ResNet50V2	0.997	1.000	0.997	0.997	0.997	0.971	0.991	0.971	0.971	0.971
	MobileNetV2	0.993	1.000	0.993	0.993	0.993	0.907	0.972	0.907	0.907	0.907
Noise	Xception	0.990	0.999	0.990	0.990	0.990	0.946	0.983	0.946	0.946	0.946
	ResNet50V2	0.987	0.999	0.987	0.987	0.987	0.900	0.957	0.900	0.900	0.900
	MobileNetV2	0.968	0.995	0.968	0.968	0.968	0.893	0.949	0.893	0.893	0.893
All	Xception	0.997	1.000	0.997	0.997	0.997	**0.989**	**0.999**	**0.989**	**0.989**	**0.989**
	ResNet50V2	**0.998**	**1.000**	**0.998**	**0.998**	**0.998**	0.975	0.999	0.975	0.975	0.975
	MobileNetV2	0.990	0.999	0.990	0.990	0.990	0.971	0.996	0.971	0.971	0.971

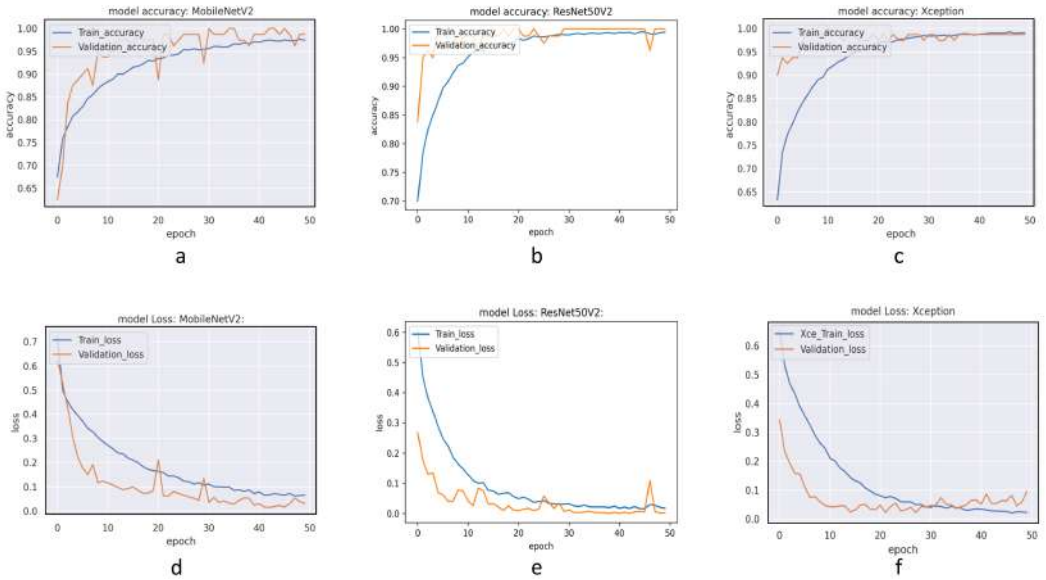

Figure 8. Training and validation accuracy plot for (**a**) MobileNetV2, (**b**) ResNet50V2, and (**c**) Xception and training and validation loss graph for (**d**) MobileNetV2, (**e**) ResNet50V2, and (**f**) Xception for the pre-processing with augmentation approach and mixing All training set.

AUC is defined as the area under the curve, with a greater AUC indicating a greater likelihood of accurate prediction. Figure 9 depicts the ROC curve of the best data-centric approach, whereas Figure 9a depicts the curve for the flip augmentation approach without pre-processing, Figure 9b illustrates the AUC of three different models for the pre-processing-only approach, and Figure 9c depicts the AUC performance of the pre-processing with augmentation approach while mixing the All training set. The Xception model performs the best in terms of accuracy and AUC across all three approaches, indicating that the prediction rate for diverse test samples in the real-world scenario is greater. Figure 10 depicts the confusion matrix for the 280 test samples, where blue boxes represent accurate predictions

of the autistic or non-autistic classes, and white boxes represent incorrect predictions, i.e., individuals who were incorrectly identified as autistic or non-autistic despite belonging to the opposite class. In Figure 10a, the first row displays the confusion matrices for the augmentation-only approaches of the ResNet50V2, MobileNetV2, and Xception models for flip augmentation, from left to right. Here, the performance of the models is abysmal, as Xception has a total of 21 misclassified test samples, which is the lowest of the models.

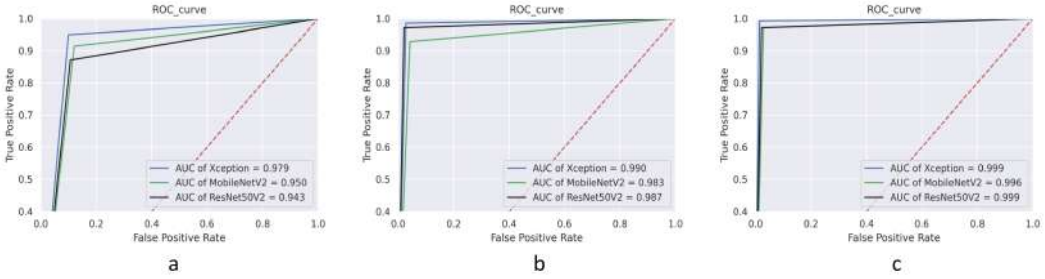

Figure 9. ROC curve of the best data-centric approach. (**a**) Flip augmentation approach without pre-processing. (**b**) Pre-processing-only approach. (**c**) Pre-processing with augmentation approach and mixing All training sets.

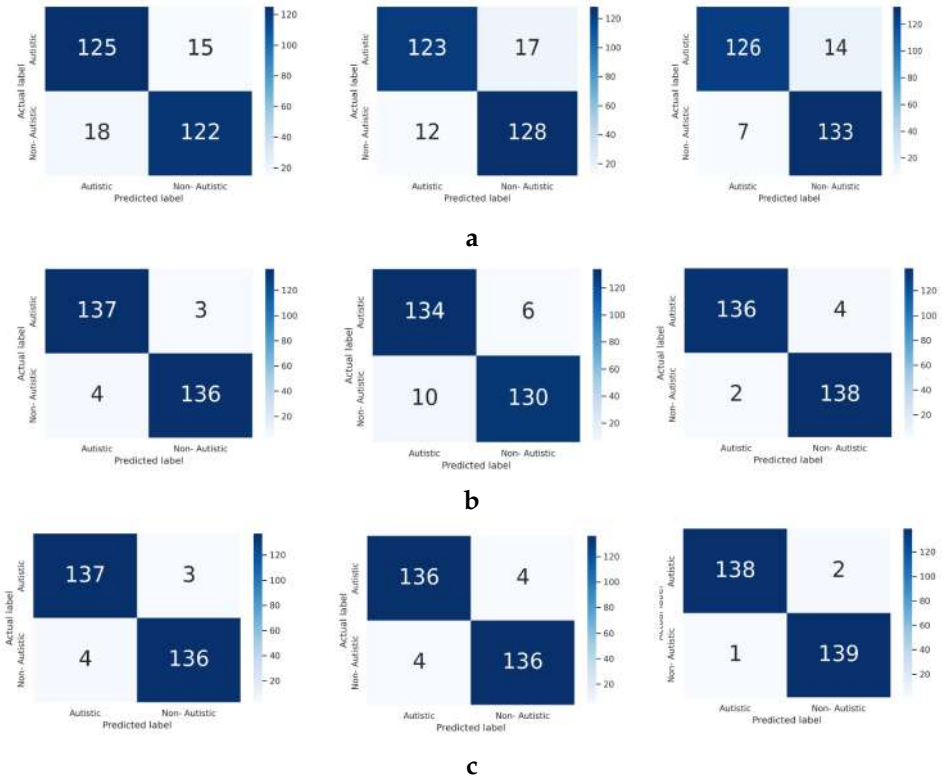

Figure 10. Confusion matrix of (**a**) ResNet50V2, MobileNetV2, and Xception algorithm, respectively, for flip augmentation approach without pre-processing (**b**) ResNet50V2, MobileNetV2, and Xception algorithm, respectively, for pre-processing-only approach and (**c**) ResNet50V2, MobileNetV2, and Xception algorithm, respectively, for pre-processing with augmentation approach and mixing All training set.

The confusion matrices for the pre-processing-only approaches of the ResNet50V2, MobileNetV2, and Xception models are displayed from left to right in the second row in Figure 10b. Here, the performance of the models is significantly enhanced by the pre-processing of the training dataset, and the Xception model has the fewest misclassified test samples, with only six samples.

The confusion matrices of the ResNet50V2, MobileNetV2, and Xception models are depicted from left to right in the last row as stated in Figure 10c for the augmentation with pre-processing approach while mixing the entire training set. Applying the augmentation after pre-processing on the training dataset results in the greatest model performance, and the Xception model has the maximum prediction accuracy with only three misclassifications. Table 9 depicts the total number of incorrectly predicted samples throughout training and testing using various data-centric approaches. The total number of misclassifications is the sum of false positive and false negative classifications. MTr is the number of incorrectly predicted samples during CNN model training. MTs are the number of incorrectly predicted samples while evaluating the performance of a model using the unique test set. As the number of test samples is the same in all scenarios, the Xception model achieves the best results for all data-centric approaches, with only three mispredictions, when the training set is pre-processed, augmented (both flipped and noise added), and then completely synthesized. Table 10 details the training duration for three distinct DL models. Te denotes the time required for model training per epoch in seconds. While training sizes vary amongst approaches, a new parameter, Tes, represents the training time per epoch per sample that the model requires. This parameter hints at which model is time-efficient for a certain data-centric approach. The training times are highest for Xception's unprocessed dataset, at 18.23 milliseconds, and lowest for ResNet50V2 after pre-processing and augmentation, at 13.46 milliseconds. Xception's training time is longer due to its complicated structure and huge number of layers, as well as its accuracy being the highest of any approach.

Table 9. Statistics of incorrectly predicted training and test samples for each data-centric approach. MTr = missed prediction of the train set, MTs = missed prediction of the test set.

DNN	Augmentation Only						Pre-Processing Only		Augmentation with Pre-Processing					
	Flip		Noise		All				Flip		Noise		All	
	MTr	MTs	MTr	MTs	MTr	MTs	MTr	MTs	MTr	MTs	MTr	MTs	MTr	MTs
Xception	8	**21**	22	23	23	27	24	**6**	20	6	52	15	28	**3**
ResNet50V2	**7**	33	11	28	9	28	**15**	7	**14**	8	67	28	18	7
MobileNetV2	21	29	95	39	50	24	52	16	38	26	166	30	99	8

Table 10. Comparative training durations for three deep learning models. Te = time required per epoch in seconds, Tes = time required per epoch per sample in ms.

Net	Augmentation Only						Pre-Processing Only		Augmentation with Pre-Processing					
	Flip		Noise		All				Flip		Noise		All	
	Te	Tes	Te	Tes	Te	Tes	Te	Tes	Te	Tes	Te	Tes	Te	Tes
Xception	95.88	18.06	96.42	18.17	145.16	18.23	88.80	16.73	89.62	16.88	88.68	16.71	170.84	16.09
ResNet50V2	83.32	15.70	83.56	15.74	124.20	15.60	75.24	14.17	75.56	14.24	76.84	14.48	142.88	13.46
MobileNetV2	82.10	15.47	83.64	15.76	124.90	15.69	74.84	14.10	75.78	14.28	75.84	14.29	144.74	13.63

Table 11 displays the accuracy and loss results for the validation dataset. The graphical representation for each epoch is shown in Figures 6–8 for the three different approaches for the best accuracy performance. ResNet50V2 demonstrates the best result for the validation set with 100% validation accuracy and nearly no validation error. It demonstrates the quality of the model's training and learning for feature extraction.

Table 11. Accuracy and loss performance on the validation set. Vac= Validation accuracy, Vlo = validation loss.

Net	Augmentation Only						Pre-Processing Only		Augmentation with Pre-Processing					
	Flip		Noise		All				Flip		Noise		All	
	Vac	Vlo	Vac	Vlo	Vac	Vlo	Vac	Vlo	Vac	Vlo	Vac	Vlo	Vac	Vlo
Xception	0.96	0.17	0.90	0.30	0.91	0.22	0.99	0.05	0.96	0.08	0.99	0.08	0.99	0.09
ResNet50V2	0.93	0.21	0.94	0.24	0.95	0.19	0.98	0.17	1.00	0.01	0.95	0.27	1.00	0.00
MobileNetV2	0.93	0.18	0.94	0.21	0.98	0.04	0.96	0.13	0.95	0.10	0.94	0.26	0.99	0.03

Explainable AI

Explainable AI refers to the ability of an AI system to explain its reasoning and decision-making processes in a way that is understandable to humans. Transfer learning, on the other hand, refers to the ability of an AI system to transfer knowledge learned from one task to another related task. These two concepts are related in that explainability can enhance the effectiveness of transfer learning by providing insights into the decision-making process of the AI system.

One method of explaining the decision-making process of a neural network is through visualization techniques, such as Grad-CAM. It highlights the most important regions of an input image that contribute to the neural network's prediction. Thus, ASD classification based on facial images can be used to explain where the transfer learning models concentrate on extracting the ASD-specific features. By visualizing the acquired knowledge, we can comprehend how and where to focus on facial images, which can aid in debugging transfer learning models and discovering transferability restrictions.

Figure 11 depicts the facial feature region of autistic and non-autistic individuals. These two samples were selected at random from 280 test samples. Grad-CAM shows where the various models acquire the characteristics that characterize them as ASD or normal control children. The Grad-CAM heatmaps were built using the model weights from the data-centric models with the highest performance. With Xception, the primary focus area is between the eyes, whereas ResNet50V2 focuses mostly on the nose and lips for autistic children. For a normal child, the Xception and ResNet50V2 models focus mostly on the nose, the area below the nose, the area between the eyes, and the upper and lower lips. In contrast, the MobileNetV2 model focuses primarily on the upper nose sections, such as the eyes and upper nose.

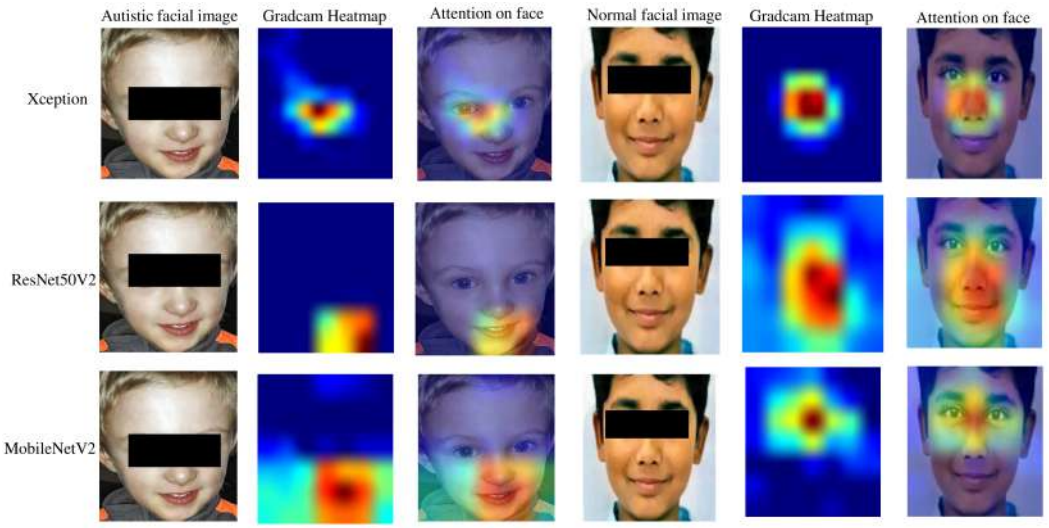

Figure 11. Grad-CAM feature map for the two random samples from the test set (autistic and normal control child).

While the test set contains numerous sorts of images, it is impossible to draw a simple conclusion about the results, as there are so many differentiating aspects, such as gender and face pose, which might lead to varied Grad-CAM outcomes. To generalize the focus areas for autistic and non-autistic children, we require a generalized image pattern, i.e., a mean image. The mean image is a critical component of the pre-processing pipeline in image classification tasks, which helps to normalize the data. It is composed of the average value of every pixel in a collection of photos. All the photos in the test set are either autistic or non-autistic. Each sort of image is distinguished by gender, as the demographics of male and female faces should differ. Moreover, images can be divided into two categories based on the position of face landmarks, namely the frontal pose and the side pose. In the frontal pose, the face is at a zero-degree angle with the surface, and the picture is therefore staring directly at the viewer. It is assumed to be the perfect candidate for testing, as the facial feature may be easily predicted from the frontal view. There are photos in the test set that are not precisely straight and whose facial alignment is skewed to the left or right but not the front. This is referred to as the side pose.

Additional factors may be responsible for distinguishing features and the distinct focus area in the Grad-CAM output. Instead, we separated the images of the test set into autistic and non-autistic groups. The photos were then divided in two based on gender for autistic and non-autistic groups. For both autistic and non-autistic samples, the photos were grouped afterward according to frontal and side face poses. Table 12 describes the grouping details for deducting the mean image. Hence, based on the above table, we obtained five mean images for the autistic sample, representing the average of all autistic samples, male and female samples, and frontal and side pose samples. Similarly, for the non-autistic group, we could deduce the same images. Subsequently, the Grad-CAM was applied to these images to locate the normalized region of interest for these various deep neural network models in order to extract autistic or non-autistic characteristics.

Table 12. The number of test samples.

Criteria	Number of Test Samples				Total
	Autistic = 140		Non-Autistic = 140		280
Gender	Male = 112	Female = 28	Male = 88	Female = 52	280
Facial orientation	Frontal Posture = 108	Side posture = 32	Frontal Posture = 103	Side posture = 37	280

Using the Grad-CAM heatmap, Figure 12 depicts the focal regions of the autistic facial images for the three CNN models. The first column contains the mean images from the total number of images of autistic children as seen in Table 11. These photos contain the facial characteristics of autism for various scenarios, such as gender and pose. So, we may conclude that the heatmap derived from the trained model with Grad-CAM on these mean images indicates the region of interest or features to be discovered in the autistic samples. Instead of selecting a random sample of male autistic children, it is justified to examine the heatmap on the mean image of all male autistic children in the test dataset. In the first row of Figure 12, the common areas where the different models focus on male autistic samples are depicted. These areas include the nose, the middle of the eye, and the upper lips for the Xception model, a portion of the nose primarily on the lips for ResNet50V2, and the forehead for MobileNetV2. Even for other types of mean images, the Xception focuses mostly on the nose, the center of the eyes, and the upper lips.

For the frontal stance, which is regarded as the optimal condition for this type of binary classification, Xception focuses mostly on the nose, the center of the eyes, and the upper lips, whereas ResNet50V2 focuses primarily on the nose and lips, and MobileNetV2 focuses on the center of the forehead. The final row of Figure 12 is the mean of all autistic samples in the test set, and the general focus of autistic children is identical to that of the preceding mean images.

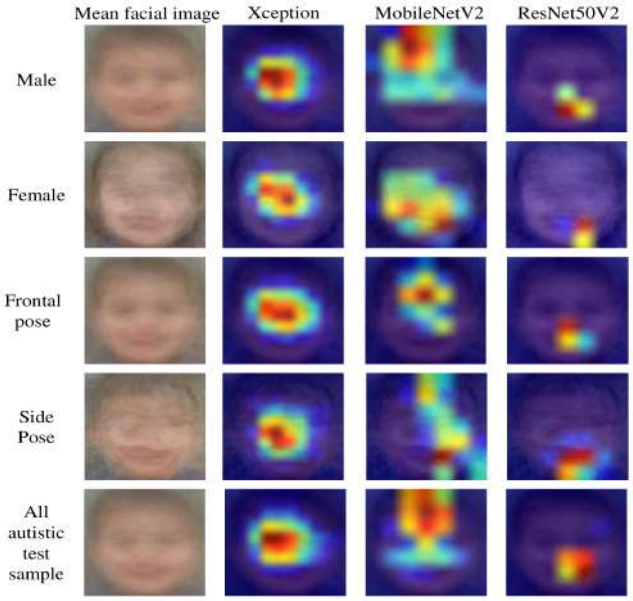

Figure 12. Grad-CAM results of the mean images created from various groups from the test set for autistic samples.

Although the second row of Figure 13 primarily shows the mean of three models, it can be regarded as the general compiled area of focus for recognizing autistic children. The mean image of the three models' Grad-CAM heatmaps illustrates all the regions where the models concentrate while extracting autistic traits. A "T"-shaped region consisting of the forehead, the center of the eyes, the nose, and the lips are the frequent regions of focus when attempting to forecast the autistic sample irrespective of different genders and postures. Occasionally, models may also concentrate on the cheeks.

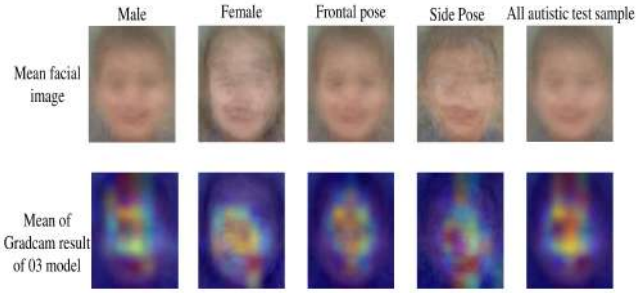

Figure 13. Overall Grad-CAM focus areas of a particular mean image summarizing the Grad-CAM outcome for three CNN models for autistic samples.

Figure 14 displays the mean images of non-autistic samples derived using the same methods as for the preceding autistic images. The center of the eyes, the nose, and the upper lip are the principal focal points for all non-autistic samples of the Xception model, which remain the same as those identified previously for autistic children. With the other two models, ResNet50V2 and MobileNetV2, non-autistic characteristics are identified from the peripheral area of the face rather than the center.

Hence, the overall performance and feature isolation on the face for Xception is rather consistent for almost all the cases, regardless of gender, face pose, or autistic and non-autistic samples. The Grad-CAM results conclude that the Xception model selected the same area for feature extraction and that the classification is based on nearly identical facial regions. For the other two models, however, the autistic traits are nearly identical and isolated. In contrast, the non-autistic features originate from many regions of the face, as indicated by the decentralized Grad-CAM heatmap for the mean photos of non-autistic samples.

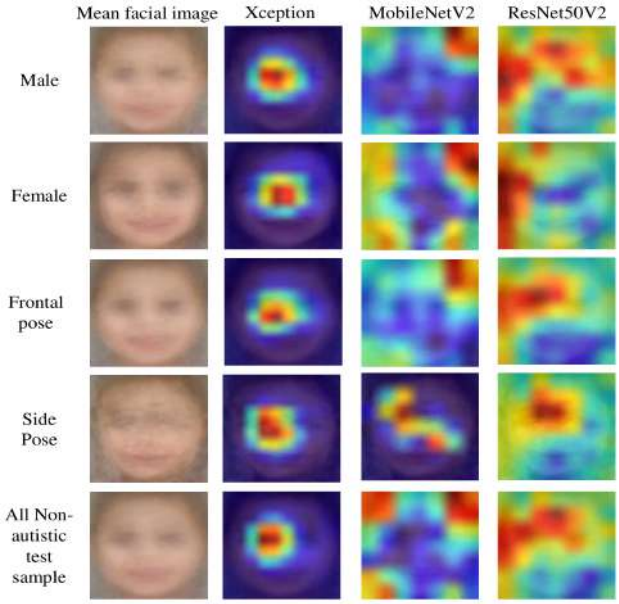

Figure 14. Grad-CAM results of the mean images created from various groups from the test set for non-autistic samples.

Figure 15 shows the Grad-CAM results or focal areas of the face for the incorrectly predicted autistic samples for the pre-processing with augmentation approach, where we obtained the most accuracy in prediction. The fundamental cause is that the models focus on the incorrect areas. Uncertain as to why the models failed to concentrate on the areas required to extract facial features, we can assume that these images are in a side posture or extreme facial expression. The fact that there is no repletion in the failed images is intriguing since it indicates that various models fail to forecast distinct test samples.

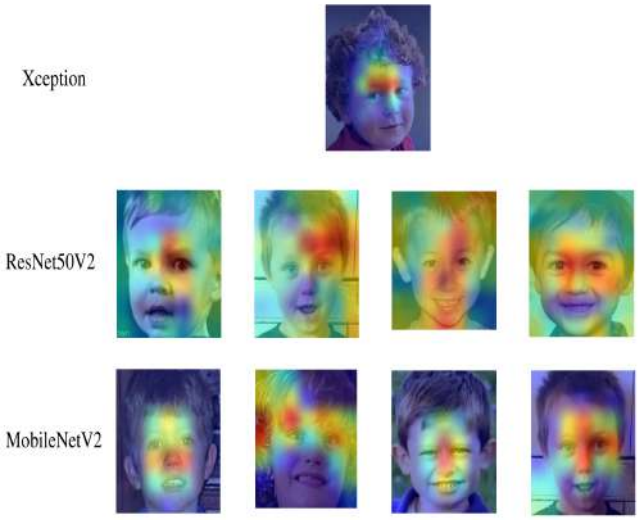

Figure 15. Grad-CAM results of the failed images different models for autistic samples for the best results (pre-processing with augmentation approach mixing all dataset).

4. Discussion

This research discusses the use of DL and explainable artificial intelligence (AI) to diagnose ASD using facial images. The paper highlights the importance of the early diagnosis of ASD and focuses mainly on the use of AI in this emerging medical application. The paper proposes a data-centric approach that involves pre-processing and synthesizing a large dataset of facial images of children with and without ASD. We then train some DL models using the dataset to accurately diagnose ASD from facial images using different pre-

processing and augmentation techniques. In addition to providing insights into the model's decision-making process and the components that contribute to the diagnosis, explainable AI techniques are also applied. Finally, we discuss the efficiency of this approach and compare it to other state-of-the-art methods in order to show that it beats other approaches in terms of accuracy and efficiency.

For data pre-processing, we adopt two important steps—alignment and cropping. Alignment is the process of adjusting the image orientation so that the object of interest is in a consistent position across all images in the dataset. Cropping is another process of removing unwanted parts of the image, such as the background or other objects that are not relevant to the task at hand. In addition to improving model accuracy, alignment, and cropping can also help to reduce the computational complexity of CNN models. By removing unwanted parts of the image, cropping reduces the input size of the CNN model, which can significantly reduce the number of parameters and computation required. Several studies have shown that these pre-processing steps can significantly improve the accuracy of CNN models trained on image datasets. A study by Junliang Xing et al. [60] showed that alignment improved the accuracy of CNN models for the face recognition dataset by up to 6%. Similarly, a study by Ruoning Song et al. [61] showed that cropping improved the accuracy of CNN models on an object recognition dataset by up to 1%.

There are quite a few studies that have been undertaken exclusively in this area. To our knowledge, this data-centric approach has never been tried for ASD diagnosis using facial image datasets to achieve higher accuracy. Previous research showed that poor image quality in the training dataset substantially contributes to inaccurate model outcomes. Pictures of children's faces often suffer from noise, poor resolution, misalignment, and other issues. Rather, more researchers likewise concentrate on optimizing the models or set of hyperparameters with no promising improvement in accuracy. The results of the most recent studies in this area are compared in Table 13.

Table 13. Performance comparison of our proposed model with the existing related studies.

Ref.	CNN Model	Sample Size	Accuracy	Precision	Recall	F1-Score
[15]	MobileNet	3014	94.64	-	-	-
[16]	MobileNet	3014	95	94	97	95
[17]	AutoML	2936	96.6	96.2	96	96
[18]	ASDDTLA	2940	87.7	87.6	88	87
[12]	Xception	3014	95	95	95	95
[20]	VGG16	2940	70	-	-	-
[19]	Densenet	2897	98	98.1	97.7	97.9
Our Proposed data-centric approach						
Augmentation-only approach	Xception	5668	92.5	92.5	92.5	92.5
Pre-processing only	Xception	5668	97.9	97.9	97.9	97.9
Augmentation with Pre- processing approach	Xception	10,976	98.9	98.9	98.9	98.9

Mohammad-Parsa et al. [15] and Zeyad A. T. Ahmed et al. [16] both applied the same strategy utilizing the MobileNet model, obtaining the highest accuracy of 95%. Although M. S. Alam et al. [12] conducted an exhaustive ablation study to determine the optimal models and hyperparameters, they were only able to achieve an accuracy of 95% at best. In a later study, Basma R. G. Elshoky et al. [17] employed the automated tool Hyperpot with tree-based pipeline optimization to attain a prediction accuracy of 96.6%. The 98% success rate claimed by Mohamed Ikermane et al. [19] is not backed up by the data. The comparison with two other studies, by Taher M. Ghazal et al. [18] and Narinder Kaur et al. [20], whose claimed accuracy was only 87.7% and 70%, respectively, is not so significant in this regard.

Compared to the previous research, the augmentation-only approach has a prediction accuracy of 92.5% with the Xception algorithm. Subsequently, after the training dataset was pre-processed, this model's performance increased to 97.9% with the same CNN model, which is a substantial improvement in this regard. When both pre-processing and augmentation are applied to the training dataset, we obtain a prediction accuracy of 98.9%, which clearly outperforms all prior ASD diagnosis research results.

The implementation of Grad-CAM, an artificial intelligence (AI) tool that exposes the diagnostic outcomes of transfer learning models carries substantial clinical ramifications for the domain of medical diagnosis, specifically within the realm of assessing autism spectrum disorder (ASD). This explainable AI enables healthcare practitioners to enhance their ability to make informed and precise evaluations, improving patient care and facilitating well-informed treatment decisions. Lastly, we highlight the importance of carefully observing distinct facial characteristics, including the forehead, area between the eyes, nostrils, lips, and occasionally the cheeks, in children diagnosed with autism spectrum disorder (ASD) as well as normal control individuals. The identification of reliable and readily observable facial markers linked with autism spectrum disorder (ASD) can contribute to the early detection of the illness, facilitating prompt interventions and care for children affected by it. If these non-intrusive visual cues are confirmed and integrated into clinical practice, they have the potential to function as an extra screening tool that complements current diagnostic approaches. This, in turn, can have substantial advantages for individuals affected by ASD and their families, as it enables prompt access to suitable interventions and support services.

Limitation of the Study

During our research, we encountered a number of limitations that can be addressed in future studies as follows:

- Firstly, there are some potential drawbacks to dataset pre-processing, as alignment and cropping can introduce some loss of information as parts of the image are removed or altered. Training time is greatly decreased after pre-processing; however, processing big datasets with high-resolution photos can be computationally expensive. Hence, we can automatically highlight specific facial areas utilizing improved nets rather than eliminating portions of the image.
- Secondly, this Kaggle ASD dataset is the only openly accessible dataset in this regard on the internet and is not backed by clinical evidence. Moreover, the dataset consists only of RGB modality, not 3D (depth or shape) facial images. The lack of supporting data, such as gender, age, nationality, and sibling information, for each sample makes it impossible to validate the results demographically.
- Third, this Kaggle ASD dataset is not distributed symmetrically regarding gender, facial postures, or emotions. Additionally, the data were not collected using a certain protocol or attention mechanism. Hence, when analyzing the explainability of CNN models, it is quite difficult to develop a normalized pattern or specific facial regions to focus on. We hope that the medical research institute will publish or share a comprehensive dataset that can answer all of these issues.

5. Conclusions

The primary objective of this study is to explore the various data-centric approaches for diagnosing ASD using deep CNN models and to maximize the accuracy of prediction. We utilize the Kaggle ASD dataset. Rather than focusing on model and hyperparameter tuning, we apply several pre-processing and augmentation techniques to the training set to determine the most effective method for ASD diagnosis. After pre-processing and combining two augmentation approaches—flipping and adding noise—the best performance parameters were obtained with 98.9% accuracy, precision, recall, and F1-score and 99.9% AUC while evaluating the trained models with a fixed dataset. We use the pre-processing and synthesis technique for the training dataset to overcome the limitations of earlier

research in this area. We present Grad-CAM, an AI approach that reveals the test results of transfer learning models. We prefer to observe, rather than conclude, that the forehead, area between the eyes, nostrils, lips, and rarely the cheeks are diagnostic of autism spectrum disorder (ASD) or normal control children. This technique may serve as a beneficial tool for the early detection and diagnosis of ASD, resulting in improved clinical outcomes for affected individuals if these facial regions are validated as ASD biomarkers that require future investigations. The acquisition of a comprehensive clinical dataset containing diverse modalities and detailed demographic information and then incorporating advanced techniques, such as active learning, attention learning, and vision transformers, has significant potential to drive future advancements in this domain.

Author Contributions: M.S.A. (Mohammad Shafiul Alam): Conceptualization, Methodology, Software, Writing—Original draft preparation. M.M.R.: Data curation, Software, Writing—reviewing and editing. A.R.F.: Data curation, Writing—reviewing and editing. H.F.M.Z.: Data curation, Writing—reviewing and editing. T.E.A.: Formal analysis, Writing—reviewing and editing. M.S.A. (Md Shahin Ali): Data curation, Formal analysis, Writing—reviewing and editing. K.D.G.: Formal analysis, Writing—reviewing and editing. M.M.A.: Formal analysis, Writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The author would like to express the deepest gratitude to the International Islamic University Malaysia for their support through the Tuition Fee Waiver Scheme 2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al Banna, M.H.; Ghosh, T.; Taher, K.A.; Kaiser, M.S.; Mahmud, M. A monitoring system for patients of autism spectrum disorder using artificial intelligence. In Proceedings of the Brain Informatics: 13th International Conference, BI 2020, Padua, Italy, 19 September 2020; Proceedings 13; Springer: Cham, Switzerland, 2020; pp. 251–262.
2. Habayeb, S.; Kenworthy, L.; De La Torre, A.; Ratto, A. Still left behind: Fewer black school-aged youth receive ASD diagnoses compared to white youth. *J. Autism Dev. Disord.* **2022**, *52*, 2274–2283. [CrossRef] [PubMed]
3. Sheldrick, R.C.; Maye, M.P.; Carter, A.S. Age at first identification of autism spectrum disorder: An analysis of two US surveys. *J. Am. Acad. Child Adolesc. Psychiatry* **2017**, *56*, 313–320. [CrossRef]
4. Perinelli, M.G.; Cloherty, M. Identification of autism in cognitively able adults with epilepsy: A narrative review and discussion of available screening and diagnostic tools. *Seizure* **2023**, *104*, 6–11. [CrossRef]
5. Ahsan, M.M.; Luna, S.A.; Siddique, Z. Machine-learning-based disease diagnosis: A comprehensive review. *Healthcare* **2022**, *10*, 541. [CrossRef] [PubMed]
6. Khodatars, M.; Shoeibi, A.; Sadeghi, D.; Ghaasemi, N.; Jafari, M.; Moridian, P.; Khadem, A.; Alizadehsani, R.; Zare, A.; Kong, Y.; et al. Deep learning for neuroimaging-based diagnosis and rehabilitation of autism spectrum disorder: A review. *Comput. Biol. Med.* **2021**, *139*, 104949. [CrossRef]
7. Shoeibi, A.; Khodatars, M.; Jafari, M.; Ghassemi, N.; Moridian, P.; Alizadesani, R.; Ling, S.H.; Khosravi, A.; Alinejad-Rokny, H.; Lam, H.; et al. Diagnosis of brain diseases in fusion of neuroimaging modalities using deep learning: A review. *Inf. Fusion* **2022**, *93*, 85–117.
8. Sadek, E.T.; Seada, N.A.; Ghoniemy, S. Neural Network-Based Method for Early Diagnosis of Autism Spectral Disorder Head-Banging Behavior from Recorded Videos. *Int. J. Pattern Recognit. Artif. Intell.* **2023**, *37*, 2356003. [CrossRef]
9. Elbattah, M.; Guérin, J.L.; Carette, R.; Cilia, F.; Dequen, G. Vision-based Approach for Autism Diagnosis using Transfer Learning and Eye-tracking. In Proceedings of the HEALTHINF, Online, 9–11 February 2022; pp. 256–263.
10. Lee, J.H.; Lee, G.W.; Bong, G.; Yoo, H.J.; Kim, H.K. Deep-learning-based detection of infants with autism spectrum disorder using auto-encoder feature representation. *Sensors* **2020**, *20*, 6762. [CrossRef]
11. Hendr, A.; Ozgunalp, U.; Erbilek Kaya, M. Diagnosis of Autism Spectrum Disorder Using Convolutional Neural Networks. *Electronics* **2023**, *12*, 612. [CrossRef]
12. Alam, M.S.; Rashid, M.M.; Roy, R.; Faizabadi, A.R.; Gupta, K.D.; Ahsan, M.M. Empirical study of autism spectrum disorder diagnosis using facial images by improved transfer learning approach. *Bioengineering* **2022**, *9*, 710. [CrossRef]

13. Ghosh, T.; Al Banna, M.H.; Rahman, M.S.; Kaiser, M.S.; Mahmud, M.; Hosen, A.S.; Cho, G.H. Artificial intelligence and internet of things in screening and management of autism spectrum disorder. *Sustain. Cities Soc.* **2021**, *74*, 103189. [CrossRef]
14. Ghosh, T.; Al Banna, M.H.; Al Nahian, M.J.; Taher, K.A.; Kaiser, M.S.; Mahmud, M. A hybrid deep learning model to predict the impact of COVID-19 on mental health form social media big data. *IEEE Access* **2021**, *11*, 77009–77022.
15. Hosseini, M.P.; Beary, M.; Hadsell, A.; Messersmith, R.; Soltanian-Zadeh, H. Deep learning for autism diagnosis and facial analysis in children. *Front. Comput. Neurosci.* **2022**, *15*, 789998. [CrossRef] [PubMed]
16. Ahmed, Z.A.; Aldhyani, T.H.; Jadhav, M.E.; Alzahrani, M.Y.; Alzahrani, M.E.; Althobaiti, M.M.; Alassery, F.; Alshaflut, A.; Alzahrani, N.M.; Al-Madani, A.M. Facial features detection system to identify children with autism spectrum disorder: Deep learning models. *Comput. Math. Methods Med.* **2022**, *2022*, 3941049. [CrossRef] [PubMed]
17. Elshoky, B.R.G.; Younis, E.M.; Ali, A.A.; Ibrahim, O.A.S. Comparing automated and non-automated machine learning for autism spectrum disorders classification using facial images. *ETRI J.* **2022**, *44*, 613–623. [CrossRef]
18. Ghazal, T.M.; Munir, S.; Abbas, S.; Athar, A.; Alrababah, H.; Khan, M.A. Early Detection of Autism in Children Using Transfer Learning. *Intell. Autom. Soft Comput.* **2023**, *36*, 11–22. [CrossRef]
19. Ikermane1, M.; Mouatasim, A.E. Web-based autism screening using facial images and convolutional neural network. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *29*, 1140–1147. [CrossRef]
20. Kaur, N.; Gupta, G. Refurbished and improvised model using convolution network for autism disorder detection in facial images. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *29*, 883–889. [CrossRef]
21. Kaggle. Autism—Google Drive. Available online: https://drive.google.com/drive/folders/1XQU0pluL0m3TIlXqntano12d68peMb8A (accessed on 7 March 2023).
22. Kaggle. Kaggle-Autism: Detecting Autism Spectrum Disorder in Children with Computer Vision—Adapting Facial Recognition Models to Detect Autism Spectrum Disorder. 2020. Available online: https://github.com/mm909/Kaggle-Autism (accessed on 7 March 2023).
23. Talib, M.A.; Majzoub, S.; Nasir, Q.; Jamal, D. A systematic literature review on hardware implementation of artificial intelligence algorithms. *J. Supercomput.* **2021**, *77*, 1897–1938. [CrossRef]
24. Nandy, A.; Duan, C.; Kulik, H.J. Audacity of huge: Overcoming challenges of data scarcity and data quality for machine learning in computational materials discovery. *Curr. Opin. Chem. Eng.* **2022**, *36*, 100778. [CrossRef]
25. Azam, M.A.; Khan, K.B.; Salahuddin, S.; Rehman, E.; Khan, S.A.; Khan, M.A.; Kadry, S.; Gandomi, A.H. A review on multimodal medical image fusion: Compendious analysis of medical modalities, multimodal databases, fusion techniques and quality metrics. *Comput. Biol. Med.* **2022**, *144*, 105253. [CrossRef]
26. Huynh, T.; Nibali, A.; He, Z. Semi-supervised learning for medical image classification using imbalanced training data. *Comput. Methods Programs Biomed.* **2022**, *216*, 106628. [CrossRef] [PubMed]
27. Varoquaux, G.; Cheplygina, V. Machine learning for medical imaging: Methodological failures and recommendations for the future. *NPJ Digit. Med.* **2022**, *5*, 48. [CrossRef] [PubMed]
28. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
29. Sarrionandia, X.; Nieves, J.; Bravo, B.; Pastor-López, I.; Bringas, P.G. An Objective Metallographic Analysis Approach Based on Advanced Image Processing Techniques. *J. Manuf. Mater. Process.* **2023**, *7*, 17. [CrossRef]
30. Dong, H.; Zhu, B.; Zhang, X.; Kong, X. Use data augmentation for a deep learning classification model with chest X-ray clinical imaging featuring coal workers' pneumoconiosis. *BMC Pulm. Med.* **2022**, *22*, 271. [CrossRef] [PubMed]
31. Oyelade, O.N.; Ezugwu, A.E.; Almutairi, M.S.; Saha, A.K.; Abualigah, L.; Chiroma, H. A generative adversarial network for synthetization of regions of interest based on digital mammograms. *Sci. Rep.* **2022**, *12*, 6166. [CrossRef] [PubMed]
32. Deepak, S.; Ameer, P. MSG-GAN based synthesis of brain MRI with meningioma for data augmentation. In Proceedings of the 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2–4 July 2020; pp. 1–6.
33. Ju, L.; Wang, X.; Zhao, X.; Bonnington, P.; Drummond, T.; Ge, Z. Leveraging regular fundus images for training UWF fundus diagnosis models via adversarial learning and pseudo-labeling. *IEEE Trans. Med. Imaging* **2021**, *40*, 2911–2925. [CrossRef]
34. Srivastav, D.; Bajpai, A.; Srivastava, P. Improved classification for pneumonia detection using transfer learning with gan based synthetic image augmentation. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 433–437.
35. Elbattah, M.; Loughnane, C.; Guérin, J.L.; Carette, R.; Cilia, F.; Dequen, G. Variational autoencoder for image-based augmentation of eye-tracking data. *J. Imaging* **2021**, *7*, 83. [CrossRef]
36. Ali, M.S.; Islam, M.K.; Haque, J.; Das, A.A.; Duranta, D.; Islam, M.A. Alzheimer's disease detection using m-random forest algorithm with optimum features extraction. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 1–6.
37. Ali, M.S.; Islam, M.K.; Das, A.A.; Duranta, D.; Haque, M.; Rahman, M.H. A novel approach for best parameters selection and feature engineering to analyze and detect diabetes: Machine learning insights. *BioMed Res. Int.* **2023**, *2023*, 8583210. [CrossRef]
38. Xiang, J.; Zhu, G. Joint face detection and facial expression recognition with MTCNN. In Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, Chin, 21–23 July 2017; pp. 424–427.

39. Kumar, R. Analysis of shape alignment using Euclidean and Manhattan distance metrics. In Proceedings of the 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Bhopal, India, 27–29 October 2017; pp. 326–331.
40. Valkov, V. Face Detection on Custom Dataset with Detectron2 and PyTorch Using Python. 2020. Available online: https://towardsdatascience.com/face-detection-on-custom-dataset-with-detectron2-and-pytorch-using-python-23c17e99e162 (accessed on 7 March 2023).
41. Shijie, J.; Ping, W.; Peiyi, J.; Siping, H. Research on data augmentation for image classification based on convolution neural networks. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 4165–4170.
42. Fu, B.; Zhao, X.; Song, C.; Li, X.; Wang, X. A salt and pepper noise image denoising method based on the generative classification. *Multimed. Tools Appl.* **2019**, *78*, 12043–12053. [CrossRef]
43. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef]
44. Duranta, D.; Ali, M.S.; Das, A.A.; Rahman, M.M.; Ahsan, M.M.; Miah, M.S.; Islam, M.K. Enhancing Atrial Fibrillation detection accuracy: A wavelet transform filtered single lead ECG signal analysis with artificial neural networks and novel feature extraction. *Mach. Learn. Appl.* **2023**, *12*, 100472. [CrossRef]
45. Ali, M.S.; Miah, M.S.; Haque, J.; Rahman, M.M.; Islam, M.K. An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. *Mach. Learn. Appl.* **2021**, *5*, 100036. [CrossRef]
46. Hasan, I.; Ali, S.; Rahman, H.; Islam, K. Automated detection and characterization of colon Cancer with deep convolutional neural networks. *J. Healthc. Eng.* **2022**, *2022*, 5269913. [CrossRef]
47. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part IV 14; Springer: Cham, Switzerland, 2016; pp. 630–645.
49. Kaiser, L.; Gomez, A.N.; Chollet, F. Depthwise separable convolutions for neural machine translation. *arXiv* **2017**, arXiv:1706.03059.
50. Ahsan, M.M.; Uddin, M.R.; Ali, M.S.; Islam, M.K.; Farjana, M.; Sakib, A.N.; Al Momin, K.; Luna, S.A. Deep transfer learning approaches for Monkeypox disease diagnosis. *Expert Syst. Appl.* **2023**, *216*, 119483. [CrossRef] [PubMed]
51. Murugan, P.; Durairaj, S. Regularization and optimization strategies in deep convolutional neural network. *arXiv* **2017**, arXiv:1712.04711.
52. Anil, R.; Gupta, V.; Koren, T.; Singer, Y. Memory-efficient adaptive optimization for large-scale learning. *arXiv* **2019**, arXiv:1901.11150.
53. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
54. Shin, D. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI. *Int. J. Hum.-Comput. Stud.* **2021**, *146*, 102551. [CrossRef]
55. Ahsan, M.M.; Ali, M.S.; Hassan, M.M.; Abdullah, T.A.; Gupta, K.D.; Bagci, U.; Kaushal, C.; Soliman, N.F. Monkeypox Diagnosis with Interpretable Deep Learning. *IEEE Access* **2023**, *11*, 81965–81980. [CrossRef]
56. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Gradient-weighted Class Activation Mapping. *arXiv* **2016**, arXiv:1610.02391.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
58. Chattopadhyay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V. Grad-CAM++: Improved visual explanations for deep convolutional networks, arXiv. *arXiv* **2018**, arXiv:1710.11063.
59. Chollet, F. Keras. GitHub Repository. 2015. Available online: https://github.com/fchollet/keras (accessed on 22 April 2023).
60. Xing, J.; Niu, Z.; Huang, J.; Hu, W.; Zhou, X.; Yan, S. Towards robust and accurate multi-view and partially-occluded face alignment. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 987–1001. [CrossRef]
61. Song, R.; Zhang, L.; Zhu, C.; Liu, J.; Yang, J.; Zhang, T. Thyroid nodule ultrasound image classification through hybrid feature cropping network. *IEEE Access* **2020**, *8*, 64064–64074. [CrossRef]