ID N. 28

UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

DEPARTMENT OF ENGINEERING

NATIONAL RESEARCH COUNCIL OF ITALY (CNR)

INSTITUTE OF INFORMATION SCIENCE AND TECHNOLOGIES (ISTI)

Italian National Ph.D. in Artificial Intelligence
Health and Life Sciences
XXXVII Cycle

Exploring Machine Learning for Image Enhancement and Multimedia Understanding towards Autistic People's behavior analysis

Supervisors

Davide Moroni

Candidate

Ali Reza Omrani

January, 2025

To my family and friends.

Acknowledgements

I would like to thank my mother for her support throughout my life and for pushing me to be the best version of myself.
Moreover, I would like to express my deep gratitude to my supervisor, Dr. Davide Moroni, for his continuous guidance, support, and patience during my Ph.D. program. I could not have imagined having a better advisor and mentor for my Ph.D. journey.
I would also like to thank Prof. Marc J. Lanovaz for accepting me into his lab and helping me work in the field of autism and accomplish my final research.
Lastly, I would like to thank my friends and fellow researchers for their companionship and for making this journey an unforgettable experience.

Abstract

This dissertation investigates the application of machine learning techniques for monitoring and analyzing signs and behaviors associated with Autism Spectrum Disorder (ASD), with a particular emphasis on image and audio processing. From an Explainable Artificial Intelligence perspective, the research begins by examining the capabilities of deep learning models in the analysis of facial features of autistic and non-autistic individuals. A crucial element in this process is the enhancement of input image quality. To address this, we also propose a multi-exposure High Dynamic Range (HDR) imaging method, which improves image detail through segmentation and deep learning-based reconstruction. The proposed method can be used for various topics, one of which is for our research on distinguishing autistic from non-autistic individuals. The IEEE ICASSPW paper highlights the superior performance of empirical thresholding over Otsu thresholding; however, the integration of these methods led to overfitting, prompting the adoption of Otsu segmentation. The results of the reconstruction network, consisting of Visual Attention Module (VAM), attention and alignment modules, and refinement stages, outperformed state-of-the-art techniques, as published in IEEE Access.

Additionally, the research explores the use of Vision Transformers and ResNets to differentiate children with autism from their neurotypical peers, achieving a 92% accuracy rate. We also use explainable AI techniques to clarify the model's decision-making process, with findings submitted to the Journal of Research in Autism Spectrum Disorders. Furthermore, the study investigates vocal stereotypy measurement in autistic children using machine learning applied to audio files, resulting in tailored models for individual patients. These findings were submitted to the Journal of Applied Behavior Analysis.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Autism Spectrum Disorder (ASD) is a neurological and developmental disorder that impacts individuals' social activities, including their learning processes, behaviors, and interactions with others. The prevalence of ASD has significantly increased in recent years, with approximately 1 in 36 children being identified as autistic [44]. Machine learning methods can assist professionals in identifying individuals with autism and conducting necessary analyses.

Various machine learning methods have been applied in the field of psychology, more specifically, in ASD. For instance, extensive research has been conducted on diagnosing ASD based on brain MRI images. However, image classification can serve various purposes in the ASD field, such as classifying the emotions of individuals with autism or distinguishing between autistic and non-autistic individuals. Moreover, in addition to image processing, audio processing is also utilized during analyses, aiding professionals in observing individuals with autism more conveniently.

Additionally, within this context, analyzing the behavior of individuals with autism may contribute to enhancing their social skills. Therefore, this dissertation focuses on two distinct aspects of autism: image and audio processing.

Image processing has been a tremendous aid in various fields, including psychology and, more specifically, ASD. For instance, images have been used to analyze eye tracking or to distinguish individuals with ASD from those without, using facial or Magnetic Resonance Imaging (MRI) scans. Therefore, one of the critical aspects of this research over the past three years has been the use of facial images to identify individuals with autism.

However, as with other domains of image processing projects, possessing images with the highest detail and superior quality plays a significant role. For this purpose, we conducted research in HDR imaging to enhance the details of images. We proposed a new multi-exposure HDR imaging method using image segmentation during this research. The general idea is to first extract the areas with the highest detail from images with different exposures and

merge them to produce an initial version of the image with enhanced detail. Subsequently, we established a deep learning-based image reconstruction pipeline with several stages to create the final image with increased detail.

On this topic, we published a conference paper [57] that focused on the segmentation stage, where we established two methods: empirical thresholding and Otsu thresholding. Several experts investigated the optimal range for extracting image details for the empirical technique. Conversely, we calculated a threshold for the optimal range using the images' histogram in the Otsu segmentation technique. After comparing these two methods, the results demonstrated that the manual method outperformed the Otsu technique. Subsequently, we trained a model to segment images. The paper was published in the **2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) Workshop**.

However, when we integrated the segmentation network with the reconstruction network, it led to overfitting. As a result, we removed the segmentation stage. Upon further evaluation, we identified that the problem with empirical thresholding is that each scene has wide ranges, and thresholding with a static range can cause distortions in some cases. Therefore, we employed Otsu segmentation for the segmentation stage and fed the segmented masks and the input images into the reconstruction network.

The reconstruction network comprises several stages, including the VAM, attention and alignment modules, reconstruction, and refinement. In brief, the input images and their masks were fed into the VAM module to extract areas of images with more detail. Simultaneously, the input images were fed into the attention and alignment modules to align different exposures to the reference images. Subsequently, the outputs of the modules were concatenated in the reconstruction stage to produce the initial version of the HDR image. Finally, the model's output was fed into the refinement stage to address potential distortions, such as blurriness and ghosting. Moreover, after comparing the results of our proposed method with those of the state-of-the-art methods, our results outperformed the state-of-the-art in most cases. The paper was published in the **IEEE Access journal** [58].

In another part of our research [55], we explored the potential of using a single facial image to differentiate between children diagnosed with autism and those who are not. We utilized various versions of Vision Transformers and ResNets for this purpose and achieved a high accuracy rate of 92%. We also extended our investigation to understand the specific features that the model used to make these distinctions. To achieve this, we employed techniques from the field of explainable AI. The results of this research were submitted to the **Journal of Research in Autism Spectrum Disorders**.

In addition to facial research in autism, we also explored the detection of vocal stereotypy

in children with autism using machine learning methods applied to audio files [56]. For this purpose, we structured our research into several stages, ranging from pre-processing to training various models on the data. We studied three different analyses on audio files recorded from autistic children and trained a model for each patient to detect vocal stereotypy, as their type was different. The paper was submitted to the **Journal of Applied Behavior Analysis**.

Chapter 2

SUPERVISED IMAGE SEGMENTATION FOR HIGH DYNAMIC RANGE IMAGING

2.1 Introduction

Natural scenes have a vast luminosity; however, regular cameras can capture a limited dynamic range of that luminance. Therefore, the generated image has regions with high (overly bright) and low exposure (too dark), and the detail is not well visible. These types of pictures are called Low Dynamic Range (LDR) images.

The first solution to this problem is to utilize cameras with special sensors, which can obtain more luminance than regular cameras and produce images with more details that are more similar to the real world [54, 81, 48, 79, 25, 99, 75]. However, due to the high cost of such equipment, it is not affordable or usable for regular users.

Another solution for this issue is using software development methods known as HDR imaging. Various algorithms have been proposed recently, and the existing techniques can be divided into HDR imaging with Single-Exposure and Multi-Exposure methods. In the Single-Exposure approach, various techniques can produce an HDR image starting from a single LDR image. However, these methods are not satisffactory since the details cannot be restored well. In [15], the authors proposed an algorithm to generate an HDR image from an LDR image. However, their method was affected by two problems: the inability to reconstruct details of dark and overly saturated areas. More precisely, this algorithm could not retrieve the details in the excessively saturated regions. Therefore, [60] proposed merging input images with different exposures and afterward feeding the wavelet coefficient of the merged

image to the network to produce more details in a shorter time. Fortunately, unlike the Single-Exposure methods, Multi-Exposure methods are more effective and can reconstruct more information. Several LDR images are combined in such techniques to produce an HDR image. Although Multi-Exposure methods perform almost perfectly on static scenes, they can encounter problems such as ghosting in dynamic scenes due to moving objects. However, several algorithms have been proposed to solve this issue [35, 30, 88, 90, 66, 65].

HDR imaging can be used in various applications, from entertainment to medical and security. This technique produces images with more detail in both highlights and shadows in photography. Additionally, HDR imaging helps the entertainment industry provide more vivid colors and more realistic content; currently, many modern TVs and streaming services support HDR content. HDR imaging can also be used in various medical fields to improve the visibility of medical images such as MRIs and CT scans, aiding better diagnoses. This technique can also help to have clearer surveillance footage in terms of lighting conditions.

In consideration of the relevance of HDR imaging, new methodologies based on deep learning have been a great help in recent years, in providing significant progress over the state of the art. For instance, [15] used a deep neural network to produce an HDR image in the logarithmic domain. Also, [85] used deep learning to reconstruct the detail of an image with different row-wise exposure in the irradiance domain. The works [39, 17], unlike previous methods, used neural networks to produce several LDR images with different exposures from a single LDR image. Additionally, [30] first aligned images with the optical flow and eventually used deep learning to fuse the aligned images to produce an image with more detail. In [35], two deep learning methods were used to align images and generate an HDR image. Neural networks with different scales of images were used in [91] to learn the relative relation between input images and their ground truth.

Image segmentation is one of the tasks in computer vision, and its objective is to simplify image analysis. This task is typically used to detect objects or better understand images, such as medical ones. Image segmentation can extract the regions of pictures with more details. In [83], the authors analyzed images in HSV color space to segment pixels based on the value of Intensity or Hue. Additionally, other works proposed two methods for image segmentation based on luminance: histogram division [34] and clustering based on the Gaussian Mixture Model (GMM) of the histogram [33]. Furthermore,[38] proposed a method to find the optimal valley point based on the slope between the histogram value of each pixel and other neighboring points and used that valley point to segment regions.

Thus, to cope with this problem, highly advanced cameras [54, 81, 48, 79, 25, 99, 75] can be used, which are equipped with special sensors that capture more light. However, such devices are mainly too expensive and overly heavy, making them unsuitable for daily life and

primarily used in industries.

A possible resolution for this drawback is developing software algorithms called HDR imaging techniques. Moreover, HDR images can be implemented using a single image [15, 77, 36, 8] or by fusing a stack of images with different exposures, which are called single-exposure and multi-exposure methods, respectively. In algorithms utilizing a single LDR image, an HDR image can be produced from one image; however, the generated picture may not be as informative as an HDR image produced by several LDR images because the amount of detail in one picture is limited compared to several images with different exposures. More precisely, [15] implemented an algorithm that only reconstructs the details of bright saturated areas; however, the model cannot restore detail in dark regions and performs poorly when the bright saturation is excessive. Thus, [60] first combined several LDR images and then fed the low-frequency response of the wavelet transform to the network to produce more detail in a shorter time.

Luckily, multi-exposure methods are more effective and informative compared to single-exposure techniques. Moreover, these methods perform well when the images are static [39, 17]; however, when there is movement in the sequence of pictures, the ghosting problem emerges, which is almost solved in [35, 30, 88, 90, 66, 65].

Deep learning has been a significant means of producing HDR images for the past decade. For instance, [15] produced an HDR picture in the logarithmic domain with the help of a deep neural network. Additionally, [85] used a neural network to reconstruct details in an image with different exposures in each row in the irradiance domain. In contrast, unlike other multi-exposure methods, [39, 17] used a neural network to produce synthetic LDR images with different exposures from a single image. Furthermore, [30] proposed aligning images using the optical flow and then using a deep neural network to combine them. In addition, [35], instead of using optical flow for alignment, proposed using two different neural networks: first to align them and then combining the aligned images with the second neural network. Finally, [91] used a neural network to learn the relative relationship between the input images and their Ground Truth using images at different scales.

The main contributions of this chapter are as follows:

1. We propose two methods to extract areas of the image with greater detail.

2. We compare the proposed methods to identify the best one.

2.1.1 Basic definitions in HDR[5]

HDR

HDR is a technology that improves the color range and brightness of displays. This technique creates more realistic images than those produced by Standard Dynamic Range (SDR) cameras. HDR expands the range of brightness and color in images to make them similar to what we see with our eyes. This helps us easily recognize the brightest and darkest parts of the image and shows more detail in both shadows and highlights.

HDR Challenges

A common challenge in HDR techniques is ghosting artifacts, which occur when multiple images with different exposures are merged. If there is movement in the scene while taking a bracketed series of images or the images are not aligned, the merging process causes ghosting artifacts, which can make the picture blurry. However, nowadays, there are various techniques, both machine learning and non-machine learning, to address this problem.

Tone mapping

Although it is increasingly common for digital displays and streaming services to support HDR content, such devices are still not commonplace, and mostly support LDR content. Since most displays cannot produce the full dynamic range of HDR content, tone mapping is a technique that converts the wide range of luminance values into a displayable range on standard screens.

Sigmoid Space

Training a model for HDR imaging in HDR space results in challenges due to the high pixel values. To address this, we mapped these values into sigmoidal space, normalizing them to a range of 0-1 for more effective training. Once the model is trained, we convert the values back from sigmoidal space to HDR space using the inverse sigmoid function.

Color Spaces

Various color spaces can be used in HDR imaging, and these will be discussed briefly. **RGB** color space shows colors by combining red, green, and blue lights, and is the most common color space in imaging and displays. **YCbCr** is a color space that is used in video compression and broadcasting. Y represents the luminance component, while Cb and Cr represent color

difference components. **YUV** is similar to YCbCr, in which Y represents the luminance, while U and V represent chrominance information.

2.2 Proposed Method

2.2.1 Producing ground truth

Most of the algorithms in the literature for HDR imaging are concentrated on how to produce the actual images, while less attention has been paid to extracting suitable features. In this thesis, the proposed method focuses on extracting the most suitable regions for HDR imaging. Indeed, by finding the areas with more detail, the HDR algorithm can produce an image free of overly saturated or dark parts. More specifically, an image segmentation method is introduced to identify areas with the most detail. A neural network can then extract the desired regions of input images, which will be discussed in the next chapter. Additionally, two different methods, manual thresholding and Otsu segmentation, were used to produce the Ground Truth, and these will be compared with each other.

In the manual technique, several experts investigated the best possible range of intensity in the YCbCr color (explained in 2.1.1) space for empirically extracting the areas with the most detail. Eventually, an average of the ranges was calculated for each image. The selected ranges for image intensity with low- and high-exposure are [120,255] and [0,200], respectively. Generally, the objective is to acquire areas with less darkness and saturation. Therefore, because most regions in low-exposure images are dark, we would like to extract the areas with the highest pixel values, which indicate the most visible ones. Conversely, because most pixels in high-exposure images are saturated, the objective is to extract pixels with the lowest values. Indeed, some visible pixels with the lowest values cannot be selected by choosing pixel values in the luminance channel. For example, although the gray area of the mountain in Figure 2.1 is visible, it was not chosen in the segmentation process.

Figure 2.1: The image on the left is the input image, and on the right is its Ground Truth produced by the manual method. The picture was taken from [18].

The second method is called the Otsu technique [61], which calculates a threshold based on the intensities of images and segments pixels. More precisely, the pixels greater than the threshold are considered foreground (white), and those with lower values are considered background (black). The difference between these two methods is that the Otsu technique threshold is computed based on the histogram of each image. In the manual method, all the pictures of each exposure are within the same range. In Otsu, all the pixels of low-exposure photos greater than the threshold are considered the desired pixels. In contrast, those lower than the threshold in high-exposure pictures are desirable. In both cases, the segmentation masks are made of ones and zeros.

2.2.2 Neural network structure

Unfortunately, each image encountered in practice has various intensities and peculiarities; for this reason, it would be challenging to use non-machine learning methods to predict the informative areas in low- and high-exposure images. Moreover, it is a time-consuming task to extract a range for each image separately. Therefore, a neural network has been proposed in this research to learn how to extract the best area of each image based on the proposed ranges in the training stage.

Two similar U-Net-shaped networks [72] were used for segmentation in this research, and each network attempts to learn how to map from each exposure to its Ground Truth. These two networks are different from the original version regarding blocks and block sizes. As seen in Figure 2.2, the U-Net consists of 2 parts. In the first part, the subnetwork strives to extract features, and the second subnetwork tries to produce an output similar to the Ground Truth. The encoder section includes five blocks, each with two convolutional layers with ReLU function, Dropout, and MaxPool layers. Additionally, kernels of convolutional

layers in each block are 16,32,64,128,256, respectively. Moreover, the decoder has four blocks, each consisting of one transpose convolutional layer, a concatenation, a convolutional layer with ReLU activation, Dropout, and another convolutional layer with ReLU, respectively. Furthermore, all convolutional and transposed convolutional layers use a kernel size of 3x3, and the last layer uses a kernel size of 1x1.

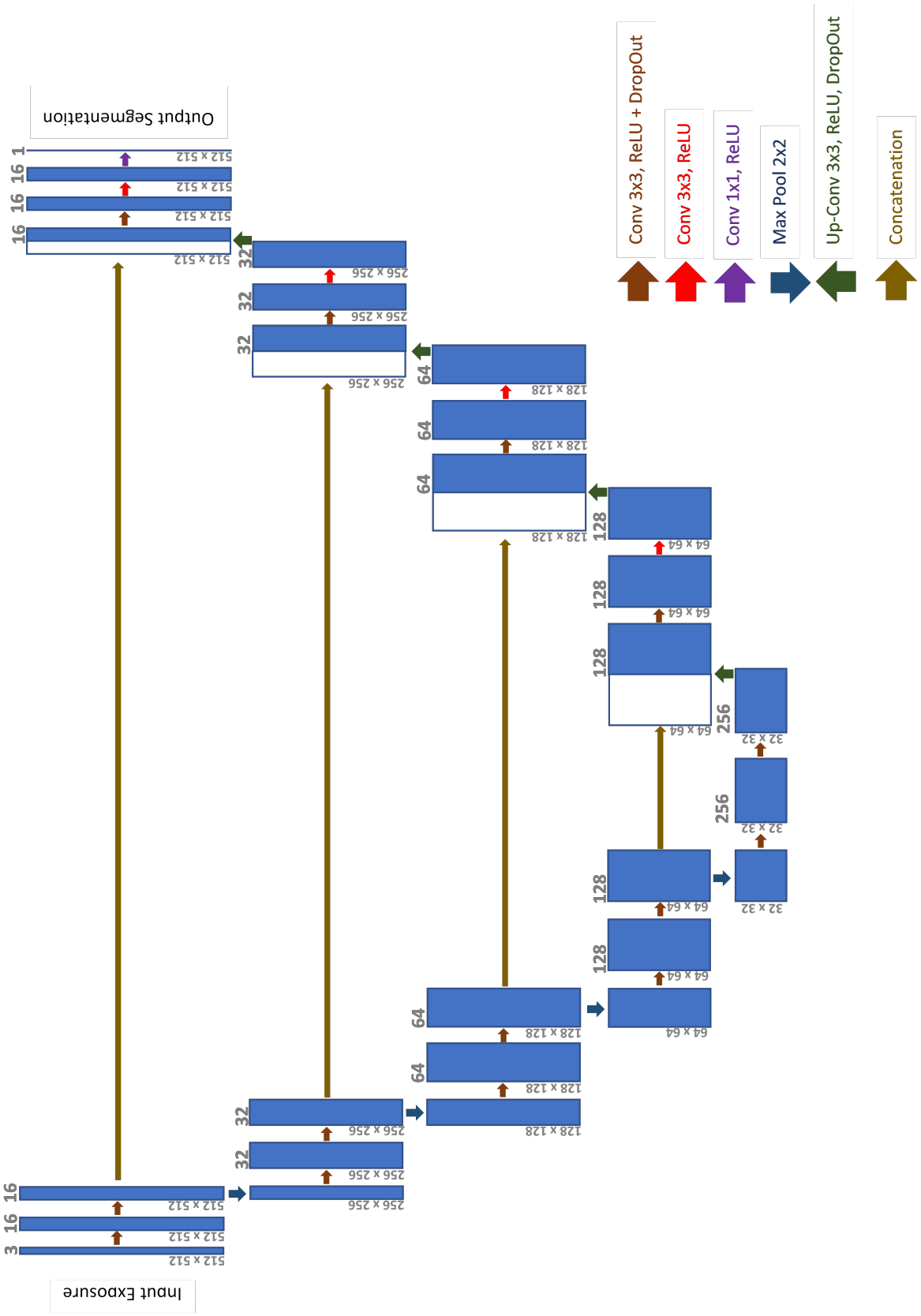

Figure 2.2: A schematic diagram of the U-Net architecture was used in this experiment. The blue boxes denote feature maps; their number is on the top of each box, and their size is indicated on the lower-left side of each box.

2.2.3 Loss functions

The loss function is one of the essential components of deep learning. Thus, three loss functions are used and compared to select the best loss function for segmenting the regions with the most detail. The three loss functions used are as follows:

1. Binary Cross Entropy (BCE): One of the most common functions, which is used in most image segmentation research, is the BCE loss function, and it can be represented as follows:

$$L_{BCE} = -\sum(y \log \hat{y} + (1-y)\log(1-\hat{y})) \tag{2.1}$$

   Where y and ŷ represent Ground Truth and the network's output, respectively, the sum is over all the pixels.

2. Focal Loss [42]: This loss function is used for imbalanced data and focuses on hard data:

$$L_{focal} = -\sum(\alpha y(1-\hat{y})^{\gamma} \log \hat{y} + (1-\alpha)(1-y)(\hat{y})^{\gamma} \log(1-\hat{y})) \tag{2.2}$$

   Where α and γ are hyperparameters and, as a default, equal to 0.25 and 2.0, respectively.

3. Combo Loss (Dice Cross-Entropy) [22]: This loss function is also used for imbalanced data and is produced by combining Cross-Entropy and Dice loss functions. Eq 2.3 represents Dice loss, and Eq 2.4 is for Combo loss:

$$L_{Dice}(y, \hat{y}) = 1 - \frac{2y\hat{y} + 1}{y + \hat{y} + 1} \tag{2.3}$$

   The value 1 added to the numerator and the denominator avoids undefined errors, such as y=ŷ=0.

$$L_{DiceCE} = L_{Dice} + L_{BCE} \tag{2.4}$$

2.3 Experiment Results

2.3.1 Dataset

Recently, a new dataset was collected for HDR Imaging Challenge called NTIRE 2021 [62]. In this dataset, two types of pictures (Single-Exposure and Multi-Exposure images) were provided; however, Multi-Exposure photos were used only in this research. This dataset includes images from [21] that were generated as follows. First, HDR images were produced

natively by two Alexa Arri cameras with a mirror rig; then, their corresponding LDR images were generated synthetically with noise sources. Approximately 1500 pairs of HDR/LDR images are in this dataset for the training set, 40 for the validation set, and 200 for the test set, with a resolution of 1900x1060. Moreover, all the images were already aligned and gamma-corrected.

2.3.2 Evaluation metrics

Several evaluation parameters have been used in this research to evaluate the results and are discussed as follows:

1. **Dice Index:** This metric is region-based and evaluates the similarity and the overlap of two samples.

$$Dice(A, B) = 2\frac{|A \cap B|}{|A| + |B|} \tag{2.5}$$

2. **Jaccard Index:** This metric works similarly to Dice and calculates the similarity of two samples.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.6}$$

3. Two other metrics are Sensitivity and Specificity, which calculate True Positive and True Negative pixels.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.8}$$

4. **AUC:** This metric is commonly used in image segmentation algorithms.

$$AUC = 1 - \frac{1}{2}(\frac{FP}{FP + TN} + \frac{FN}{FN + TP}) \tag{2.9}$$

2.3.3 Ground truth generation

As the dataset used does not include ground truths for segmentation, the first objective of this research is to produce ground truths that cover most of the scenes. Thus, after frequent

visual studies of the ground truths produced by both manual and Otsu techniques, it became evident that the manual method has more coverage than the latter. For instance, as seen in Figure 2.3, both approaches worked almost the same on low-exposure images. However, the manual method succeeded in covering more areas in images with high-exposure. Additionally, as seen in the last row, the total area covered by the manual technique is more significant than that covered by the Otsu technique. Therefore, the ground truth produced using the manual method will be used for the rest of the research.

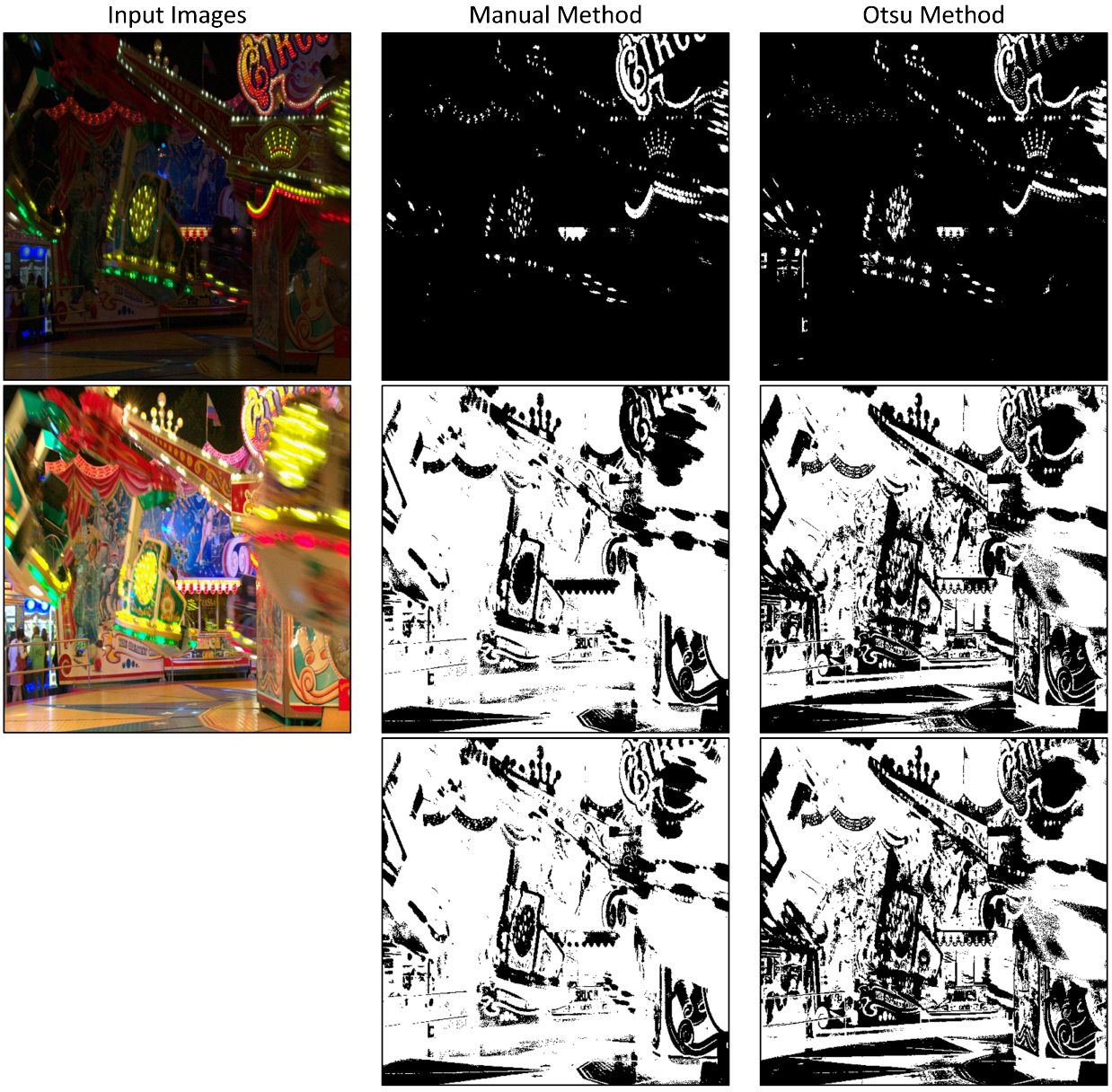

Figure 2.3: Produced ground truth of both manual and Otsu methods. The first row is generated from the low-exposure image, the second is obtained from the high-exposure image, and the third is a merged output of both rows.

2.3.4 Other details

Additionally, the training process for each loss function was 50 epochs, which took around 200 minutes on NVIDIA DGX A100 and less than 2 minutes for testing, and the model was trained in parallel on 4 GPUs. Moreover, the number of images for the training and validation sets was about 1300 and 200 images, with a resolution of 512x512 and a batch size of 32, respectively. Furthermore, the Adam optimizer with a learning rate of 0.001 was used. Finally, the neural network was implemented in the TensorFlow (Keras) framework. During experiments, three input images with different exposures were used for image segmentation, in which, after obtaining the suitable areas of low- and high-exposure photos, the remaining regions were extracted from the medium-exposure images. However, the acquired areas of the medium-exposure image were not sensitive because most were only a few pixels with no shapes. Thus, it was difficult for the network to segment them. Figure 2.4 demonstrates an example of the extracted regions in the medium-exposure image.

2.3.5 Results

The predicted segmentation outputs from three different loss functions were quantitatively compared with their ground truth, which was produced by a manual technique. As shown in Table 2.1, which demonstrates the evaluation results of low-exposure image segmentation, different loss functions outperformed the others in various evaluation metrics. For instance, the Focal Loss function performed better than the others in Jaccard and Sensitivity evaluation metrics. Although they have equal values in the AUC evaluation metric, the average Focal Loss was better than Dice-BCE and BCE. Additionally, Table 2.2 indicates that the Dice-BCE loss function worked better than the other two losses in Jaccard and Sensitivity evaluation metrics. However, BCE performed better on average. Therefore, it can be concluded that the Focal Loss function segments low-exposure images with better illumination, while BCE performs better in high-exposure images. Figure 2.5 and 2.6 demonstrate outcomes produced by different loss functions for both images with low-exposure and high-exposure. As can be seen, although all the outputs are visually almost identical and it is difficult to distinguish differences between them, the quantitative results show that the output of Dice-BCE is not as good as that of the other two.

Figure 2.4: An example of extracted areas from a medium-exposure image. The image is taken from [18].

Table 2.1: Quantitative evaluation results of low-exposure Image Segmentation. The rows represent the metrics, as M1: Dice, M2: Jaccard, M3: Sensitivity, M4: Specificity, M5: AUC, and AVG represnts the average of the metrics.

Loss Functions	m1	m2	m3	m4	m5	AVG
DCE	0.951	0.905	0.912	**0.999**	0.498	0.853
Foocal	0.916	**0.936**	**0.997**	0.997	0.498	**0.869**
Dice - BCE	**0.965**	0.933	0.912	**0.999**	0.498	0.861

Table 2.2: Quantitative evaluation results of high-exposure Image Segmentation. The rows represent the metrics specified in Table 1.

Loss Functions	m1	m2	m3	m4	m5	AVG
DCE	**0.994**	0.909	0.765	0.754	0.68	**0.82**
Foocal	0.989	0.89	0.753	**0.763**	0.675	0.814
Dice - BCE	0.991	**0.912**	**0.77**	0.73	0.67	0.815

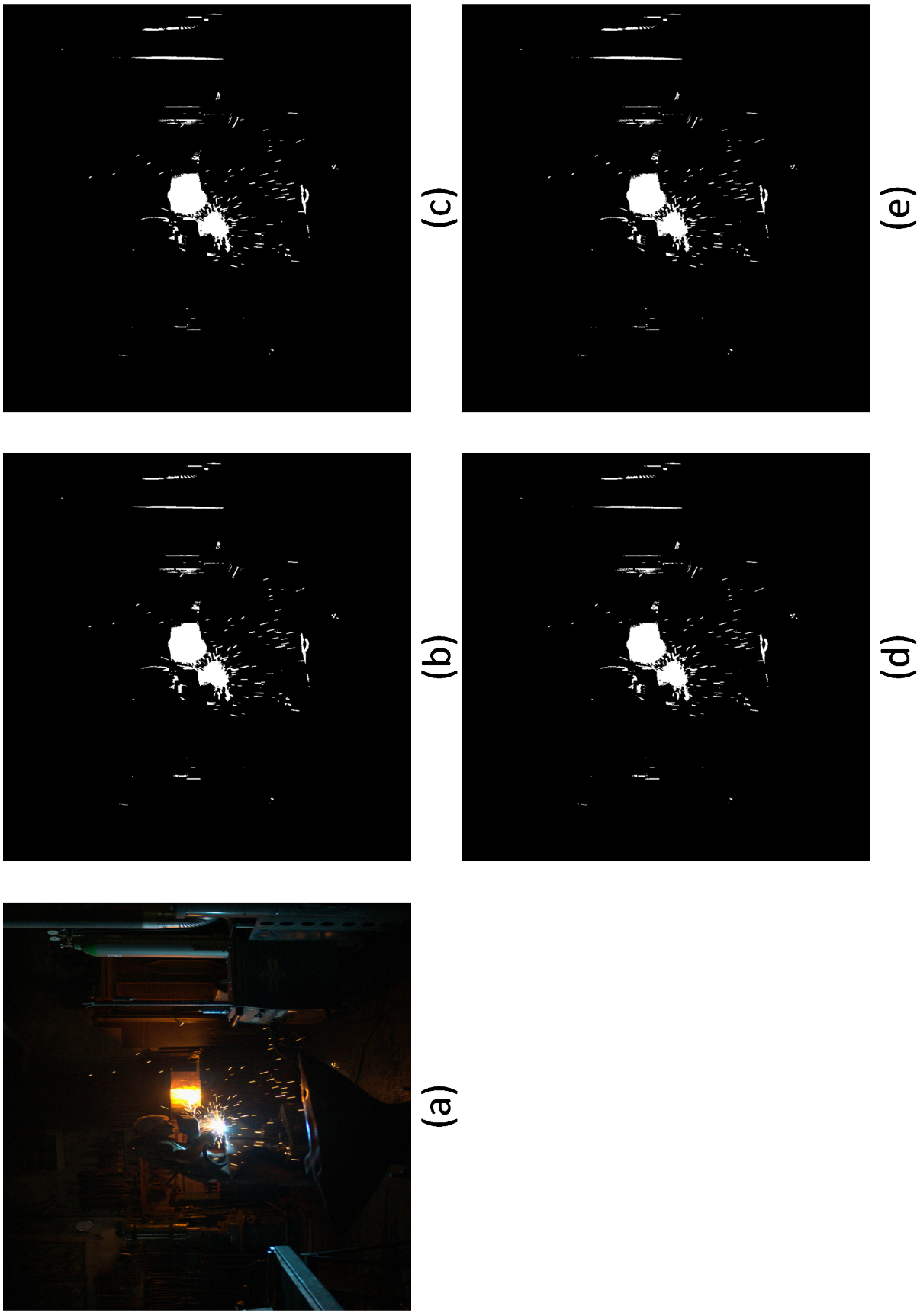

Figure 2.5: Output results of different loss functions. (a) Low-Exposure input image, (b) Dice-BCE output, (c) BCE output, (d) Focal output, (e) Ground truth.

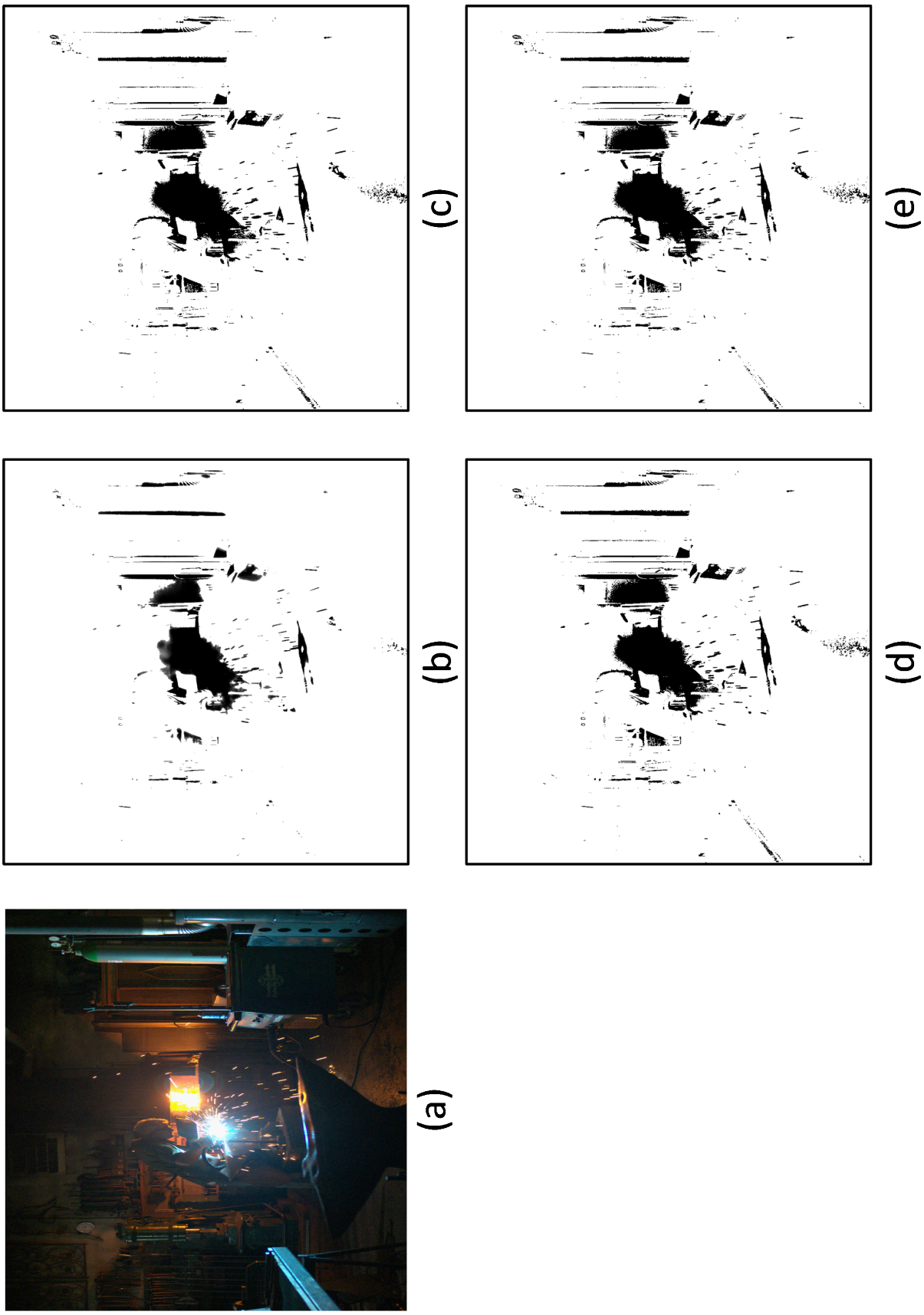

Figure 2.6: Output results of different loss functions. (a) High-Exposure input image, (b) Dice-BCE output, (c) BCE output, (d) Focal output, (e) Ground truth.

2.4 Conclusion and future works

As discussed in Section 2.2, Otsu and manual methods were used in this research, and a range was computed empirically using the manual technique. Although experiments demonstrated that the empirical approach produced better outcomes than Otsu, it has two drawbacks. Firstly, fails to recognize dark, visible areas, such as the mountain peak illustrated in Figure 2.1. Secondly, when the segmentation process is performed using the manual technique, the calculated range must be applied to all images, and the computed span may be unsuitable for some photos; thus, calculating a specific range for each picture is also time-consuming. Therefore, it would be better to develop a new automatic technique to estimate these ranges for each image. In addition to working on a novel method for calculating an automatic range for each image in future work, it is feasible to use extracted regions from segmentation techniques in HDR imaging to produce an HDR image with more detail. Additionally, this work can help reduce the complexity and computational demands of networks for generating HDR images, as we will explore in the next chapter.

In summary, this research used two methods for segmenting visible regions, and a manual, empirical approach was chosen after comparing them to produce the ground truth. Moreover, deep neural networks were used to learn to extract the regions with the help of the produced ground truths in each exposure. Additionally, three different loss functions were utilized in this chapter, and the quantitative metrics demonstrated that the focal and BCE loss functions outperformed others in low-exposure and high-exposure images, respectively.

Chapter 3

High Dynamic Range Imaging via Visual Attention Modules

3.1 Introduction

In the context of photography, the real world consists of an unlimited range of luminance. However, as introduced in Chapter 2, most imaging devices can capture only a limited amount of that light. Therefore, the captured images are often undesirable and containing saturated regions where some parts of the images are too dark (underexposed) or overly bright (overexposed), resulting in that are called LDR images.

In this chapter, we will apply image segmentation, as described in the previous chapter, to HDR imaging to extract the most visible areas of the images and help the model generate pictures with more detail. By doing so, we can reduce the complexity of the neural network and achieve similar or better results. Indeed, deep learning methods have demonstrated outstanding capabilities in identifying the most relevant features in images. For the current task, these methods are, in principle, capable of identifying the most informative areas in photos with different exposures. However, this may require higher network complexity and an intractable number of parameters. Conversely, this paper investigates the potential role of segmentation in guiding the network architecture toward superior HDR reconstruction. To achieve this, VAMs will be proposed to extract such regions. Additionally, this research incorporates Spatial and Attention modules from SOTA methods, and a new architecture for the Reconstruction stage is designed and implemented, in which the visual attention and the reference image are used in the decoder part. Finally, although VAMs helped produce images with more detail and outperformed most SOTA methods, the results still showed a slight amount of noise extracted from the input images.

In section 3.2, the SOTA methods in HDR imaging and related image segmentation are presented. In Section 3.3, the proposed method is discussed in detail. Section 3.4 demonstrates the experimental results and compares them with the SOTA methods. Moreover, Section 3.5 presents ablation studies to validate the significance of each proposed step, focusing on the use of VAM and the Refinement stage. Finally, Section 3.6 concludes the chapter with ideas for future work. The code is available **here**.

3.2 Related Work

In this section, we will discuss the SOTA methods in HDR imaging within the Multi-Exposure category (Section 3.2.1) and review unsupervised image segmentation methods for region extraction (Section 3.2.2).

3.2.1 Multi-Exposure Methods

Given the extensive literature on HDR imaging, as well as the availability of excellent books and surveys on the topic [86, 5], this section reviews methods closely related to the one proposed in this chapter, aiming to highlight potential commonalities, differences, and innovative aspects. [41] proposed a two-stage algorithm in which, in the first phase, features were extracted from the input images and merged to produce the HDR image in the second phase. Additionally, to address the noise introduced by the gamma correction operation on input images - i.e., the gamma-corrected low-exposure image becoming similar to medium-exposure - they used a U-net to extract noiseless features. Furthermore, [12] implemented a model in which images at lower scales were used to reduce resource consumption. A novel loss function was also defined to emphasize motion. Moreover, [45] forwarded features at different scales to deformable and spatial attention blocks to align images in the feature space and extract the features from specific areas of the input images. Additionally, [93] proposed a model that first estimated the optical flow from two input images at different scales and then fused them to produce the final output. In [89], features were extracted from various scales and processed by sampling and aggregation modules to align the pixels of the non-reference features.

The work in [92] implemented a baseline with lower computational resources and acceptable results compared to other SOTA models. They used a dual attention module, which includes both spatial and channel attention modules, to address misalignment and better learn the details of the produced areas. In [95], the authors proposed a model that first extracts features from input images using multi-scale encoding modules and then creates an

HDR image by applying progressively dilated U-shape blocks.

[11] demonstrated that the ghosting problem (explained in Section 2.1.1) primarily affects low-frequency signals. Therefore, they proposed a wavelet-based model to merge images in the frequency domain and avoid ghosting issues. [64] implemented an algorithm that extracted dynamic areas of the images using image segmentation and applied two neural networks separately to the static and dynamic scenes. Finally, they merged the information to produce an HDR image without ghosting. In [84], a model based on bidirectional motion estimation was proposed, where the optical flow between LDR images was estimated using motion estimation with cyclic cost volume and spatial attention maps. Eventually, an HDR image was produced with the help of the extracted local and global features. [49] implemented the first multi-bracket HDR pipeline using event cameras, where they merged the extracted features of images and events to produce an HDR image. [43] proposed a transformer-based baseline, using a context-aware vision transformer to extract local and global features to model the movement of objects and the diversity of intensity.

3.2.2 Image Segmentation

Image segmentation is a crucial task in computer vision, aiming to partition images into segments for easier analysis. It can be used not only for object recognition, detection, and medical purposes, but also for extracting regions of images with more detail. In [83], images were analyzed in the HSV color space to segment pixels based on intensity or hue values. Moreover, two image segmentation methods based on luminance were proposed: histogram division [34] and clustering using GMM of the histogram [33]. Furthermore, [38] calculated an optimal valley point based on the slope between the histogram value of each pixel and its neighboring points, using this computed valley point to segment regions. The literature on the topic is vast, with methods ranging from level set methods [53] to graph cuts [94] to recent deep learning-based frameworks [52].

3.3 Proposed Method

3.3.1 Overview

As cited in [87], it may be beneficial to first segment images based on exposure information to extract the most detailed regions from the over- and under-exposed areas and use this knowledge to reconstruct an HDR image. Following this idea, a model is proposed in this paper in which regions with more detail are first segmented in the processing stage with the

help of image segmentation. These regions are then fed into the model, along with the input images, to produce an HDR image using VAMs.

The model can generally be divided into several sections. First, the input images are fed into the feature extraction module, and then the extracted features pass through the attention and spatial alignment modules to address any potential misalignment. Moreover, the input images, along with their corresponding masks, are processed by the VAM simultaneously to extract the visible areas of the LDR images. Next, the outputs of the three modules are fed into the Reconstruction stage to generate the initial HDR image. Finally, the generated outcomes, along with the features of the reference image, enter the refinement section to construct the final HDR image.

3.3.2 Preprocess

In this article, the inputs consist of three LDR images with different exposures, with the image having medium exposure considered as the reference image. Before feeding the input images to the model, they are first mapped to the HDR domain using gamma correction. Finally, the images are concatenated channel-wise with their corresponding LDR images.

$$\hat{I}_i = \frac{(I_i)^\gamma}{t_i} \qquad \text{for } i = 1, 2, 3 \tag{3.1}$$

where t_i is the exposure time of I_i. γ is the gamma correction parameter(set to 2.24), and $\hat{I}_i$ is the gamma-corrected image.

Segmentation

As discussed in the previous chapter, most current algorithms in HDR imaging focus primarily on the image production process, with less attention given to how to extract the most helpful features. In this thesis, however, the regions of the images containing more detail are first segmented and extracted as a preprocessing step. These regions are then fed into the proposed model, along with the LDR images, as inputs.

In the first chapter, we used the empirical and Otsu methods for segmentation, with the empirical technique outperforming the latter. However, during the implementation of the overall scheme in this chapter, we observed that the empirical method led to overfitting in our network. Various methods, such as neural network-based approaches and the Otsu method, were also explored for the image segmentation stage. However, in our experiments, the neural network-based approaches resulted in overfitting. As a result, the widely adopted Otsu method, with its simplicity, was selected to segment the visible areas of the images. To

begin, the images are first converted to the YUV color space (explained in Section 2.1.1). The luminance channel (Y) is then used to compute a threshold based on the histograms of low- and high-exposure images. Different thresholds are calculated for each sample based on the histogram of each image in each exposure. Thus, the threshold parameter for each image is a variable, depending on the sample.

$$\text{thresh}_i = G(Y_i) \qquad \text{for } i = 1, 3 \tag{3.2}$$

Where Y_i is the luminance channel of the LDR image, $G()$ is the Otsu function, and thresh_i is the threshold value of image i.

In the case of the low-exposure image, where most pixels are dark, the objective is to extract the regions with visible pixels. Therefore, values equal to or greater than the threshold are set to 1, and the rest are set to 0 for the low-exposure mask.

$$\begin{cases} 1 & p \geq \text{thresh}_1 \\ 0 & p < \text{thresh}_1 \end{cases} \tag{3.3}$$

where thresh_1 is the threshold value of the low-exposure image, and p represents the pixel.

In contrast, in the high-exposure image, where most pixels are saturated, the visible pixels have the lowest values. Therefore, values less than the threshold are set to 1, and the rest are set to 0 for the high-exposure mask.

$$\begin{cases} 0 & p \geq \text{thresh}_3 \\ 1 & p < \text{thresh}_3 \end{cases} \tag{3.4}$$

By doing this, the masks of the areas with more detail are extracted, which can aid in producing an HDR image.

Generally, most pixels in low- and high-exposure images are either too dark or too bright, respectively. Thus, the areas with surplus information are extracted and fed to the model. This reduces computational load and helps produce an HDR image with more detail. Figure 3.1 illustrates the segmented and visible regions of both low- and high-exposure images (the first and second masks from the left, respectively), with the third mask being the sum of both generated masks.

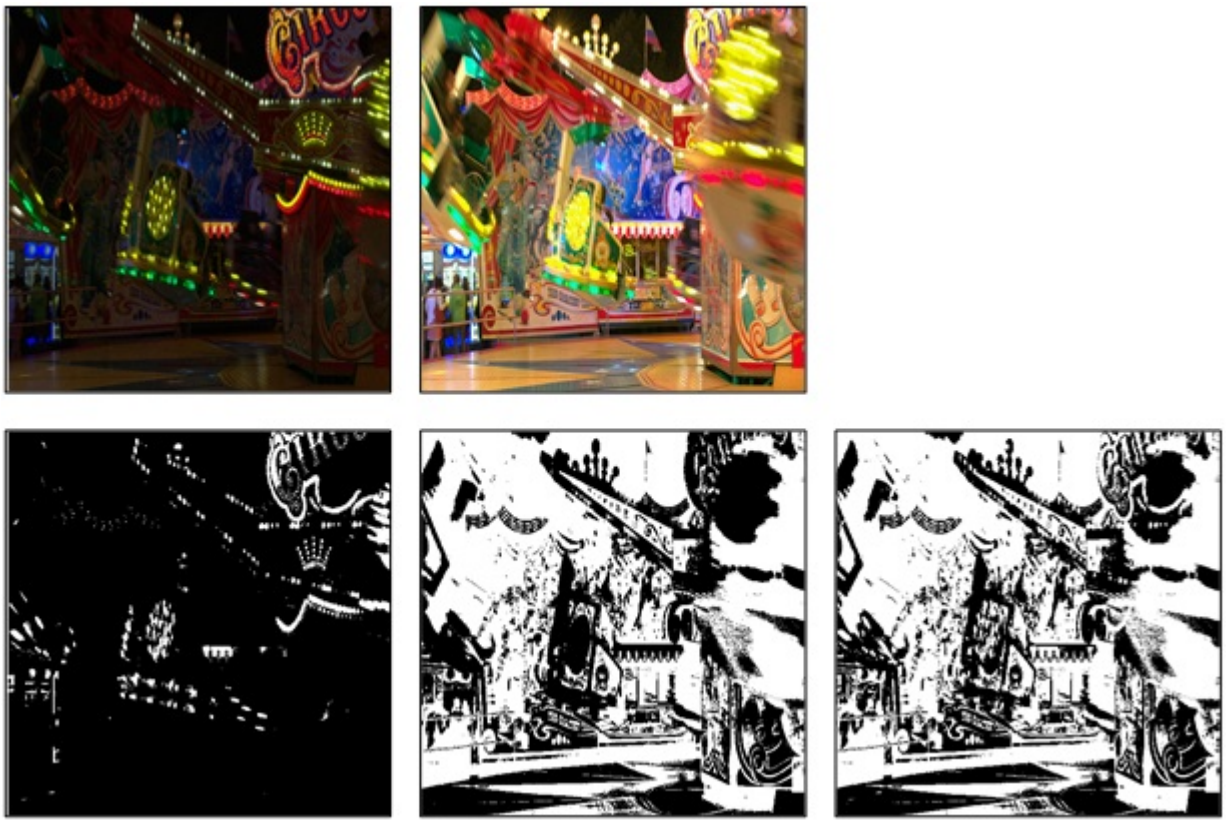

Figure 3.1: Generated masks for low- and high-exposure images.

Furthermore, during our experiments, three input images with different exposures were used for image segmentation. After identifying the suitable areas from the low- and high-exposure images, the remaining regions were extracted from the medium-exposure image. However, the extracted regions from the medium-exposure image were not as useful, as they mainly consisted of a few pixels. Therefore, there are two reasons for not using the medium-exposure image in the segmentation stage. First, it would be challenging to calculate a range for the visibility of the pixels. Second, since the medium-exposure image is the reference image, and the picture will be used in the neural network. Therefore, it will be used in the neural network. Therefore, it is not necessary to use information derived from the segmentation of the medium-exposure image.

3.3.3 Proposed Method Structure

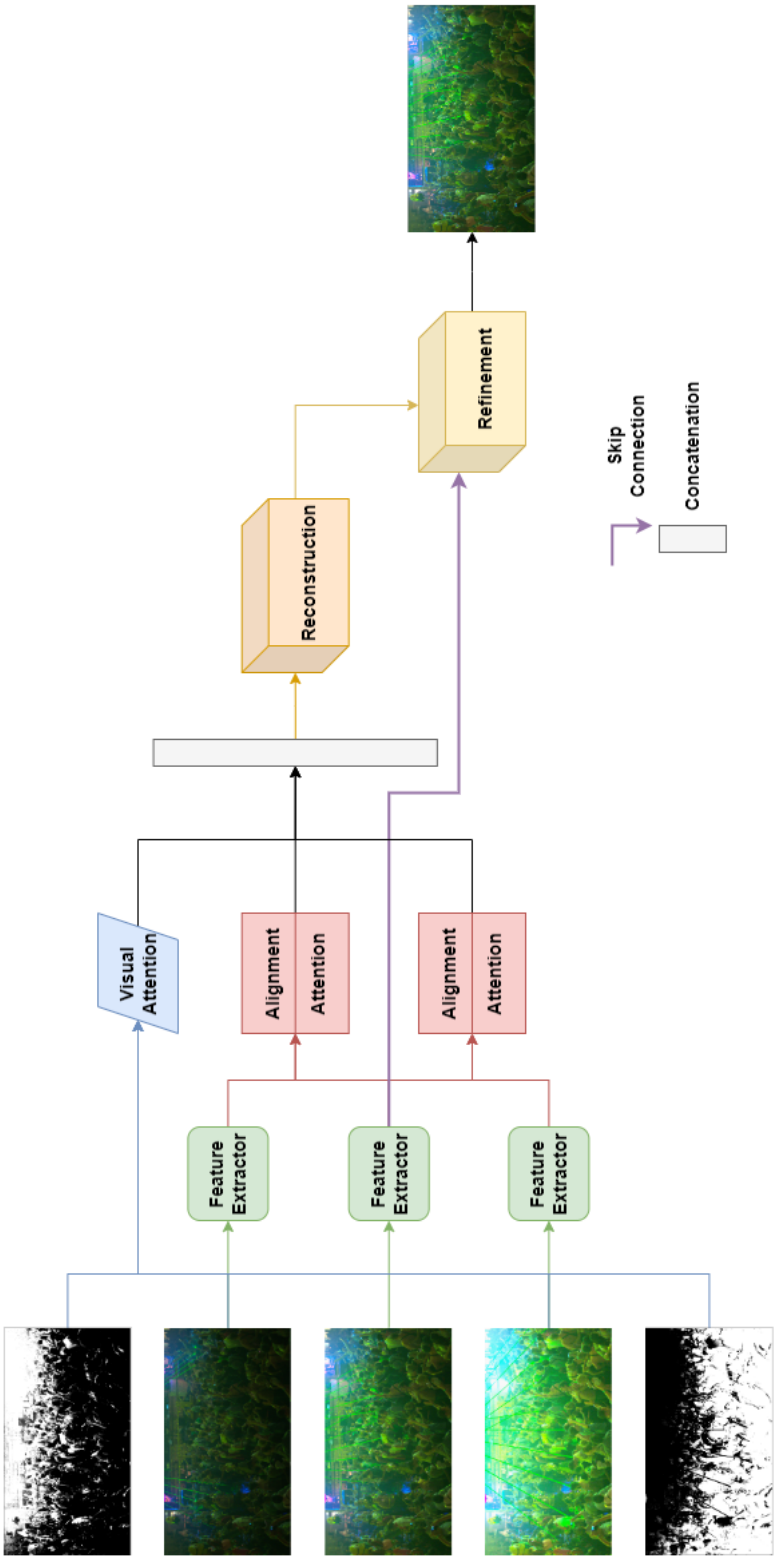

Figure 3.2: The complete pipeline of the proposed method.

As shown in Figure 3.2, the proposed algorithm consists of six stages, each of which will be discussed separately and in detail.

Feature Extraction

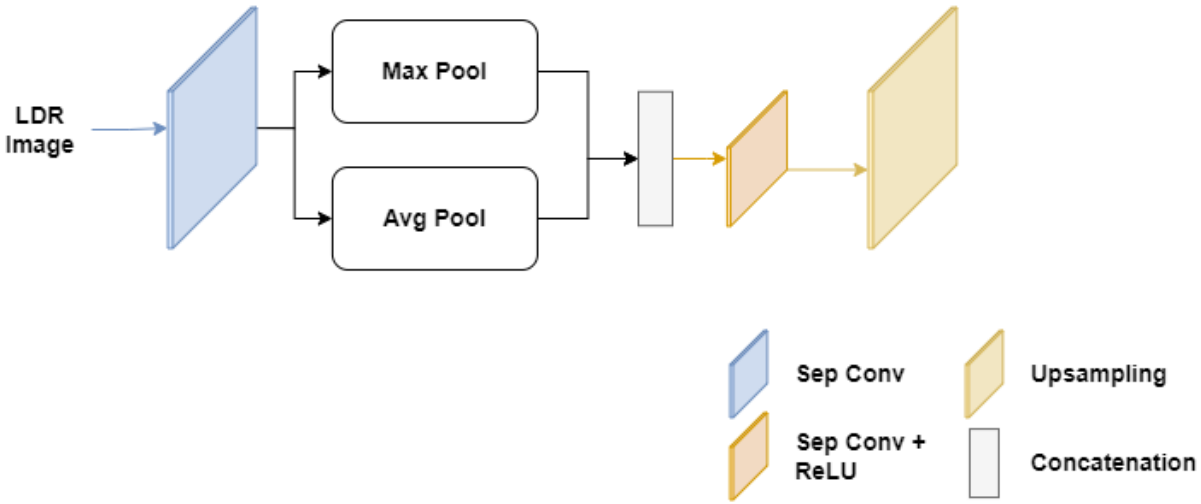

Figure 3.3: Structure of the Feature Extraction Block.

Figure 3.3 illustrates the Feature Extraction block, where a SepConv is applied to the image to extract 32 feature maps. Next, Max Pooling and Average Pooling operations are used to smooth the features, emphasizing the details, and focusing more on the edges. The outputs of these pooling operations are concatenated, and another SepConv followed by a ReLU activation function is applied to reduce the number of channels to 32. Finally, the extracted features are upsampled to match the input image's dimensions. The feature extraction process can be written as follows:

$$\text{features}_i = \text{SepConv}(I_i) \tag{3.5}$$

$$C_i = \text{concat}\left(M(features_i), A(features_i)\right) \tag{3.6}$$

$$F_i = \text{Upsample}(\text{ReLU}(\text{SepConv}(C_i))) \tag{3.7}$$

for $i = 1, 2, 3$, where $M()$ and $A()$ represent Max Pooling and Average Pooling functions, respectively, and C_i denotes the output of Concatenation. Finally, F_i is the output of the Feature Extraction Block.

Visual Attention Module

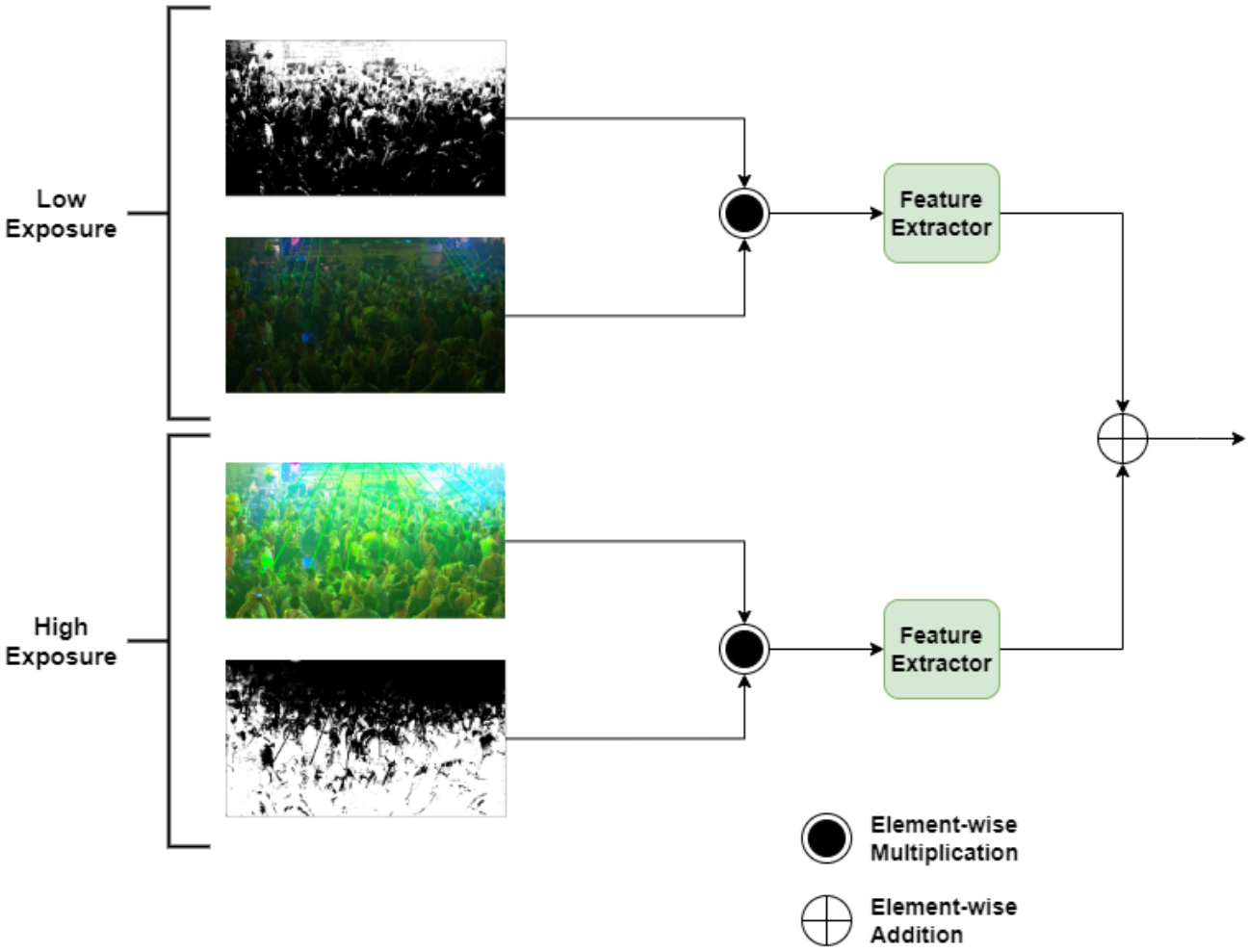

Figure 3.4: Structure of the Visual Attention Module (VAM).

As discussed in Chapter 2, image segmentation is used to assist the model in producing a better image. As shown in Figure 3.4, the input images are first multiplied element-wise by their corresponding masks. This process preserves the regions with more detail while removing those that are overly dark or too bright. The masked images are then fed into the Feature Extractor to extract features. Finally, the extracted features are combined element-wise. The VAM can be formally defined as follows:

$$\text{features}_L = F(\text{multiply}(\text{mask}_L, I_L)) \tag{3.8}$$

$$\text{features}_H = F(\text{multiply}(\text{mask}_H, I_H)) \tag{3.9}$$

$$V = \text{add}(\text{features}_L, \text{features}_H) \tag{3.10}$$

where F is the feature extraction function, and V is the output feature of the VAM.

Spatial Alignment Module

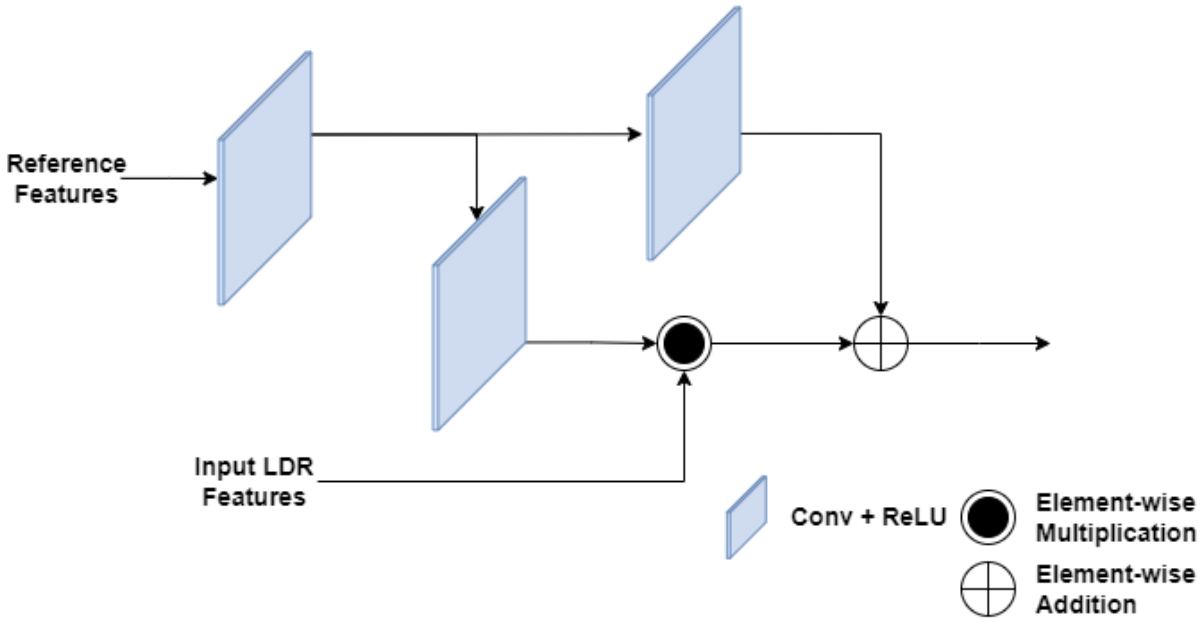

Figure 3.5: Structure of the Spatial Alignment Module.

Since the input LDR images are not aligned, the extracted features from the LDR images (without gamma correction) are fed into an *ad hoc* module for alignment. To achieve this, we use the same Feature-Alignment Module as described in [95]. As shown in Figure 3.5, a Conv + ReLU is first applied to the Reference Features, denoted as Ref_1. Next, a Conv + ReLU is applied to Ref_1, and the result is element-wise multiplied by the input LDR features, denoted as M_i (for $i = 1, 3$). Finally, another Conv + ReLU is applied to Ref_1, and the output is element-wise added to M_i. The operation in this module can be formally written as follows:

$$\text{Ref}_1 = \text{ReLU}(\text{Conv}(\text{ref features})) \tag{3.11}$$

$$M_i = \text{multiply}(\text{ReLU}(\text{Conv}(\text{Ref}_1)), \text{inp features}_i) \tag{3.12}$$

$$\text{out}_i = \text{add}(\text{ReLU}(\text{Conv}(\text{Ref}_1)), M_i) \tag{3.13}$$

Attention Module

The Attention Module is similar to [95] in terms of structure, but it differs in details. As shown in Figure3.6, feature maps are produced for low- and high-exposure images and merged

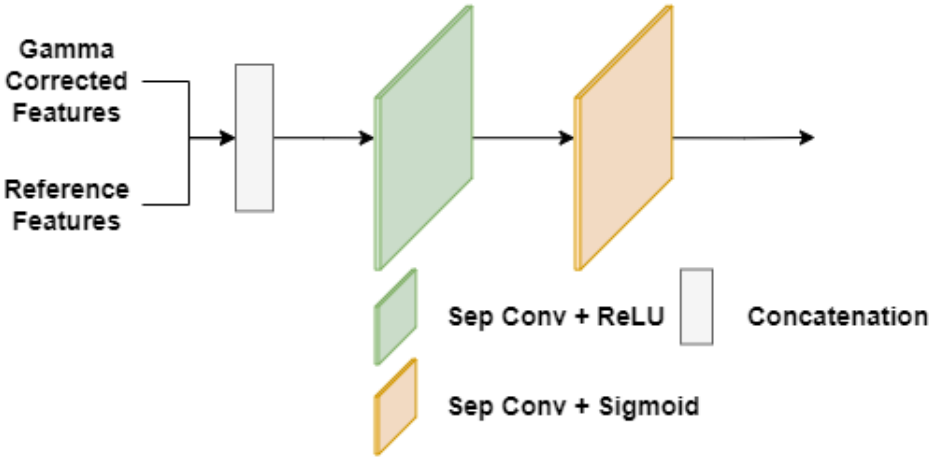

Figure 3.6: Structure of the Attention Module.

with the reference image as guidance. These are concatenated after feeding the features of gamma-corrected images with the reference image. Afterward, SepConv + ReLU and SepConv + Simgoid operations are applied to them. The module can be considered as follows:

$$R_i = \text{ReLU}(\text{SepConv}(\text{concat}(f_i, f_r)) \quad \text{for } i = 1, 3 \tag{3.14}$$

$$S_i = \text{Sigmoid}(\text{SepConv}(R_i)) \tag{3.15}$$

where f_i and f_r are the features of the gamma-corrected and reference images, respectively.

Reconstruction

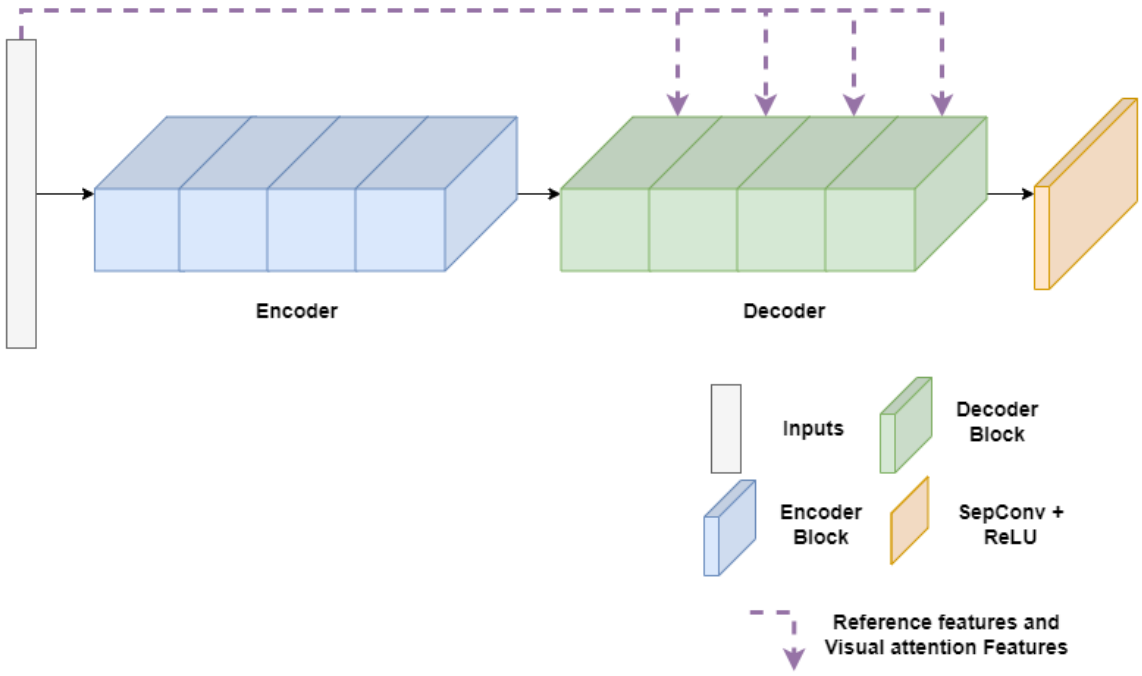

Figure 3.7: The overall Scheme of the Reconstruction stage.

All the extracted features from the modules are concatenated and fed into the reconstruction stage. As shown in Figure 3.7, the input is merged with the help of four encoder blocks, and new features are produced. Next, each decoder block receives features from the encoder, as well as features from the reference image and VAM. Finally, a SepConv + ReLU is applied to produce the output of the stage.

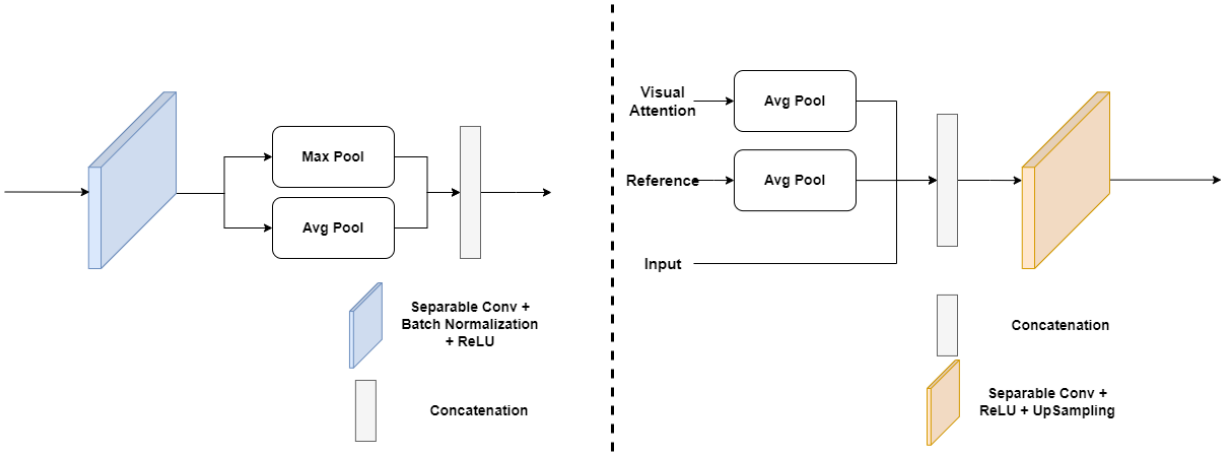

Figure 3.8: Structure of the the encoder (**left**) and the decoder (**right**) blocks.

Each encoder block (Figure 3.8, left) initially applies SepConv, Batch Normalization, and ReLU layers to the inputs. Afterward, similar to the Feature Extraction Module, Max and AVG Pooling operations are used. Finally, the results are concatenated and sent to the next block.

Moreover, each decoder block (Figure 3.8, right) consists of three inputs: features from the VAM, features from the reference image, and the output of the previous block. First, AVG pooling is applied to the first two inputs to make them the same size as the output of the previous block, and then they are concatenated. Finally, SepConv + ReLU and Upsampling are applied, respectively.

Refinement

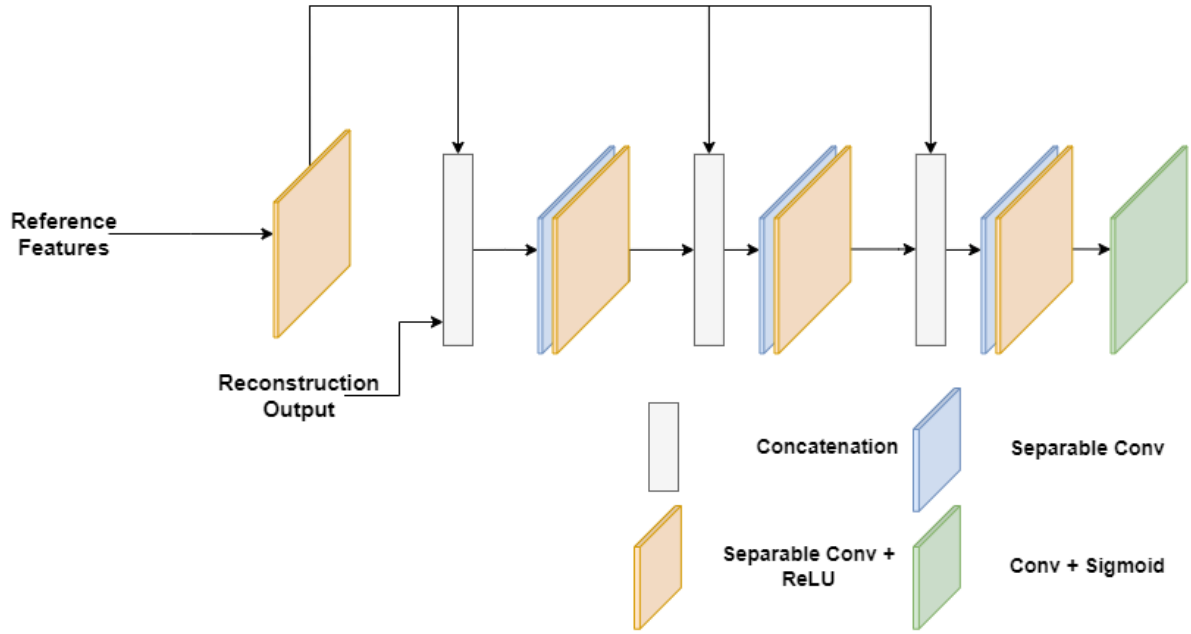

Figure 3.9: Structure of the Refinement Stage.

Unfortunately, the output of the reconstruction stage may have blurry, saturated, or dark areas. Therefore, a refinement section has been added to address these potential issues with the help of the reference image's features.

As Figure 3.9 illustrates, SepConv + ReLU is applied to the features of the reference image to reduce the number of feature maps. Furthermore, after concatenating the inputs, SepConv and SepConv + ReLU are applied, respectively. This process is repeated twice, and eventually, $Conv + Sigmoid$ is applied to produce the final image in Sigmoidal space. The process in the Refinement stage can be represented in pseudo-code, as shown in Algorithm

1.

Algorithm 1 Pseudo-code for the Refinement Stage.

Inputs: The output of the Reconstruction stage (denoted as $Reconstruction_o$) and the extracted features of the referenced image (f_r).

Output: The final image in the Sigmoidal Space.

$\hat{f}_r = \text{ReLU}(\text{SepConv}(f_r))$

$i \leftarrow 0$

while $i < 3$ **do**

  if $i == 0$ **then**

    $c \leftarrow concat(Reconstruction_o, \hat{f}_r)$

    $x \leftarrow \text{ReLU}(SepConv(SepConv(c)))$

  else

    $c \leftarrow concat(x, \hat{f}_r)$

    $x \leftarrow \text{ReLU}(SepConv(SepConv(c)))$

  end if

  $i \leftarrow i + 1$

end while

$out \leftarrow Sigmoid(Conv(x))$

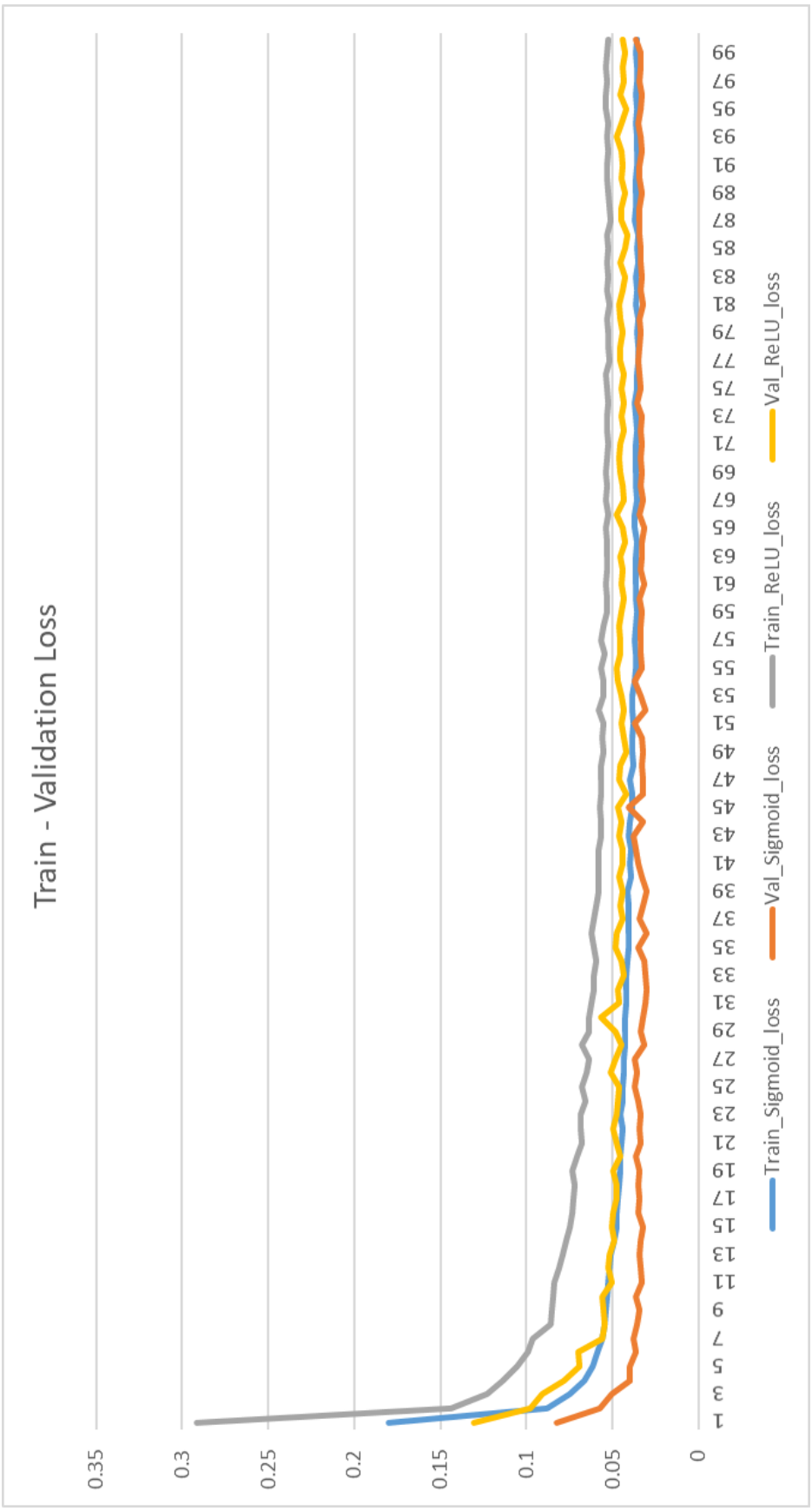

Figure 3.10: Training and Validation loss in Sigmoidal and HDR Spaces.

As shown in Algorithm 1, the first two lines display the inputs and outputs of the refinement stage. Moreover, the *concat*, $SepConv$, and $SepConv + ReLU$ steps can be considered a block of the stage, which is applied three times. The first block receives $reconstruction_o$ and $\hat{f}_r$ as inputs (lines 6-8). However, the output of the previous block and $\hat{f}_r$ are fed to the subsequent blocks.

Note that, in this research, the ground truth images are mapped from HDR Space into sigmoidal space. Based on our experiments, transforming the values into sigmoidal helps the network converge more effectuvely (see Figure 3.10 for a comparison of training and validation loss in Sigmoidal and HDR space). The reason for this transformation is that the values in HDR space are too large, and a model with a low number of parameters is unable to learn to produce an HDR image correctly. Conversely, by mapping the values to sigmoidal space, they are constrained between 0 and 1, which helps the proposed model to learn the data more efficiently.

3.4 Experiments and Results

3.4.1 Datasets

Standard benchmark datasets were used to test and validate the proposed method. The primary dataset is the NTIRE dataset, which was collected for the HDR Imaging Challenge (NTIRE) [68, 62] (2.3.1). In this research, we randomly selected 200 images from the training set as a test set and trained the model with approximately 1300 pairs.

In addition to the NTIRE dataset, we also tested our method on two other datasets: Kalantari et al. [30] and Hu et al. [28]. Both datasets contain dynamic scenes with large motions between the medium, low-, and high-exposure images. The Kalantari et al. [30] dataset was created by capturing static scenes and introducing motion either by having a human actor move or by shifting the camera position between the acquisitions of the different LDR images. The Hu et al. [28] dataset consists of a set of sensor-realistic synthetic images generated using Unreal Engine, which were then calibrated to match the color gamut of an actual sensor.

3.4.2 Implementation Details

Table 3.1: Key highlights of the training and validation settings for the proposed method.

Dataset	NTIRE Challenge	
Optimizer	Adam Optimizer	
Initial LR	0.001 with LR decay	
Batch Size	Train	Validation
Input Size	16	2
Augmentation	256x256	1920x1088
	True	False
Epoch	100	
Loss	Mean Absolute Error (MAE)	

The key highlights of the model are summarized in Table 3.1. The model weights were initialized randomly, and no pre-trained weights were used. Finally, the details of the proposed method will be discussed in the following subsections.

Loss function

Among various potential loss functions, the MAE loss was selected for training the model. This decision is based on the experimental findings reported in [98], particularly in the closely related task of image denoising. In that study, the authors demonstrated that three loss functions - MS-SSIM+MAE, MAE, and MS-SSIM - consistently performed best. In this work, MAE is preferred due to its computational simplicity, making it both practical and effective for the proposed model.

Operatively, the difference lies in the fact that the ground truth is first mapped to the Sigmoidal Domain. MAE is then computed in Sigmoidal space between the ground truth and the model output.

$$GT_n = sigmoid(GT) \tag{3.16}$$

$$L(\hat{y}, GT_n) = |GT_n - \hat{y}| \tag{3.17}$$

where GT_n is the ground truth image in the sigmoidal domain, and L represents the loss between ground truth and the output.

Furthermore, the inverse Sigmoid function is used to remap the output to HDR space after the model has been trained in the sigmoidal space. The inverse sigmoid is defined as

follows:

$$HDR = \log(\frac{(\hat{y})}{1-\hat{y}}) \tag{3.18}$$

where HDR is the resulting image in HDR space and $\hat{y}$ is the image in the sigmoidal domain.

Training

Flipping the images vertically or horizontally is also used as an augmentation method during training. Moreover, before feeding the images to the model, they are resized to 256x256. The reason for doing so instead of producing patches is that some generated patches from the masks may be totally black or completely white, which causes the model to pay less attention to images with low-exposure.

Additionally, the batch size and the number of epochs are set to 16 and 100, respectively. In this article, the Adam optimizer with an initial learning rate of 0.001 is used, and the learning rate will be reduced by a factor of 0.1 if the validation accuracy does not improve. Finally, the whole model is implemented in the TensorFlow (Keras) framework and is trained on a DGX-A100 GPU.

Validation

The images are first padded from 1900x1060 to 1920x1080 and then fed to the model without any augmentation methods during validation.

3.4.3 Evaluation Metrics and Comparison

Quantitative Comparison

As Table 3.2 demonstrates, the results in this chapter are compared with the SOTA methods using $PSNR$ and $SSIM$ in HDR and Tone-mapped domains. The $\mu-PSNR$ and $\mu-SSIM$ refer to the tone-mapped versions, where the images were tone-mapped in $\mu-law$. Moreover, in addition to PSNR and SSIM, the results are compared with the SOTA methods using LPIPS [97], delta-E, $GMACs$, and the number of parameters. Learned Perceptual Image Patch Similarity (LPIPS) is a metric that computes the perceptual similarity of two images using a neural network. Delta-E is a metric that calculates the color difference between two images.

Table 3.2: Comparison with the SOTA methods, including ours, also considering it without the refinement and segmentation stages as described in Section 3.5. The bold numbers represent the best values, and the underlined ones represent the second best.

Methods	PSNR	μ-PSNR	SSIM	μ-SSIM	LPIPS	delta-E	GMACs	Param. $\times 10^3$
GSANet [41]	36.88	35.57	<u>0.996</u>	<u>0.873</u>	**0.02**	**0.40**	<u>199.38</u>	**80**
DRHDR [45]	38.5	**36.91**	<u>0.996</u>	0.86	0.21	**0.40**	1701.932	1190
Vien et al. [84]	39.44	35.39	0.994	0.837	0.34	<u>0.45</u>	**198.819**	1301
ours	**43.25**	<u>35.86</u>	**0.997**	**0.90**	<u>0.03</u>	0.57	234.107	570
ours-w-r	<u>41.71</u>	35.30	0.993	0.857	0.04	0.51	227.59	567
ours-w-s	40.27	34.99	0.993	0.842	0.05	0.66	223.96	<u>545</u>

As mentioned in [62], the challenge focused on two tracks: Fidelity and low complexity. In the first track, the methods were required to achieve the highest $\mu - PSNR$ while ensuring the $GMACs$ value was less than 200. In the second track, the goal was to reduce the $GMACs$ value to below that of the baseline method, while keeping the $PSNR$ and $\mu - PSNR$ values nearly the same as the baseline method. The proposed method has been compared with GSANet [41], DRHDR [45], and Vein et al. [84]. As can be seen in Table 3.2, the proposed method achieves the highest value in terms of $PSNR$ while having the second highest value in $\mu - PSNR$.

Additionally, all the methods were close in SSIM; however, our method outperformed the SOTA in both SSIM and $\mu - SSIM$. Furthermore, although our result with a value of 0.03 is the second best in LPIPS, it performed the worst in delta-E. On the other hand, Vien et al. [84] had the lowest GMACs value, and GSANet ranked second lowest. Moreover, it is evident that in terms of the number of parameters, GSANet has the lowest, and the proposed method is in second place among the algorithms. Table 3.2 shows that the "ours-w-r" and "ours-w-s" methods refer to our method without refinement and segmentation. Although the number of parameters and the GMACs value in those methods are lower than in the total model, the full model still produces better results in terms of the metrics.

Table 3.3: Comparison between the proposed method in HDR and Sigmoidal Spaces.

Methods	PSNR	Mu-PSNR
Ours (HDR Space)	42.4	35.28
Ours (Sigmoidal Space)	43.25	35.86

Furthermore, for further study, the proposed method was trained and tested in HDR and Sigmoidal Spaces to determine which space is superior for training the model. As shown in Table 3.3, the proposed method in Sigmoidal Space outperformed the algorithm in the HDR domain. Moreover, during training, the model in Sigmoid space converged more quickly than the model in the HDR domain.

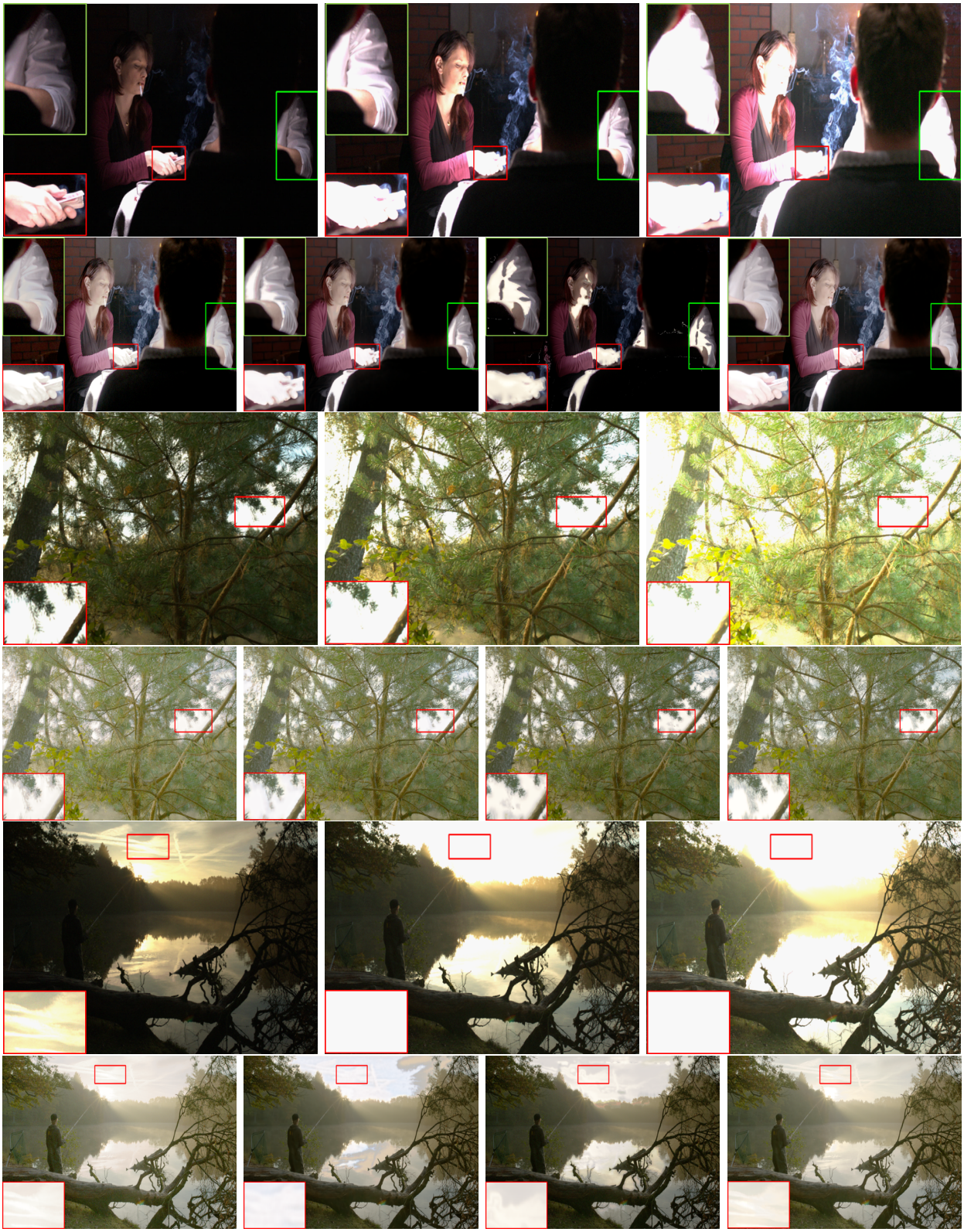

Figure 3.11: Qualitative comparison with the SOTA. The first row of each scene contains low-, medium-, and high-exposure images, respectively. The second row includes the outcomes of ours, DRHDR, Vien et al., and GSANet.

Qualitative Comparison

In terms of qualitative comparison, we used images of the NTIRE [68, 62], Kalantari [30], and Hu [28] datasets. As seen in Figure 3.11, our produced images performed better in terms of image reconstruction compared to the DRHDR and Vien et al. methods. More specifically, Figure 3.11 demonstrates the results of ours, DRHDR [45], Vien et al. [84], and GSANet [41]. As can be seen, the output of Vien et al. in the first scene shows distortion in the bright areas, and it is evident that the algorithm cannot restore the details from these areas correctly. Furthermore, there is some degradation in the dark regions as well. Moreover, although DRHDR performed well and reconstructed both areas, it could not capture the details in over-saturated regions. For instance, in the two red and green boxes, the model failed to reconstruct the details of the hands and the shirt, whereas the proposed method produced more detailed results in these regions. The image generated using the GSANet method shows significant information and is almost similar to ours. More precisely, although both methods could reconstruct the shirt nicely, GSANet captured more detail in the hand than our method.

In the second scene, the DRHDR and Vien et al. methods failed to reconstruct the branches visible in the short-exposure image, restoring only part of them. In contrast, the proposed method and the GSANet performed almost equally. Finally, in the last scene, it is clear that the proposed method outperformed the first two algorithms by reconstructing more details in both the dark and bright areas. The details of the sky highlight this improvement.

As further research, we tested the model and the SOTA methods on two additional datasets with much more movement [30, 28]. Unfortunately, since all the models were trained on datasets with low movement, they did not perform as well on these datasets as they did on the NTIRE dataset. Therefore, we only used qualitative results for comparison, as the quantitative results were not as satisfactory as those on the NTIRE dataset. As seen in Figure 3.12 (first scene), three methods, including ours, worked well in addressing the motion problems. However, GSANet encountered a ghosting issue. Similar to the NTIRE dataset, our model reconstructed more local details than the other methods. On the other hand, as shown in the second scene of Figure 3.12, none of the methods could produce images without ghosting problems when there was more motion. The red box in the photos highlights the common area where all methods encountered a ghosting issue.

Furthermore, although segmentation helped the model produce better results, the method might encounter two possible issues. Firstly, due to plausible noise in the input images, segmentation for extracting visible areas may also capture the noise, leading to noisy output images. Secondly, although spatial alignment and attention modules are used to avoid ghosting issues, the output might still experience ghosting if the input images have significant

motion. This is because segmentation is applied to both the low- and high-exposure images, extracting their visible areas. Consequently, some parts of the images may not be properly aligned. Moreover, for future research, we plan to investigate methods to use segmentation while minimizing noise or misalignment.

Figure 3.12: Qualitative Comparison with the SOTA: The first row of each scene contains low-, medium-, and high-exposure images, respectively. The second row includes the outcomes of ours, DRHDR, Vein et al., and GSANet, respectively. The first and second scenes are taken from [28] and [30], respectively.

3.5 Ablation Study

In this chapter, we proposed a model that included several stages: Attention, Reconstruction, and Refinement, each of which played an essential role in this method. Therefore, to demonstrate the importance of each stage, we sequentially removed the VAM and refinement stages, retraining the model each time to compare them with the complete method.

Additionally, we tested our model on two other datasets [30, 28].

3.5.1 Without visual attention module

As mentioned in Section 3.3.3, the VAM module is helpful for improving image reconstruction. To demonstrate this, we kept the refinement stage, retrained the model without the VAM module, and compared the results with those of the main model.

Given the results in Figure 3.13,the segmentation has both benefits and drawbacks. The zoomed-in portions of the images show that the segmentation stage helps the model reconstruct details more effectively. As seen, the model with segmentation better reconstructs the wall and cracks in the ground compared to the model without segmentation. Moreover, although the model successfully handled motion in the first scene of Figure 3.13, it could not resolve the motion problem in the second scene due to the high volume of movement. Unfortunately, by retaining information from each exposure, the VAM module introduces the ghosting problem.

Figure 3.13: Qualitative comparison between the proposed method (on the left) and the proposed method without the VAM module (on the right). The image was acquired from [28].

3.5.2 without refinement

Additionally, as mentioned in Section 3.3.3, the Refinement stage was used to address potential distortions. Therefore, we retained the Segmentation part and retrained the model

without the Refinement stage.

Given the outcomes in Figure 3.14, the first row of each scene contains the input images, while the second row shows the outputs. The outputs on the right illustrate that the model without the Refinement stage distorts both under-exposed and over-exposed areas. More specifically, the box in the first scene highlights that the hair of the person is not well reconstructed and appears noisy, whereas the complete model successfully reconstructs it. Additionally, as seen in the second scene, both models exhibit ghosting problems. Moreover, the model without the Refinement stage suffers from a lack of local details.

Figure 3.14: Qualitative comparison between the proposed method (on the left) and the proposed method without the refinement stage (on the right). The image was acquired from [30].

3.6 Conclusion

In this chapter, we proposed a complete pipeline for HDR imaging that leverages image segmentation. Specifically, we first applied the Otsu method to low- and high-exposure images to identify areas with more details. The input images and segmentation outputs were then fed into the model to generate the HDR image. The results demonstrate that the proposed method outperformed the SOTA techniques and produced more detailed images. However, the model is not without its limitations. In cases of noise or misalignment in the input images, the output may exhibit slight noise or misalignment due to the extraction of areas from the input images. More specifically, the experiments show that the model cannot produce ghosting-free images when the level of motion is high, due to the Segmentation stage. Therefore, future research will focus on addressing these two issues.

Chapter 4

Towards the Development of Explainable Machine Learning Models to Recognize the Faces of Autistic Children

4.1 Introduction

Often, the first step in identifying children with autism involves screening. If the screening result is positive, an interdisciplinary team observes the child and administers a battery of tests to provide a differential diagnosis [31]. Although best practices exist, the process is far from perfect: Screening is often prone to false positives, and diagnosis involves costly, interdisciplinary assessments [40, 96]. Thus, researchers must continue their efforts to identify solutions that facilitate the timely diagnosis of autism. One potential tool to support autism diagnosis is image classification using machine learning. For instance, several studies have explored diagnosing autism with machine learning by classifying MRI brain images [67, 74]. However, one drawback of using MRI images is that the process requires expensive equipment. Additionally, acquiring such data necessitates procedures that may be uncomfortable for autistic individuals.

Recently, some studies have addressed this issue by using machine learning and image classification to identify autistic children based on facial images [2, 9, 23, 82]. In the most accurate example reported in the literature, [9] developed models with an accuracy of 95% in differentiating autistic children from non-autistic children. Although these recent models have demonstrated excellent accuracy, researchers must consider how and why these models produce their results. In other words, current models are like "black boxes" and do not inform practitioners why a facial image is classified as autistic or not. Understanding this

information is known as explainability. Explainability in healthcare is crucial, as practitioners are less likely to use models they cannot explain to their patients [46]. In this study, we explore the fundamental question of whether it is possible to detect signs of autism in children using only a single facial image. To ensure transparency and clinical relevance, we also investigate which facial features deep neural networks rely on for classification. We compare the development of explainable models using two different algorithms to better understand the decision-making process and enhance the interpretability of our findings.

4.2 Proposed Method

4.2.1 Dataset

We used the Kaggle challenge dataset, the Autistic Children Facial Image Data Set, for the current study. This dataset is publicly available and can be freely downloaded [1]. The dataset was created by the challenge developer, who searched for and downloaded images of autistic and non-autistic children from public internet sources. It contains 2,938 facial images, half of which are from autistic children and the other half from non-autistic children. The images vary in background, angle, and facial expression. Since the dataset consists of publicly available images from the internet, the demographic characteristics of the children (both autistic and non-autistic) are unknown. Additionally, there was no community involvement in the reported study.

4.2.2 Procedures

Machine learning typically involves two phases: training and testing. These phases require the data to be divided into three sets: the training set, the validation set, and the testing set. During the training phase, researchers provide the images in the training set to the model so that it can learn to correctly categorize each image. The validation set is used to select the models that produce the best accuracy during training and to set the optimal values for the algorithm's hyperparameters. This process is repeated several times with different hyperparameter values to select the best model. After training, the test set is used to assess the best model's performance during the testing phase. This step is essential to examine the generalizability of the model to images that were never used for training.

In the current study, the training set contained 2,538 images (1,269 per category), while the validation and test sets each included 200 images (100 per category).

Because the dataset contained fewer than 1,500 images per class for training, we leveraged

pre-trained models to facilitate classification. Using pre-trained models allows researchers to utilize the information learned from the initial dataset and fine-tune the models for their specific task. Specifically, our study evaluated five types of Vision Transformer (ViT) models and five ResNet models as pre-trained models (for more details on these models, see [13, 27]). The pre-trained weights for these models were obtained from the ImageNet_V1 dataset. To retrain the pre-trained models on our dataset, the last layers of all models were removed, and a dropout layer along with a linear classification layer with two units was added. The following parameters were used: a batch size of 32, 150 epochs, and a learning rate of .0001 with an Adam optimizer. All models were implemented using the PyTorch framework. We then compared the accuracy of the models and selected the one that produced the highest accuracy for our analyses. The pre-trained model that achieved the best accuracy was ViT_Huge_14.

4.2.3 Analysis

The main objective of this chapter is to use explainable techniques to analyze the best model. Therefore, the methods will be briefly explained. After identifying the best model, we employed two techniques to examine its explainability: LIME [71] and RISE [63].

LIME

In the LIME method, the input data are divided into superpixels, which are then randomly perturbed and fed to the algorithm to understand how the model functions. In other words, the method randomly removes small parts of the image (e.g., the right eye) and feeds the modified image to the model to assess its impact on the model's predictions. A saliency map is generated by identifying the importance of each superpixel. Specifically, if removing a superpixel causes a significant drop in accuracy, that area is assigned higher importance in the saliency map.

RISE

The other method, RISE, generates several random masks and applies them to the input data. The images produced by the masks are then fed to the model, and a confidence score is calculated for each image. Subsequently, the algorithm computes a saliency map based on all the confidence scores. A saliency map is an image that highlights the areas where the model focuses its attention during prediction. In this case, the saliency map indicates which parts of the image the model used to predict the class. Specifically, RISE randomly generates masks, applies them to the image, and feeds the modified images to the model.

Those images that result in higher accuracy are assigned higher importance in the saliency map.

Explainable Metrics

To evaluate which explainable model was more reliable, we used two metrics: deletion and insertion. In the deletion method, the algorithm removes parts of the image based on the explainability at each step. The method identifies the most relevant areas by assessing the removal of sections that cause the largest reductions in accuracy. Each time an area is deleted, the modified image is fed to the model to predict accuracy. The total score is the average accuracy across all steps. By deleting areas, we aim for lower accuracy, so lower accuracy values for deletion indicate more reliable explainable models. In the insertion method, the algorithm first creates a blurry image by applying a Gaussian filter. Areas are then restored individually, with the most relevant areas increasing accuracy the most. Higher accuracy values for insertion indicate more reliable explainable models. Our code is available **here**.

4.3 Results and Discussion

The best model, ViT_Huge_14, achieved an accuracy of 92%, a score of 91% on the AUC metric, a sensitivity of 90%, and a specificity of 92%. Figure 4.1 displays the mean scores of LIME and RISE for the deletion and insertion metrics across all true positive and true negative cases. The left panels of Figure 4.1 show that LIME produced lower scores on the deletion metric, indicating that, on average, it performed better than RISE. The right panels of Figure 4.1 reveal that LIME produced higher scores on the insertion metric than RISE. These results suggest that the LIME algorithm outperformed RISE for both true negative and true positive cases, making it a better tool for explaining classification.

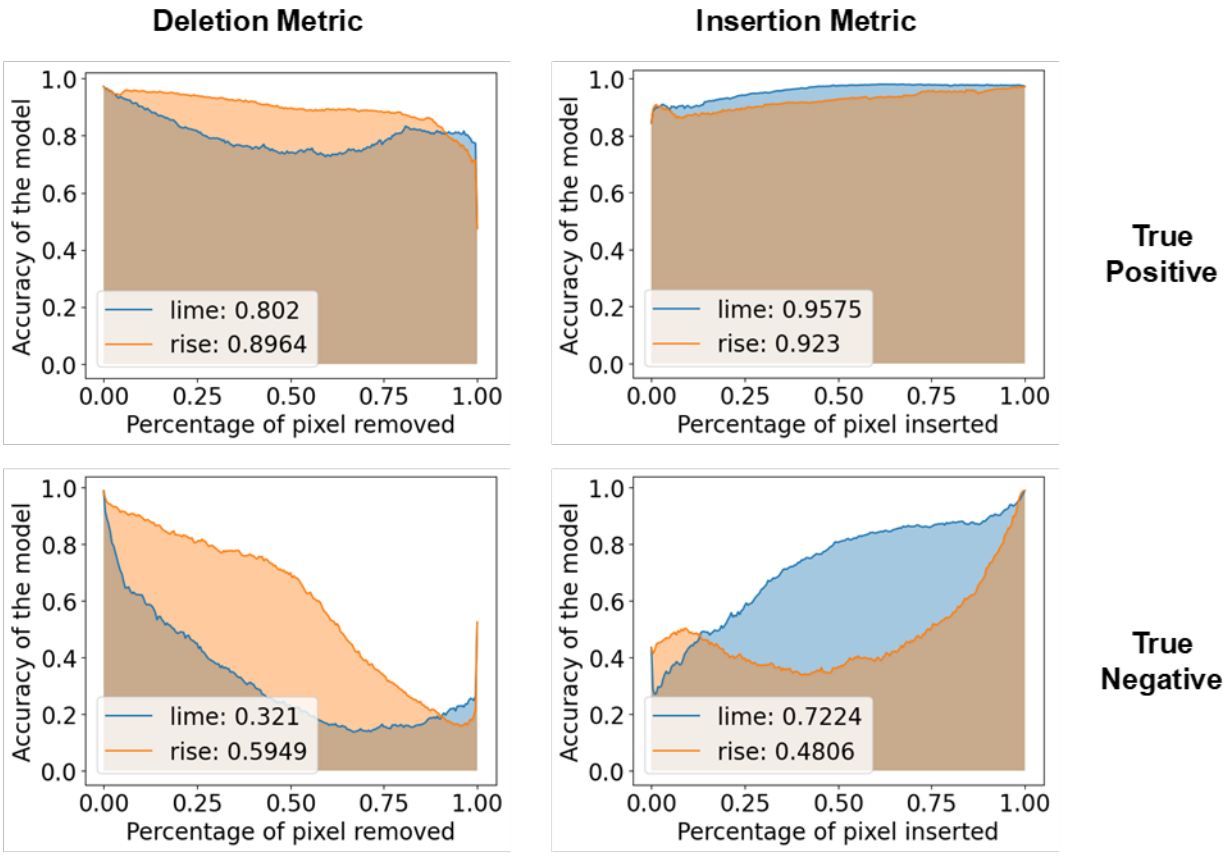

Figure 4.1: Average values for all true positive (upper panels) and true negative (lower panels) samples. The graphs on the left represent the deletion metric, while those on the right represent the insertion metric.

Figure 4.2 presents examples of the explainable images for a true positive case (upper panels) and a true negative case (lower panels) using heatmaps. For the true positive case, the LIME heatmap shows that the model primarily focused on the lower part of the face (more reddish, indicating higher importance), whereas the RISE heatmaps highlight that different areas of the image as important. For the true negative case, the heatmap generated by the LIME method indicates that the model primarily focused on the regions around the eyes and cheeks, while the RISE method emphasized the periphery. These heatmaps demonstrate that the area selected by the LIME model is better defined, which may make the algorithm's results easier to explain. Examples for the other two cases (false negatives and false positives) are provided in the Appendix A.

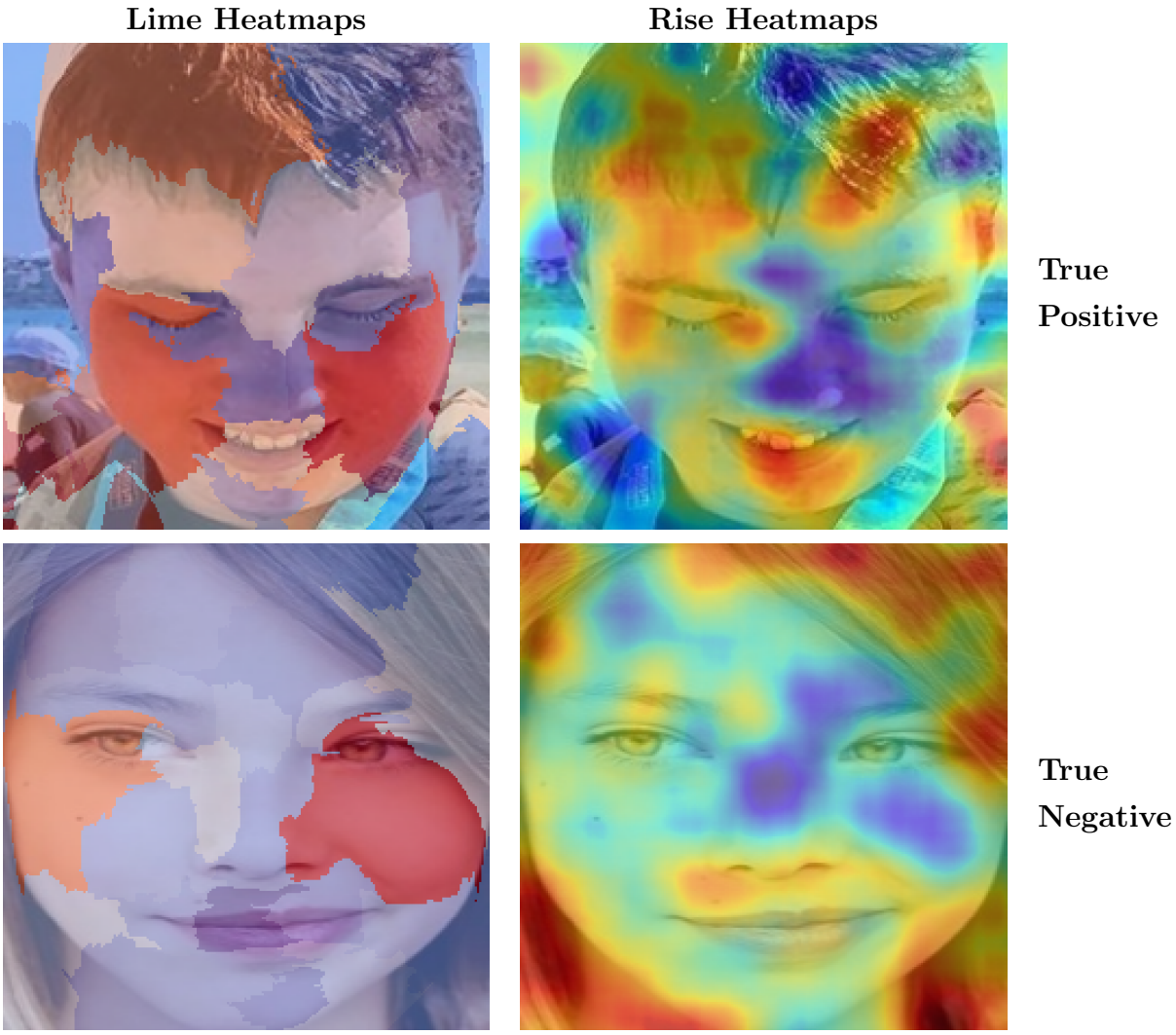

Figure 4.2: LIME (left panels) and RISE (right panels) heatmaps for the true positive case (upper panels) and true negative case (lower panels).

By combining image classification with brief behavioral observations and interviews, researchers may develop faster and more cost-effective alternatives for screening and diagnosis. The results obtained in our study align with those of other recent research papers on the topic, suggesting that autistic children may have subtle facial characteristics that can be identified through machine learning [2, 9, 23, 82]. We extended previous research by comparing two methods to examine the information the models used to identify individuals with autism from facial images. The current study showed that LIME produced more reliable explainable models than RISE. One possible explanation for this observation is that LIME focuses on local interpretability (examining each area separately), whereas RISE emphasizes global interpretability (treating the image as a whole).

Despite its accuracy, we can only recommend using our model once further studies are conducted to address some limitations. First, the small size of the dataset may not fully represent the diversity of facial features, limiting its generalizability. Similarly, the dataset was created by searching for and downloading images from public websites, which may have been biased by the availability of images and the developer's search procedures. As such, the extent to which the photos are representative of the population of autistic and non-autistic children remains unknown. These issues highlight the need for larger, higher-quality, and more diverse datasets to allow for more comprehensive and accurate analyses. Additionally, a more thorough analysis of the ethical and deontological issues associated with facial image classification in autism is necessary before adopting such technology in practice. For example, we argue that image classification should only be one component of a multi-method approach to screening and diagnosis.

Furthermore, using videos instead of photos may provide a richer and more dynamic source of data. Videos may not only help identify individuals with autism but also enable the analysis of their behavior over time, potentially revealing patterns or characteristics not evident in static images. This multi-method approach could significantly advance our understanding and detection capabilities in both autism practice and research.

Chapter 5

Machine Learning to Measure Vocal Stereotypy: An Extension

5.1 Introduction

One of the hallmarks of applied behavior analysis is the repeated measurement of behavior before, during, and after treatment [10]. Although this rigorous approach ensures that the beneficiaries of the science receive adequate treatment, the repeated measurement of behavior by human observers remains a challenging task in both practice and research. For example, parents, teachers, and technicians may struggle to measure behavior consistently while simultaneously implementing interventions with high integrity [6]. To make data collection more manageable in such contexts, practitioners and researchers may use discontinuous analysis methods; however, this approach can still produce inaccurate results [19, 37].

A more rigorous method for assessing complex behavior involves recording the behavior on video for subsequent scoring or assigning an observer exclusively to data collection. However, using human observers to measure behavior from video recordings can be costly, as additional time must be allocated for data collection [14]. Moreover, researchers and practitioners must hire a second observer to assess interobserver agreement, regardless of the data collection method [26]. Given these challenges, scoring a recorded video of one participant may require nearly double the time needed to conduct the session itself. This analysis does not even account for the time involved in recruiting and training observers. Consequently, using dedicated observers can significantly increase the cost of implementing interventions, potentially reducing the available resources for practice and research.

One potential solution to reduce the costs of measuring behavior involves using artificial intelligence, specifically machine learning. A subfield of artificial intelligence, machine learn-

ing trains computer algorithms to identify patterns in different sources of data, such as video recordings [29]. Due to its repetitive nature, one of the first areas of application of machine learning in the behavioral sciences has been the measurement of stereotypy in children with autism [4, 7, 14, 100]. Stereotypy is defined as repetitive and rhythmic patterns of behavior that typically persist without social consequences [70]. These behaviors generally fall into two broad categories based on their topography: motor stereotypy and vocal stereotypy. Motor stereotypy includes repetitive movements and gestures, such as mouthing, hand flapping, and body rocking [24, 80, 78], while vocal stereotypy includes behaviors like unclear and repetitive sounds, repeating lines from TV shows, and laughter that doesn't align with the social context [3, 47]. Although several studies have focused on measuring motor stereotypy using machine learning [20, 69, 76], this chapter will concentrate on vocal stereotypy.

Only a handful of researchers have used machine learning to detect vocal stereotypy [14, 50, 51, 32]. For example, [50] employed a machine learning algorithm to analyze audio files recorded from four children with autism, detecting the presence of vocal stereotypy in 10- to 20-second clips. Their initial method achieved a detection rate ranging from 73% to 93% across participants. In the following year, [51] proposed a convolutional neural network to detect vocal stereotypy using the same samples, with an accuracy of 86%. More recently, [32] proposed a machine learning-based software to assist human observers in assessing the vocal response of children with autism. Using 2,575 samples from 76 children, the software classified vocal responses into seven categories: speech, clapping, echolalia, non-speech, repetitive speech, unusual noises, and pronoun reversal. The best model reached an accuracy of 91%, but it only detected the presence of each behavior, without measuring its duration.

One standard limitation of these previous studies is that they only explored the presence of vocal stereotypy in short video clips. In contrast, single-case designs used to monitor behavior typically involve measuring the duration of vocal stereotypy during longer sessions. As a result, the previous algorithms have limited utility in practice and research within behavior analysis. To address this issue, [14] developed models to measure the duration of vocal stereotypy in children with autism. Their study included over 27 hours of video recordings from eight children with autism, each exhibiting various forms of vocal stereotypy. Although session-by-session correlations showed promising results, the study had several limitations. First, many of their models produced error rates too high for practical use. Second, they did not pre-train or fine-tune their models—best practices in machine learning that could have substantially improved accuracy. Lastly, their analysis relied solely on Mel frequency cepstral coefficients (MFCC; see Method section 5.2.2 for more details), and incorporating additional features, such as the Mel spectrogram, could have enhanced model accuracy. By addressing these limitations, we believe that more accurate models—suitable

for practice and research—can be developed. Thus, the goal of our study was to extend [14] by training and testing novel models on their original dataset.

5.2 Proposed Method

5.2.1 Dataset

We used the same dataset as [14], which included eight children with autism who engaged in vocal stereotypy. Table 5.1 presents the age, gender, number of sessions, total time, and topography of vocal stereotypy for each participant. Each child participated in 6 to 38 sessions, with total session times ranging from 4,015 to 27,448 seconds. In their study, [14] defined vocal stereotypy as "acontextual or unintelligible sounds or words produced by the vocal apparatus of the child" (p. 372). The original authors measured interobserver agreement for 42% of the sessions. The mean second-by-second interobserver agreement was 97% (range: 93% to 99%), and the mean kappa interobserver agreement was 0.87 (range: 0.81 to 0.94).

Table 5.1: Participant Characteristics[14].

Participants	Age	Gender	Number of Sessions	Total Time (s)	Topography of Vocal Stereotypy
Alia	10	F	10	7,091	Humming and unintelligible vocalizations
Billy-Peter	8	M	10	6,909	Monosyllable sounds and acontextual giggling
Dan	11	M	11	7,533	Phrase or word repetitions
Dave	6	M	30	20,756	Humming and unintelligible vocalizations
Emile	7	M	38	27,448	Grunting and unintelligible vocalizations
Matt	5	M	6	4,015	Monosyllable sounds and repetitive singing
Nate	6	M	12	8,351	Phrase or word repetitions
Owen	7	M	25	17,461	Phrase or word repetitions

5.2.2 Feature Extraction

Mel Spectrogram

Deep learning models rarely accept raw audio files directly as inputs. Therefore, we convert these audio files into spectrograms. Spectrograms are generated by applying Fast Fourier Transform (FFT)s to audio waves, providing a visual representation of the audio. Since spectrograms are image-like representations of audio, they can be easily fed into Convolutional Neural Network (CNN) and ViT-based networks. FFT decomposes the signal into its constituent frequencies and displays the amplitude of each frequency present in the signal.

A spectrogram splits the signal into smaller time segments and then uses the equation below (Eq. 5.1) to apply the FFT to each segment, determining the frequencies present in

that segment.

$$X(k) = \sum_{n=0}^{\infty} x(n)w(n)e^{-j2\pi kn/N} \tag{5.1}$$

In this equation, $X(k)$ represents the frequency components of the signal, n is the time domain sample, and $w(n)$ is the window function for segmenting the signal.

The FFT for all segments are then combined into a single plot, where the y-axis represents the frequency and the x-axis represents time. Different colors in the plot indicate the amplitude of each frequency; brighter colors represent higher energy levels in the signal.

Since humans do not perceive frequencies linearly and are more sensitive to a narrower range of frequencies and amplitudes, we use a Mel scale to scale the frequencies logarithmically. This adjustment aligns the representation with human auditory perception by giving more resolution to lower frequencies. The Mel scale is computed using the following formula (Eq. 5.2):

$$m = 2595 \log_{10}(1 + \frac{f}{700}) \tag{5.2}$$

where m is the Mel Frequency and f is the actual frequency in Hz. This formula translates the actual frequencies to the Mel scale.

Next, we compute the power spectrum for each Mel-frequency component. Finally, the power values are converted to decibels (dB) using a logarithmic scale, reflecting how we perceive changes in loudness. This is done using the following equation (Eq. 5.3):

$$dB = 10 \log_{10}(Power) \tag{5.3}$$

where the "power" refers to the power value of the signal.

MFCC

MFCC provides a compact representation of the spectral properties of an audio signal. The process of calculating MFCC is quite similar to that of the Mel Spectrogram. The audio signal is first divided into smaller time segments, and the FFT is applied to each segment to convert the time-domain signal into the frequency domain. Afterward, the frequency components are mapped onto the Mel scale, which aligns with human auditory perception. This mapping is done using the Mel Filter Bank, as shown in Equation 5.4:

$$S_m = \sum_{k=f_m(1)} f_m^{(2)} P(k) \cdot H_m(k) \tag{5.4}$$

where $H_m(k)$ is the mth Mel Filter and $f_m^{(1)}$ and $f_m^{(2)}$ are the filter boundaries in the frequency domain.

Following this, the logarithm of the Mel-filtered power values is computed to compress the dynamic range of the frequencies. To finally obtain the MFCCs, a Discrete Cosine Transform (DCT) is applied to the log Mel spectrum using eq. 5.5. These coefficients capture essential features of the signal.

$$MFCC(n) = \sum_{m}^{M-1} log(S_m).cos[n(m - \frac{1}{2})\frac{\pi}{M}] \tag{5.5}$$

In machine learning, the first step involves extracting features that the algorithm uses to detect vocal stereotypy. In this study, both the Mel spectrogram and the MFCC of the audio recordings were used. In simple terms, the Mel spectrogram represents the amplitude (or loudness) of sound over different frequency bands (based on human hearing) across time. The MFCC, on the other hand, is a transformed version of the Mel spectrogram, using a function called DCT to extract a series of coefficients. These coefficients contain less "information" than the original spectrogram but allow a more compact representation with fewer spectral features.

In the work of [14], a simple procedure was followed where the MFCC values were represented in numerical format. For this study, we extracted features from the audio recordings using both the Mel spectrogram and the MFCC. These features were treated as "images" of the audio clips, making it possible to use powerful image classification algorithms to improve the models. To achieve this, each recording was divided into smaller audio files, each lasting one second, and the Mel spectrogram and MFCC were extracted in an image-based format.

The audio files had a sample rate of 22,050 Hz, meaning each second contained 22,050 samples. To extract features from these audio files using the Mel spectrogram and MFCC, the window size was set to 1024, with a hop length of 128. As a result, approximately 87 features were extracted every 0.012s, or roughly 11.6 ms.

Figure 5.1 and 5.2 below present examples of the extracted features from the audio recordings using the Mel spectrogram and the MFCC, respectively.

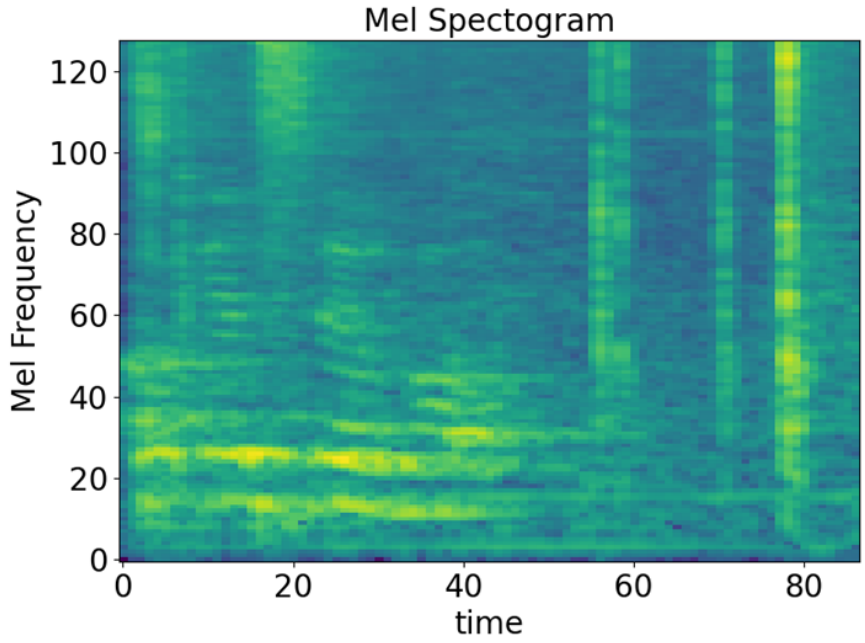

Figure 5.1: Sample of a Mel Spectrogram. The x-axis represents time, and the y-axis represents the feature values. Darker colors indicate lower values, while brighter colors represent higher values.

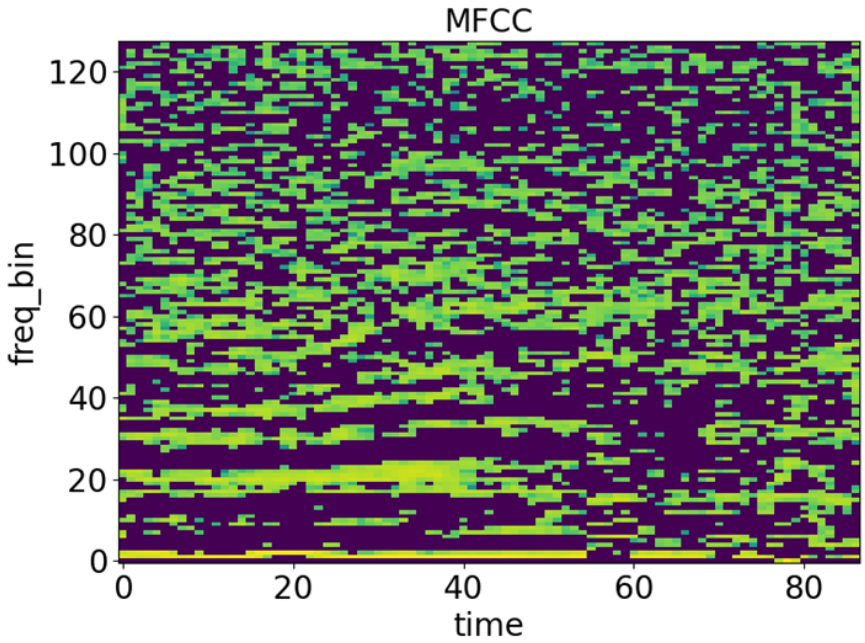

Figure 5.2: Sample of an MFCC. The x-axis represents time values, and the y-axis represents the feature values. Darker colors indicate lower values, while Brighter colors represent higher values.

5.2.3 Method

The choice of model for our study was divided into three distinct stages. In the first stage, we conducted several comparisons and initial experiments using CNNs and ViTs. The results showed that ViTs performed better than CNNs, prompting us to focus on ViTs for the second stage of our research. During this stage, we fine-tuned various ViT models. Ultimately, we discovered that the XCiT transformer [16] outperformed other models in terms of performance, which led us to use the XCiT model for our main experiment.

We applied an XCiT transformer [16] to train our audio data samples in the current study. ViTs excel at extracting global context of data, which enables them to learn complex relationships between features. To better understand how ViTs work, we will first explain the key steps involved in their operation.

The ViT divides images into square patches. Then, the patches are linearly transformed into vectors using a learnable linear layer projection. These vectorized patches are called tokens. Because ViT models do not know which token belongs to which part of the image, positional embedding is added to tokens. Positional embedding is a technique that provides the model with information about the relative order of the input patches.

The central component of ViTs is the encoding layer, which is the part that the model learns from the input tokens. It includes multi-head self-attention and feed-forward networks. Self-attention extracts the relationships between different input tokens. More specifically, the mechanism examines interactions between tokens (regardless of their distances) to capture any dependencies between them. This process can be performed multiple times in parallel, with each instance referred to as a "head." Each head focuses on different types of information using different learned linear transformations. Finally, the results from the heads are provided to a feed-forward neural network as extracted features to classify the output. A feed-forward network is a simple network that connects nodes from one layer to the next. Information flows from the input nodes to the output nodes through any hidden layers.

The problem with ViT models is their quadratic complexity (the time required to run the algorithm increases rapidly as the size of the model grows), which is caused by the self-attention mechanism. Therefore, researchers have replaced self-attention with cross-covariance attention (XCA) [16]. The difference between self-attention and XCA is that the latter operates across channels rather than tokens. In regular ViT models, each token must attend to all other tokens, making the process complex. In XCA, each feature channel attends to different feature channels at the same spatial location. This transformation reduces the model's complexity.

5.2.4 Pre-training

A considerable amount of data is needed to train a deep neural network. [14] should have pre-trained their models, but best practices in machine learning typically involve training the model on a larger dataset and then retraining it with new data. In the pre-training phase, researchers must rely on a large dataset to help the model learn general features and representations. Hence, we first pre-trained the model using an audio dataset called UrbanSound8K [73] (which can be downloaded from **here**), containing 8,732 labeled audio files from 10 different classes. The test set included 20% of the data during the pre-training procedure, while the remaining data was used for training. Table 5.2 presents the values of the hyperparameters used for pre-training. These values are the same as those used during training. The model was implemented using the PyTorch framework.

Table 5.2: Hyperparameter Values for the Algorithm.

Hyperparameter	Value
Optimizer	AdamW
Loss Function	Binary Cross-Entropy
Epochs Num	25

5.2.5 Procedure

Similarly to [14], we divided our study into three analyses: between-participant, within-participant, and hybrid. In the first analysis, we left one participant out for the testing set and trained our machine learning model on the remaining data. For the second analysis, our study focused on one participant at a time for both the training and testing phases. For the final analysis, we combined data from the target participant and the other participants for training and used the rest of the target participant's data for testing. Each procedure is explained in more detail in the sections below.

Between-Participant Analysis

The between-participant analysis aims to determine whether the model can predict the duration of vocal stereotypy for a participant that the model has never seen (i.e., generalization). This analysis is the most challenging because participants typically engage in different topographies of vocal stereotypy. We trained the models on seven participants while keeping one participant for testing. Hence, the analysis resulted in eight models (one per participant). Our analyses involved a k-fold cross-validation algorithm during the training process,

with one participant in the validation set and six participants in the training set. We selected one participant for the validation set to evaluate the model's performance on data not seen during the training procedure. As indicated in Table 5.2, the maximum number of epochs (iterations) was 25. Across each iteration, we checked the kappa value (see Outcomes) on the validation set. We kept only the best model (i.e., the one that yielded the highest kappa value during validation). To measure generalization, each model was tested on the participant who was not included in the training and validation sets. Each model was trained six times with a different participant in the validation set, so our results present the mean outcomes for each participant.

Within-Participant Analysis

Our second analysis focused on one participant at a time. The goal was to train the model on several sessions of the participant and predict the vocal stereotypy of new sessions that the model had never encountered. During training, we excluded one session for the test set and used the remaining sessions for the training and validation sets. The analysis involved four folds: one was used as the validation set, and three were set aside for the training set. Because Matt had a smaller number of sessions compared to the other participants, his data were split into three folds instead. As before, we kept only the model from the epoch that produced the highest kappa on the validation set. Furthermore, our code shuffled the sessions and repeated the cross-validation twice to ensure the model was not biased by any fold. As the analysis was repeated twice, the results section presents the mean outcomes for each session.

Hybrid Analysis

The third analysis combined the between- and within-participant approaches. We excluded one session of the target participant from the training set for the test set. In contrast, the training and validation sets included the remaining sessions from that participant and data from other participants. To balance the dataset, our code randomly selected data from the other participants so that the between- and within-participant data were equal. Similar to the within-participant analysis, our cross-validation involved four folds with two repetitions. This analysis used one fold for validation and three for training. The other procedures remained consistent with the within-participant analysis.

Outcomes

To compare our results with [14], our code computed the same outcome measures on the test set: accuracy, the kappa statistic, and the session-by-session Pearson correlation. Since the rank order of the sessions is essential when analyzing single-case graphs, we also included the Spearman correlation.

5.3 Results and Discussions

Table 5.3 compares the results of the between-participant analyses from our models and those produced by [14]. For the kappa value, the current study outperformed the previous study for all participants, with values near or above 0.50 for six of the eight participants. Additionally, all models produced correlations near or above 0.90, except for Dave. In contrast, only three of eight participants in [14] achieved correlations near or above 0.90. Figure 5.3 shows that the percentages detected by each of our models closely matched those recorded by the human observer. Table 5.4 presents the results of our within-participant analysis. Our models outperformed those reported by [14] on all metrics for each participant. The models produced kappa values near or above 0.70 for four of the eight participants, and the Pearson correlation values were equal to or greater than 0.95 for all participants. Figure 5.4 presents the correspondence between the machine and human measures in graphical format. Both approaches produced closely matched results. For the hybrid analysis, Table 5.5 shows that the new models produced better predictions than those reported by [14]. The kappa value was above 0.50 for all participants, and the Pearson correlations were above 0.90. Figure 5.5 demonstrates how the values predicted by machine learning followed patterns consistent with those produced by a human observer. Our new models outperformed those produced by [14] on all metrics. Moreover, the kappa values were lowest for the between-participant models, which was consistent with expectations and with the findings of [14]. Hybrid and within-participant models typically produce better results, but they require more effort and expertise to train in both practice and research. Interestingly, nearly all of our models produced correlations similar to those between continuous and discontinuous measurement methods ([37]). In other words, [37] reported correlation coefficients within the range observed in the current study. Since behavior analysts already tolerate this range of error when using discontinuous methods, adopting our models may prove useful.

Table 5.3: Comparison of Accuracy, Kappa, and Correlation in the Between-Participant Analysis Between the Models of the Current Study and Those Developed by [14].

Current Study					**Dufour et al [14]**		
Participants	Accuracy	Kappa	Pearson Correlation	Spearman Correlation	Accuracy	Kappa	Pearson Correlation
Alia	0.87	0.49	0.88	0.82	0.79	0.30	-0.12
Billy Peter	0.91	0.51	0.93	1.00	0.89	0.50	0.88
Dan	0.84	0.45	0.98	0.9	0.75	0.29	0.30
Dave	0.82	0.59	0.64	0.88	0.79	0.50	0.82
Emile	0.93	0.74	1.00	1.00	0.90	0.66	0.86
Matt	0.82	0.60	0.99	1.00	0.78	0.49	0.97
Nate	0.74	0.42	0.99	0.97	0.71	0.33	-0.12
Owen	0.87	0.66	0.99	0.97	0.83	0.52	0.80

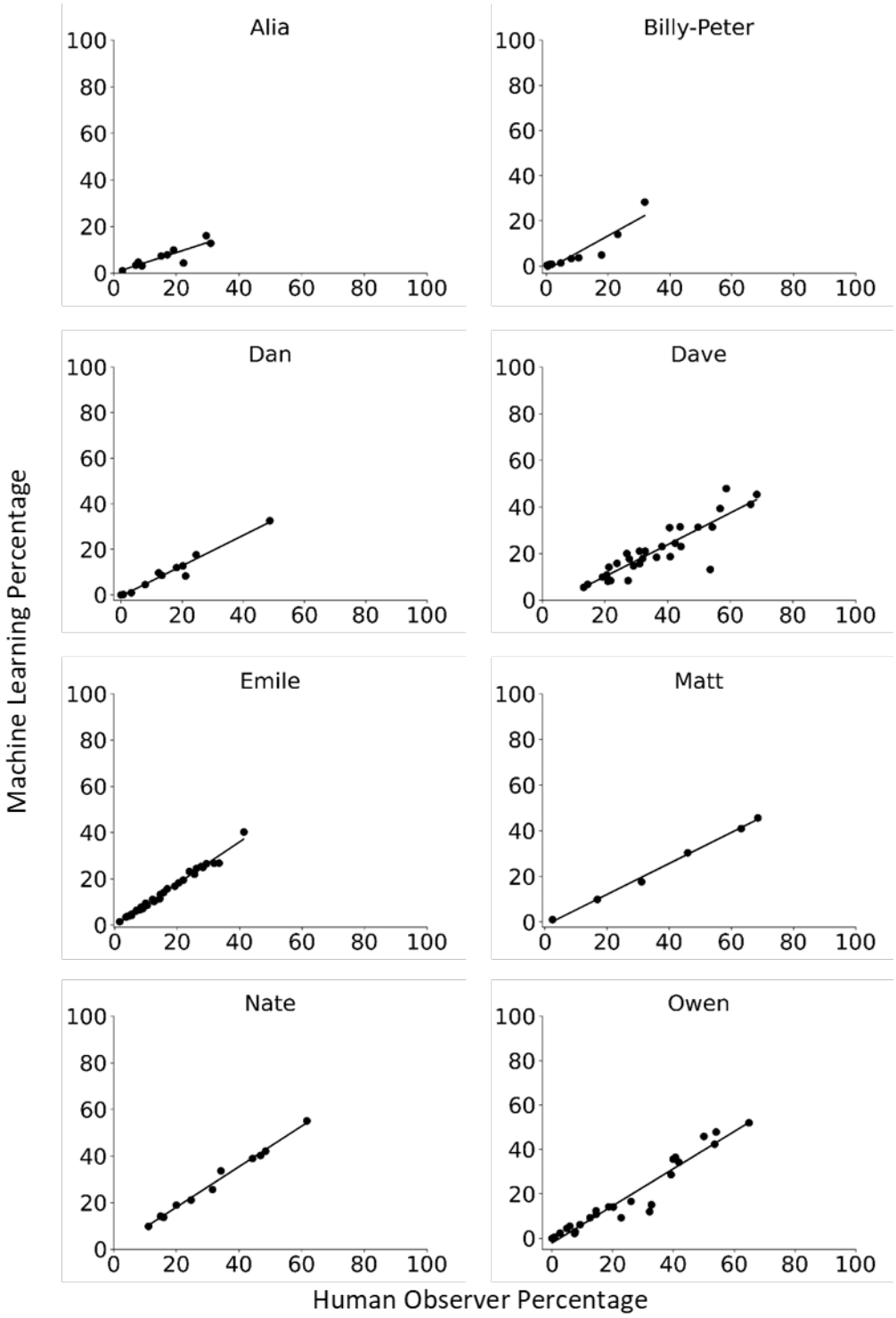

Figure 5.3: Between-Participant Analysis: Correlation Between Percentages Measured by the XCiT Model and the Human Observer Across All Sessions for Each Participant.

Table 5.4: Comparison of Accuracy, Kappa, and Correlation in the Within-Participant Analysis Between the Models of the Current Study and Those Developed by [14].

Current Study					**Dufour et al [14]**		
Participants	Accuracy	Kappa	Pearson Correlation	Spearman Correlation	Accuracy	Kappa	Pearson Correlation
Alia	0.95	0.80	1.00	1.00	0.91	0.67	0.58
Billy Peter	0.91	0.48	0.96	1.00	0.91	0.25	0.93
Dan	0.86	0.45	0.95	0.92	0.79	0.23	0.34
Dave	0.87	0.71	0.95	0.95	0.83	0.60	0.66
Emile	0.96	0.81	0.99	0.99	0.94	0.75	0.97
Matt	0.86	0.69	0.99	1.00	0.80	0.43	0.96
Nate	0.79	0.52	0.97	0.95	0.74	0.34	0.33
Owen	0.87	0.64	0.99	0.98	0.86	0.40	0.88

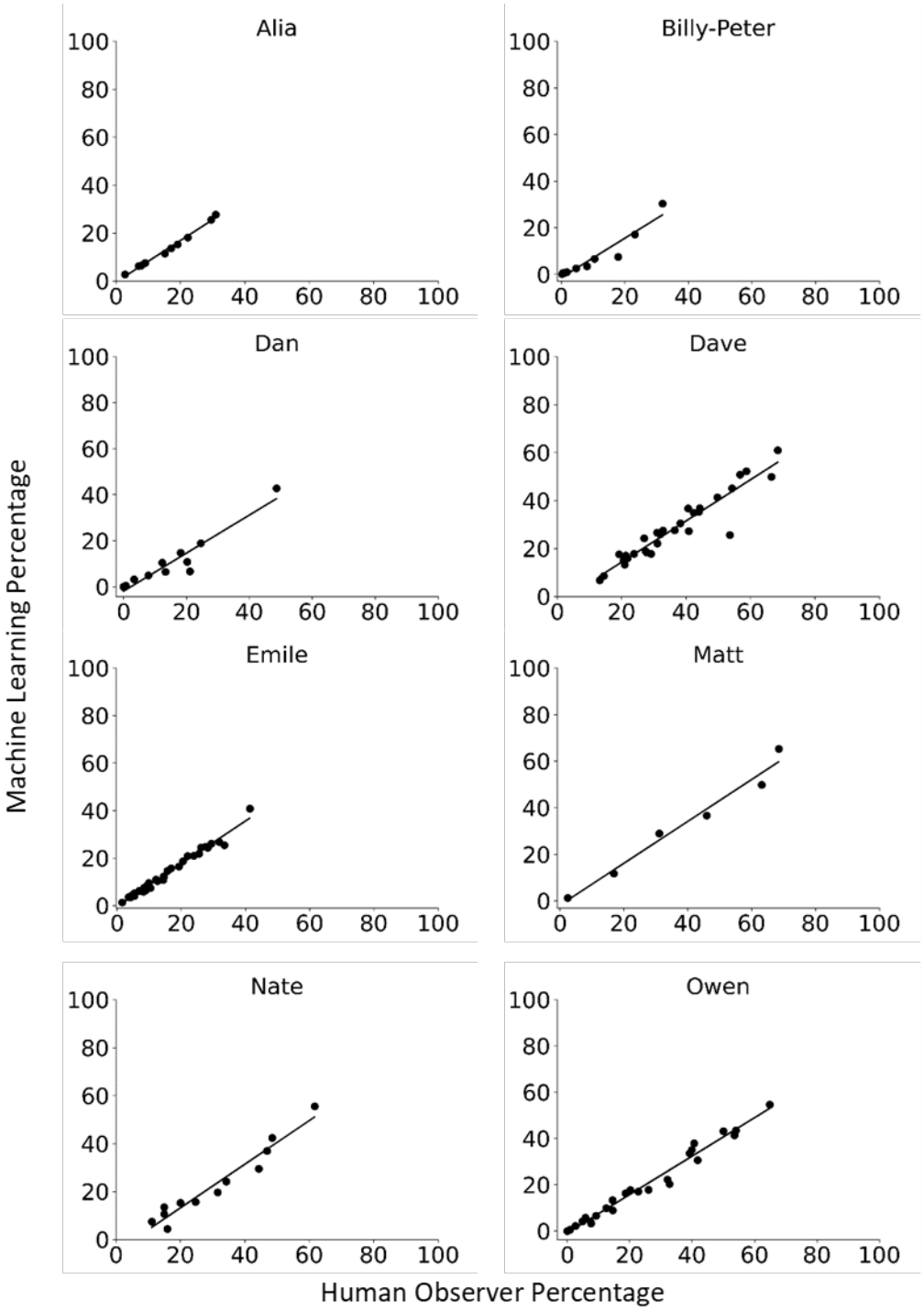

Figure 5.4: Within-Participant Analysis: Correlation Between Percentages Measured by the XCiT Model and the Human Observer Across Sessions for Each Participant.

Table 5.5: Comparison of Accuracy, Kappa, and Correlation in the Hybrid Analysis Between the Models of the Current Study and Those Developed by [14].

Current Study					**Dufour et al [14]**		
Participants	Accuracy	Kappa	Pearson Correlation	Spearman Correlation	Accuracy	Kappa	Pearson Correlation
Alia	0.93	0.72	0.94	0.92	0.92	0.60	0.79
Billy Peter	0.92	0.55	0.94	0.94	0.91	0.23	0.87
Dan	0.86	0.50	0.94	0.86	0.83	0.24	0.20
Dave	0.83	0.61	0.93	0.9	0.83	0.57	0.84
Emile	0.95	0.80	0.99	0.99	0.95	0.74	0.97
Matt	0.85	0.66	0.99	1.00	0.78	0.41	0.98
Nate	0.82	0.56	0.99	0.93	0.73	0.31	0.08
Owen	0.89	0.70	0.98	0.99	0.85	0.45	0.88

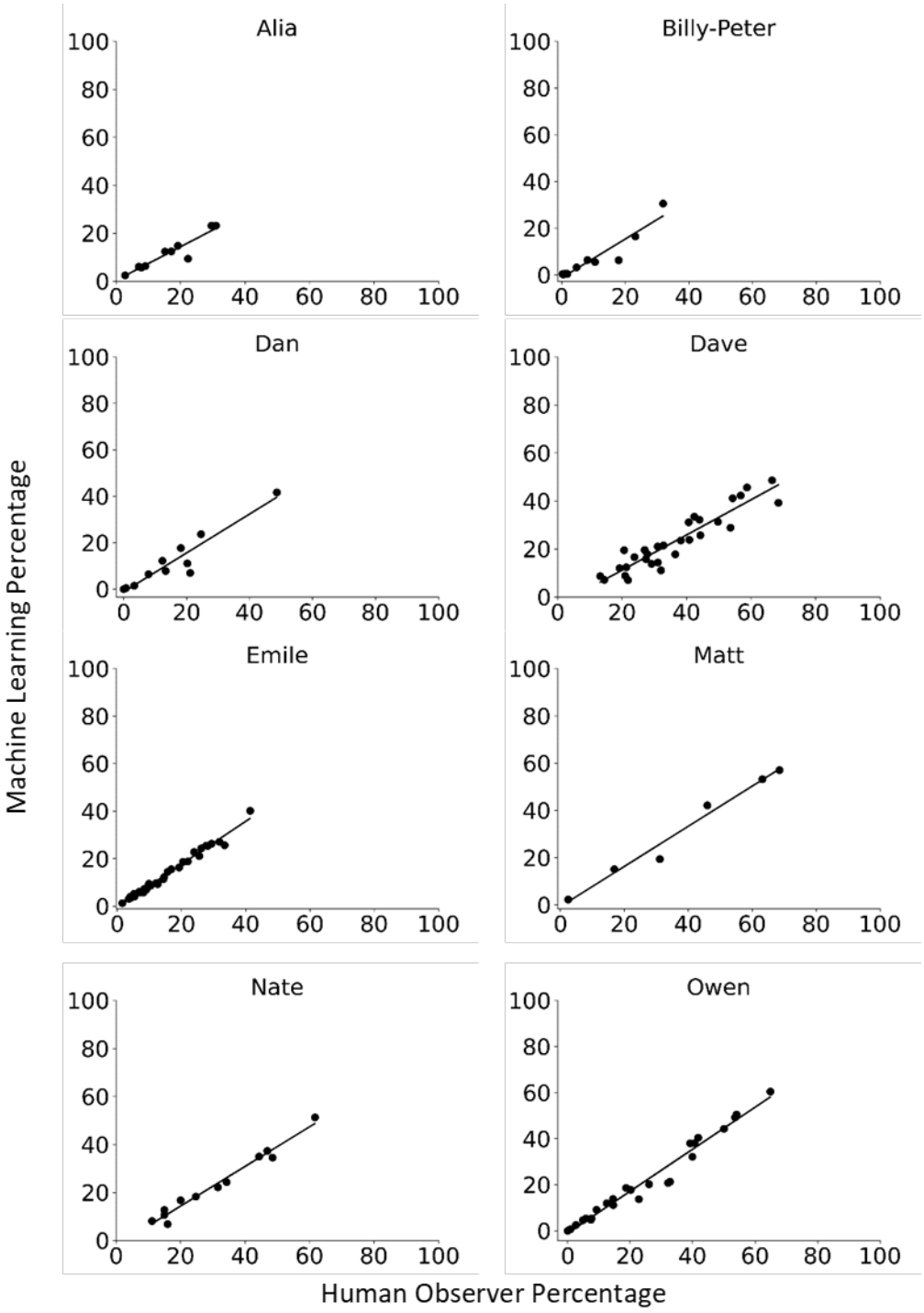

Figure 5.5: Hybrid Analysis: Correlation Between Percentages Measured by the Machine Learning Algorithm and the Human Observer Across Sessions for Each Participant.

For example, a practitioner or researcher may measure vocal stereotypy over a few sessions and then compare their results with those produced by our between-participant analysis model. If the two measures are closely correlated, the practitioner or researcher could use

our model as the main dependent measure, periodically checking to ensure that the automatic measure remains consistent with visual observation over time. To assist practitioners and researchers in using our models, we developed a free, user-friendly, Windows-based executable software (section 5.4). A more effective approach would involve adding the new participant's data to the trained model (i.e., hybrid approach) to produce even more accurate predictions. However, the drawback of hybrid models is that the practitioner or researcher would need to retrain the model for each client, requiring some programming skills. One of the next steps is to develop programs that automate this process.

Several factors may explain why the current models produced more accurate results than those previously reported by [14]: including more features may have provided a richer data representation, allowing the algorithm to model the data more comprehensively. Additionally, the XCiT model may better learn patterns and complex relationships across features than the simple feed-forward network used in the prior study.

We tuned the number of epochs and pre-trained the models in the current study, which may have further improved their performance. While our results are promising, our method still has limitations. All data originated from the same research team using a specific protocol (see [14]), which may limit generalizability and introduce bias. Furthermore, the participants engaged in a limited number of vocal stereotypy forms. Future research could address this issue by using our models as a starting point to train new models (i.e., as pre-trained models) with more participants and diverse forms of vocal stereotypy.

Another limitation is that the kappa values were lower than the correlations, meaning the models are not well-suited for analyzing within-session patterns. As mentioned earlier, models that require additional training (i.e., within-participant and hybrid) remain inaccessible to clinicians and researchers who do not have programming knowledge. Therefore, user-friendly tools should be developed to incorporate these new data into our models. Lastly, the model size is currently too large to detect vocal stereotypy in real-time. The model only supports post hoc analysis of video recordings. Future research should focus on developing more efficient, smaller models that could eventually be used for real-time analysis. Reducing the effort required for data collection will benefit not only practitioners and researchers but also those supported by the science of applied behavior analysis

5.4 Vocal Stereotypy Software

Since psychologists and practitioners may not have the expertise to run code and test models on their data, we developed user-friendly software with an easy-to-use Graphical User Interface (GUI) so that individuals without programming knowledge can benefit from our

model. The software, shown in Figure 5.6, can assist users in analyzing their data and is available at [59].

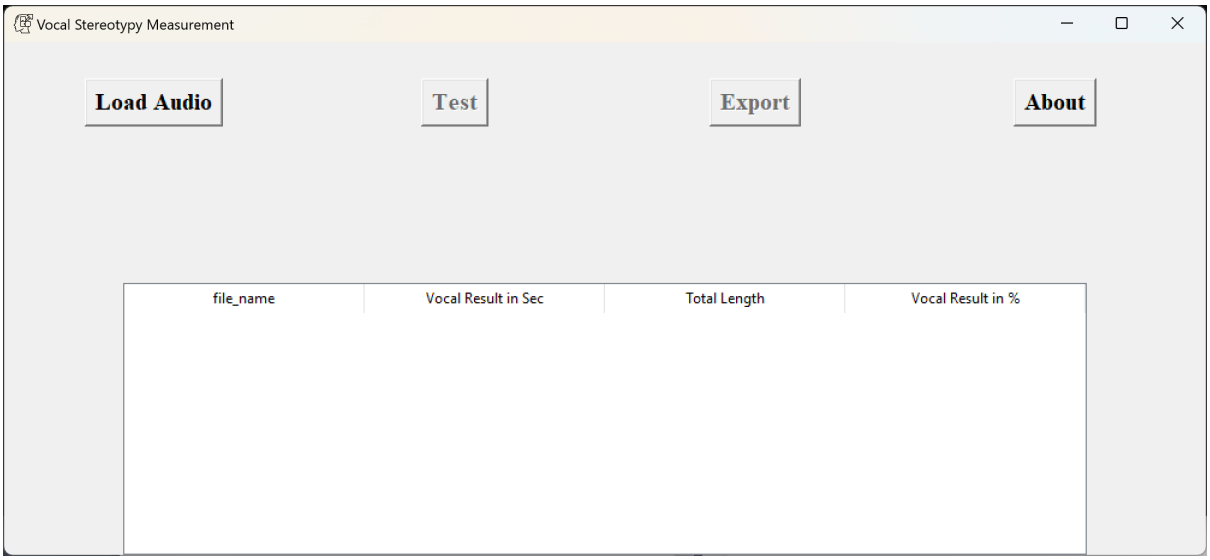

Figure 5.6: An image of the software.

Using the software is simple. First, users can click the "Load Audio" button to select the directory containing the audio files. Then, by clicking the "Test" button, the testing procedure begins, and the results will be displayed in the table below the buttons. The results will show the amount of vocal stereotypy in each audio file (session), both in seconds and as percentages. Additionally, by clicking the "Export File" button, users can save the results as a CSV file in any chosen directory. Finally, by clicking the "About" button, users can view information about the GitHub repository for the project, as shown in Figure 5.7. Our code and models are available **here**.

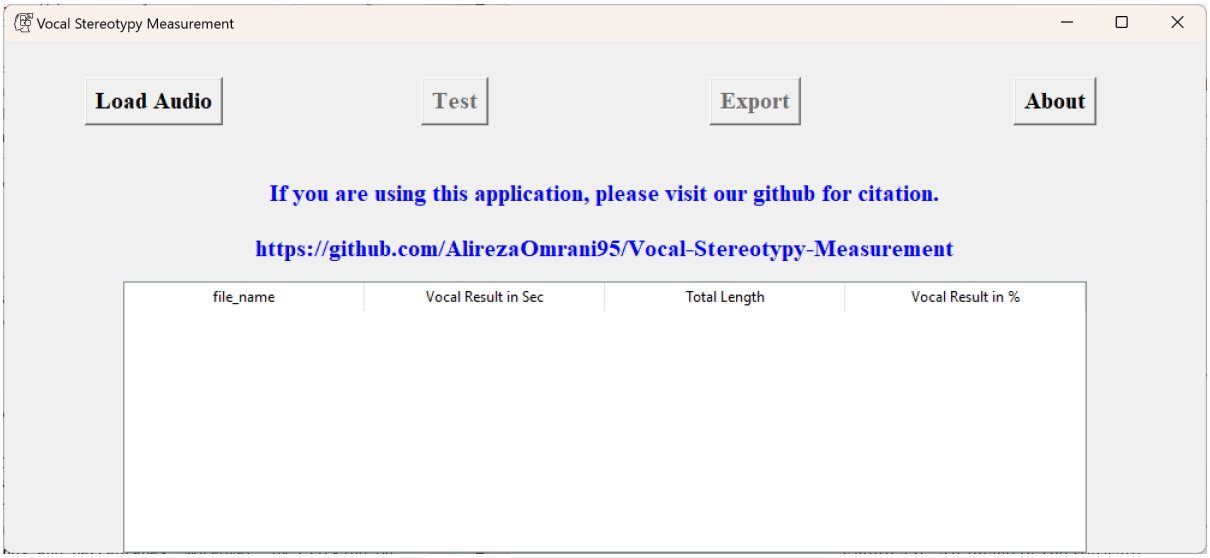

Figure 5.7: About section of the software.

Chapter 6

Conclusion

The primary goal of this dissertation was to analyze the behavior of children with ASD. To achieve this, the research was divided into two key areas: computer vision and audio processing. The specific questions addressed in this study were as follows:

Computer Vision

- Develop and evaluate methods to distinguish children with ASD from non-autistic peers using facial images, examining the suitability and effectiveness of single facial images in classifying children with ASD using deep learning models.
- Utilize explainability techniques to interpret the outputs of the computer vision models, ensuring the results are transparent and understandable, and gain deeper insights into the facial features that AI models rely on to distinguish children.

Audio Processing

- Examine the vocalizations of children with ASD to identify patterns and measure vocal stereotypy.
- Apply deep learning methods to analyze audio data, providing a robust framework for detecting and classifying vocal behaviors.

Additionally, this dissertation explored HDR imaging to enhance image details, which has broad applications in computer vision, including in our facial recognition project.

Regarding the computer vision domain, our findings demonstrate, at a fundamental level, that it is possible to differentiate children with ASD from non-autistic children using a single facial image. However, further research with more data would provide more promising results. The model's ability to achieve 92% accuracy highlights the potential of facial structure

as a diagnostic tool for ASD, suggesting that subtle facial features may serve as meaningful indicators for classification. That said, static facial images capture only a snapshot of an individual's facial structure, and important dynamic aspects of social interaction and emotional expression are not fully represented. Facial expressions, which are often dynamic and context-dependent, may contain crucial information that would improve the model's performance if considered over time.

For the audio processing section, the analysis revealed distinct patterns of vocal stereotypy in children with ASD. Deep learning techniques enabled accurate measurement and classification of these vocal patterns. The study compared the model's performance with a previous model across three factors: within-participant, between-participant, and hybrid. In the within-participant approach, the model was trained and tested using data from only one participant at a time. For the between-participant method, the model was trained using data from all participants except one, which was used as the test set. The hybrid approach combined both strategies by using one session from a participant for testing and all other data from that participant, as well as from other participants, for training. The new model outperformed the previous model for all participants across all factors, as demonstrated by higher Pearson correlation, Spearman correlation, and Kappa scores. Additionally, we developed a Windows-based software for the audio processing project, allowing psychologists without programming knowledge to effectively use the model.

The findings have substantial implications for developmental psychology and special education. By establishing methods to distinguish children with ASD based on facial images and vocal patterns, this study offers new tools for early diagnosis and intervention. The use of explainability techniques ensures that these methods can be trusted and understood by clinicians and educators.

This study was limited by a relatively small sample size, which may affect the generalizability of the findings, particularly when applied to larger, more diverse populations. The facial image analysis was limited to static images, which did not capture dynamic expressions, thus restricting the ability of the model to generalize to real-time applications where expressions vary over time. Additionally, the vocal stereotypy analysis was conducted with a limited number of participants (8 children), which may impact the robustness and reliability of the findings. The small sample size, combined with the focus on a specific group of children, limits the model's ability to generalize across different demographic groups, such as varying ages, ethnicities, or socio-economic backgrounds.

Machine learning models are sensitive to the characteristics of the data they are trained on. Thus, the results may not be easily transferable to populations outside the studied group, which could have distinct patterns of behavior or communication. To improve gen-

eralizability and robustness, future research should include larger, more diverse sample sizes and dynamic image analysis. Additionally, incorporating video analysis for dynamic facial expressions, as well as 3D joint skeletons, could enhance the model's ability to capture complex behaviors in individuals with ASD. Expanding the dataset to include a broader range of participants, alongside testing these models in different populations, would improve the model's applicability and increase the potential for more accurate predictions across diverse settings.

This dissertation has contributed to the field by providing novel methods for analyzing the behavior of children with ASD and enhancing image detail through HDR imaging. These findings underscore the importance of advanced imaging techniques and their potential to drive further advancements in computer vision and developmental psychology. The integration of image classification, behavioral observations, and interviews in ASD screening and diagnosis, combined with the superior performance of new audio processing models and the development of user-friendly software, highlights the comprehensive approach needed for effective early detection and intervention in autism research.

Appendix A

Supplimantary chapter 4

Figure A.1 shows examples of the explainable methods for false negative and false positive cases. Additionally, Figure A.2 presents the average graphs of LIME and RISE for both false positive and false negative cases. As demonstrated, similar to the true positive and true negative cases, LIME is a more effective technique for explaining the model.

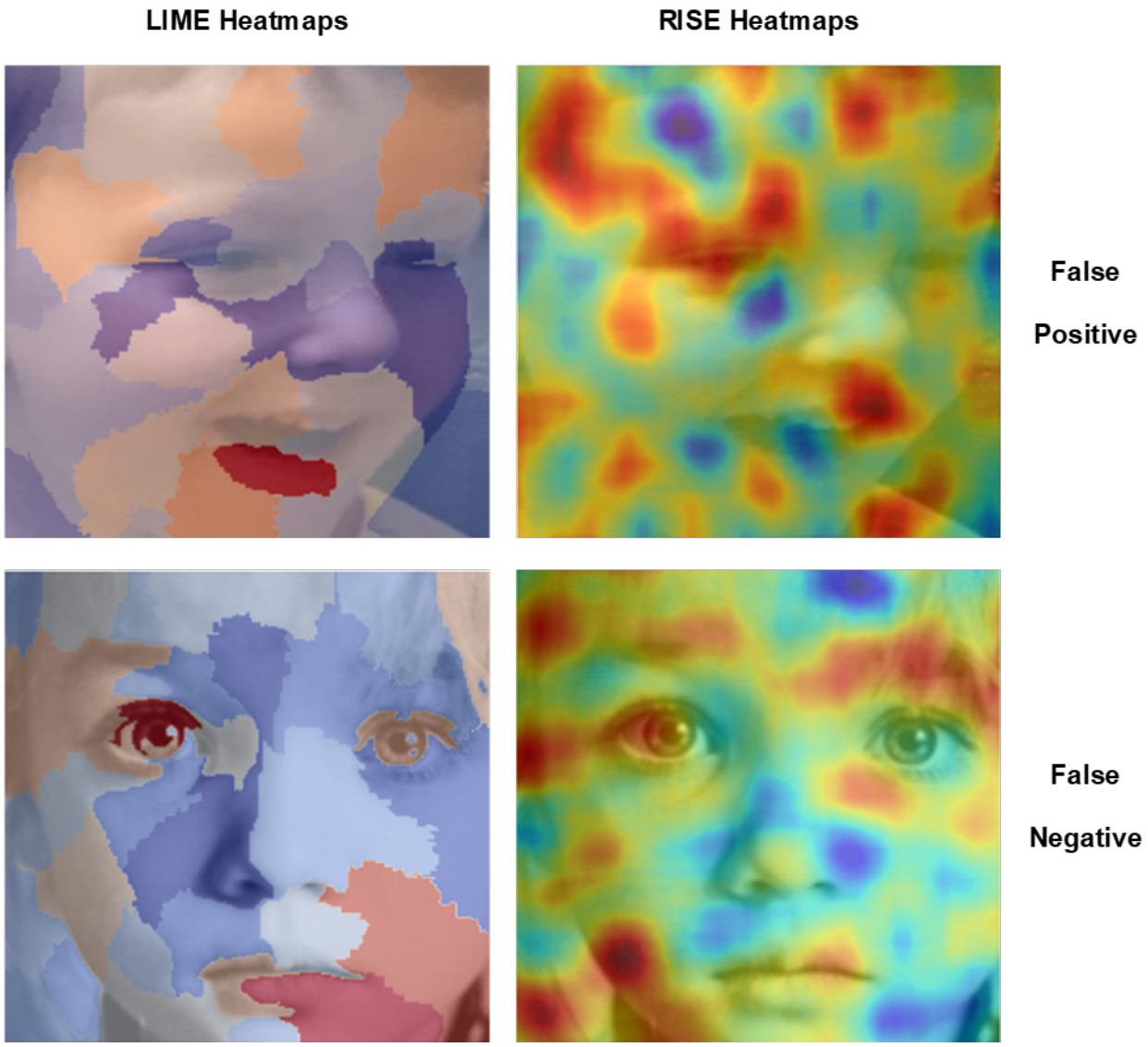

Figure A.1: LIME (left panels) and RISE (right panels) heatmaps for a false positive case (upper panels) and false negative case (lower panels).

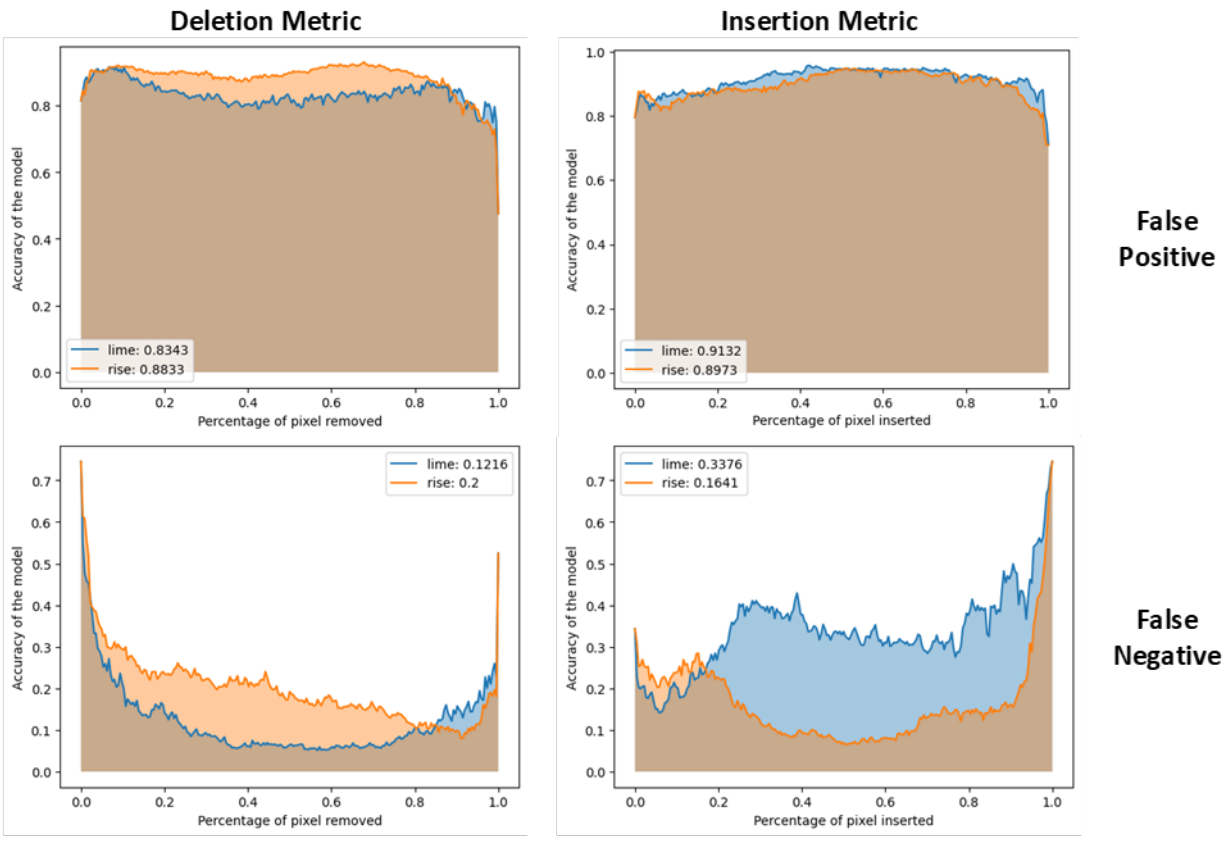

Figure A.2: Average values for all false positive (upper panels) and false negative (lower panels) samples. The graphs on the left represent the deletion metric, and on the right, the insertion metric metric.

Bibliography

[1] Autistic children facial image data set. Available at https://drive.google.com/drive/folders/1XQU0pluL0m3TIlXqntano12d68peMb8A. Last retrieved May 9, 2025.

[2] Israr Ahmad, Javed Rashid, Muhammad Faheem, Arslan Akram, Nafees Ahmad Khan, and Riaz ul Amin. Autism spectrum disorder detection using facial images: A performance comparison of pretrained convolutional neural networks. *Healthc. Technol. Lett.*, January 2024.

[3] Erin N Ahrens, Dorothea C Lerman, Tiffany Kodak, April S Worsdell, and Courtney Keegan. Further evaluation of response interruption and redirection as treatment for stereotypy. *J Appl Behav Anal*, 44(1):95–108, 2011.

[4] Mariano Alcañiz Raya, Javier Marín-Morales, Maria Eleonora Minissi, Gonzalo Teruel Garcia, Luis Abad, and Irene Alice Chicchi Giglioli. Machine learning and virtual reality on body movements' behaviors to classify children with autism spectrum disorder. *J Clin Med*, 9(5), April 2020.

[5] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced high dynamic range imaging*. AK Peters/CRC Press, 2017.

[6] Summer Bottini, Jennifer Gillis, and Raymond Romanczyk. Impact of data collection format on training and treatment integrity during discrete-trial implementation. *Behav. Anal. (Wash., DC)*, 21(2):140–152, May 2021.

[7] Kristine D Cantin-Garside, Zhenyu Kong, Susan W White, Ligia Antezana, Sunwook Kim, and Maury A Nussbaum. Detecting and classifying self-injurious behavior in autism spectrum disorder using machine learning techniques. *J Autism Dev Disord*, 50(11):4039–4052, November 2020.

[8] Gaofeng Cao, Fei Zhou, Kanglin Liu, Anjie Wang, and Leidong Fan. A decoupled kernel prediction network guided by soft mask for single image hdr reconstruction. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(2s), feb 2023.

[9] Xu Cao, Wenqian Ye, Elena Sizikova, Xue Bai, Megan Coffee, Hongwu Zeng, and Jianguo Cao. Vitasd: Robust vision transformer baselines for autism spectrum disorder facial diagnosis. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[10] John O Cooper, Timothy E Heron, and William L Heward. *PowerPoint slides for applied behavior analysis*. Pearson, Upper Saddle River, NJ, 3 edition, April 2022.

[11] Tianhong Dai, Wei Li, Xilei Cao, Jianzhuang Liu, Xu Jia, Ales Leonardis, Youliang Yan, and Shanxin Yuan. Wavelet-based network for high dynamic range imaging, 2022.

[12] Yipeng Deng, Qin Liu, and Takeshi Ikenaga. Attention-guided network with inverse tone-mapping guided up-sampling for hdr imaging of dynamic scenes. *Multimedia Tools and Applications*, 81(9):12925–12944, Apr 2022.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

[14] Marie-Michèle Dufour, Marc J Lanovaz, and Patrick Cardinal. Artificial intelligence for the measurement of vocal stereotypy. *J Exp Anal Behav*, 114(3):368–380, November 2020.

[15] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K. Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Trans. Graph.*, 36(6), nov 2017.

[16] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jegou. XCiT: Cross-Covariance image transformers. 2021.

[17] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM Trans. Graph.*, 36(6), nov 2017.

[18] Mark D. Fairchild. The hdr photographic survey. In *International Conference on Communications in Computing*, 2007.

[19] John Michael Falligant and Jennifer A Vetter. Quantifying false positives in simulated events using partial interval recording and momentary time sampling with dual-criteria methods. *Behav. Interv.*, 35(2):281–294, April 2020.

[20] Joshua Fasching, Nicholas Walczak, Vassilios Morellas, and Nikolaos Papanikolopoulos. Classification of motor stereotypies in video. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, September 2015.

[21] Jan Froehlich, Stefan Grandinetti, Bernd Eberhardt, Simon Walter, Andreas Schilling, and Harald Brendel. Creating cinematic wide gamut HDR-video for the evaluation of tone mapping operators and HDR-displays. In Nitin Sampat, Radka Tezaur, Sebastiano Battiato, and Boyd A. Fowler, editors, *Digital Photography X*, volume 9023, page 90230X. International Society for Optics and Photonics, SPIE, 2014.

[22] Adrian Galdran, Gustavo Carneiro, and Miguel A. González Ballester. On the optimal combination of cross-entropy and soft dice losses for lesion segmentation with out-of-distribution robustness. In Moi Hoon Yap, Connah Kendrick, and Bill Cassidy, editors, *Diabetic Foot Ulcers Grand Challenge*, pages 40–51, Cham, 2023. Springer International Publishing.

[23] Subash Gautam, Prabin Sharma, Kisan Thapa, Mala Deep Upadhaya, Dikshya Thapa, Salik Ram Khanal, and Vítor Manuel de Jesus Filipe. Screening autism spectrum disorder in childrens using deep learning approach : Evaluating the classification model of yolov8 by comparing with other models, 2023.

[24] Abraham Graber and Jessica Graber. Applied behavior analysis and the abolitionist neurodiversity critique: An ethical analysis. *Behav. Anal. Pract.*, pages 1–17, March 2023.

[25] Saghi Hajisharif, Joel Kronander, and Jonas Unger. Adaptive dualiso hdr reconstruction. *EURASIP Journal on Image and Video Processing*, 2015(1):41, Dec 2015.

[26] Nicole L Hausman, Noor Javed, Molly K Bednar, Madeleine Guell, Erin Schaller, Rose E Nevill, and Sungwoo Kahng. Interobserver agreement: A preliminary investigation into how much is enough? *J. Appl. Behav. Anal.*, 55(2):357–368, March 2022.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[28] Jinhan Hu, Gyeongmin Choe, Zeeshan Nadir, Osama Nabil, Seok-Jun Lee, Hamid Sheikh, Youngjun Yoo, and Michael Polley. Sensor-realistic synthetic data engine for multi-frame high dynamic range photography. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2020.

[29] Kai Hu, Junlan Jin, Fei Zheng, Liguo Weng, and Yiwu Ding. Overview of behavior recognition based on deep learning. *Artif. Intell. Rev.*, June 2022.

[30] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.*, 36(4), jul 2017.

[31] Inge Kamp-Becker. Autism spectrum disorder in icd-11—a critical reflection of its possible impact on clinical practice and research. *Molecular Psychiatry*, 29(3):633–638, Mar 2024.

[32] Soma Khan, Tulika Basu, Joyanta Basu, Madhab Pal, and Rajib Roy. System assisted vocal response analysis and assessment of autism in children: A machine learning based approach. In *Speech and Computer*, Lecture notes in computer science, pages 506–519. Springer Nature Switzerland, Cham, 2023.

[33] Yuma Kinoshita and Hitoshi Kiya. Automatic exposure compensation using an image segmentation method for single-image-based multi-exposure fusion. *APSIPA Transactions on Signal and Information Processing*, 7:e22, 2018.

[34] Yuma Kinoshita and Hitoshi Kiya. Scene segmentation-based luminance adjustment for multi-exposure image fusion. *IEEE Transactions on Image Processing*, 28(8):4101–4116, 2019.

[35] Green Rosh K.S., Anmol Biswas, Mandakinee Singh Patel, and B H Pawan Prasad. Deep multi-stage learning for hdr with large object motions. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4714–4718, 2019.

[36] Phuoc-Hieu Le, Quynh Le, Rang Nguyen, and Binh-Son Hua. Single-image hdr reconstruction by multi-exposure generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023.

[37] Linda A LeBlanc, Coby Lund, Chris Kooken, Janet B Lund, and Wayne W Fisher. Procedures and accuracy of discontinuous measurement of problem behavior in common practice of applied behavior analysis. *Behav. Anal. Pract.*, 13(2):411–420, June 2020.

[38] Byeong Dae Lee and Myung Hoon Sunwoo. Hdr image reconstruction using segmented image learning. *IEEE Access*, 9:142729–142742, 2021.

[39] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. Deep chain hdri: Reconstructing a high dynamic range image from a single low dynamic range image. *IEEE Access*, 6:49913–49924, 2018.

[40] Susan E Levy, Audrey Wolfe, Daniel Coury, John Duby, Justin Farmer, Edward Schor, Jeanne Van Cleave, and Zachary Warren. Screening tools for autism spectrum disorder in primary care: A systematic evidence review. *Pediatrics*, 145(Suppl 1):S47–S59, April 2020.

[41] Fangya Li, Ruipeng Gang, Chenghua Li, Jinjing Li, Sai Ma, Chenming Liu, and Yizhen Cao. Gamma-enhanced spatial attention network for efficient high dynamic range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1032–1040, June 2022.

[42] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[43] Zhen Liu, Yinglong Wang, Bing Zeng, and Shuaicheng Liu. Ghost-free high dynamic range imaging with context-aware transformer. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 344–360, Cham, 2022. Springer Nature Switzerland.

[44] Matthew J Maenner, Zachary Warren, Ashley Robinson Williams, Esther Amoakohene, Amanda V Bakian, Deborah A Bilder, Maureen S Durkin, Robert T Fitzgerald, Sarah M Furnier, Michelle M Hughes, Christine M Ladd-Acosta, Dedria McArthur, Elise T Pas, Angelica Salinas, Alison Vehorn, Susan Williams, Amy Esler, Andrea Grzybowski, Jennifer Hall-Lande, Ruby H N Nguyen, Karen Pierce, Walter Zahorodny, Allison Hudson, Libby Hallas, Kristen Clancy Mancilla, Mary Patrick, Josephine Shenouda, Kate Sidwell, Monica DiRienzo, Johanna Gutierrez, Margaret H Spivey, Maya Lopez, Sydney Pettygrove, Yvette D Schwenk, Anita Washington, and Kelly A Shaw. Prevalence and characteristics of autism spectrum disorder among children aged 8 years - autism and developmental disabilities monitoring network, 11 sites, united states, 2020. *MMWR Surveill. Summ.*, 72(2):1–14, March 2023.

[45] Juan Mar'in-Vega, Michael Sloth, Peter Schneider-Kamp, and Richard Rottger. Drhdr: A dual branch residual network for multi-bracket high dynamic range imaging. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 843–851, 2022.

[46] Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, 2021.

[47] Catherine K Martinez, Alison M Betz, Clare J Liddon, and Rebecca L Werle. A progression to transfer RIRD to the natural environment. *Behav. Interv.*, 31(2):144–162, April 2016.

[48] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, Billy Chen, John F. Hughes, and Shree K. Nayar. Optical splitting trees for high-precision monocular imaging. *IEEE Computer Graphics and Applications*, 27(2):32–42, 2007.

[49] Nico Messikommer, Stamatios Georgoulis, Daniel Gehrig, Stepan Tulyakov, Julius Erbach, Alfredo Bochicchio, Yuanyou Li, and Davide Scaramuzza. Multi-bracket high dynamic range imaging with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 547–557, June 2022.

[50] Cheol-Hong Min and John Fetzner. Vocal stereotypy detection: An initial step to understanding emotions of children with autism spectrum disorder. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, July 2018.

[51] Cheol-Hong Min and John Fetzner. Training a neural network for vocal stereotypy detection. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, July 2019.

[52] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

[53] Amar Mitiche and Ismail Ben Ayed. *Variational and level set methods in image segmentation*, volume 5. Springer Science & Business Media, 2010.

[54] S.K. Nayar and T. Mitsunaga. High dynamic range imaging: spatially varying pixel exposures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 472–479 vol.1, 2000.

[55] Ali Reza Omrani, Marc J Lanovaz, and Davide Moroni. Machine learning to measure vocal stereotypy: An extension. August 2024.

[56] Ali Reza Omrani, Marc J Lanovaz, and Davide Moroni. Towards the development of explainable machine learning models to recognize the faces of autistic children. April 2024.

[57] Ali Reza Omrani and Davide Moroni. Supervised image segmentation for high dynamic range imaging. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5, 2023.

[58] Ali Reza Omrani and Davide Moroni. High dynamic range imaging via visual attention modules. *IEEE Access*, 12:50911–50924, 2024.

[59] Ali Reza Omrani, Davide Moroni, and Marc Lanovaz. Vocal stereotypy measurement. Last retrieved May 9, 2025.

[60] Alireza Omrani, Mohammad Reza Soheili, and Manoochehr Kelarestaghi. High dynamic range image reconstruction using multi-exposure wavelet hdrcnn. In *2020 International Conference on Machine Vision and Image Processing (MVIP)*, pages 1–4, 2020.

[61] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[62] Eduardo Pérez-Pellitero, Sibi Catley-Chandar, Richard Shaw, Aleš Leonardis, Radu Timofte, Zexin Zhang, Cen Liu, Yunbo Peng, Yue Lin, Gaocheng Yu, et al. Ntire 2022 challenge on high dynamic range imaging: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1023, 2022.

[63] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *ArXiv*, abs/1806.07421, 2018.

[64] K. Ram Prabhakar, Susmit Agrawal, and R. Venkatesh Babu. Segmentation guided deep hdr deghosting, 2022.

[65] K. Ram Prabhakar, Susmit Agrawal, Durgesh Kumar Singh, Balraj Ashwath, and R. Venkatesh Babu. Towards practical and efficient high-resolution hdr deghosting with cnn. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 497–513, Cham, 2020. Springer International Publishing.

[66] K. Ram Prabhakar, Rajat Arora, Adhitya Swaminathan, Kunal Pratap Singh, and R. Venkatesh Babu. A fast, scalable, and reliable deghosting method for extreme exposure fusion. In *2019 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2019.

[67] Vadamodula Prasad, G. V. Sriramakrishnan, and I. Diana Jeba Jingle. Autism spectrum disorder detection using brain mri image enabled deep learning with hybrid sewing training optimization. *Signal, Image and Video Processing*, 17(8):4001–4008, Nov 2023.

[68] Eduardo Pérez-Pellitero, Sibi Catley-Chandar, Aleš Leonardis, Radu Timofte, Xian Wang, Yong Li, Tao Wang, Fenglong Song, Zhen Liu, Wenjie Lin, Xinpeng Li, Qing Rao, Ting Jiang, Mingyan Han, Haoqiang Fan, Jian Sun, Shuaicheng Liu, , et al. Ntire 2021 challenge on high dynamic range imaging: Dataset, methods and results. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 691–700, 2021.

[69] Nastaran Mohammadian Rad and Cesare Furlanello. Applying deep learning to stereotypical motor movement detection in autism spectrum disorders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, December 2016.

[70] John T Rapp and Timothy R Vollmer. Stereotypy i: a review of behavioral assessment and treatment. *Res. Dev. Disabil.*, 26(6):527–547, November 2005.

[71] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[72] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[73] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 1041–1044, New York, NY, USA, 2014. Association for Computing Machinery.

[74] Hidir Selcuk Nogay and Hojjat Adeli. Diagnostic of autism spectrum disorder based on structural brain mri images using, grid search optimization, and convolutional neural networks. *Biomedical Signal Processing and Control*, 79:104234, 2023.

[75] Ana Serrano, Felix Heide, Diego Gutierrez, Gordon Wetzstein, and Belen Masia. Convolutional sparse coding for high dynamic range imaging. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics*, EG '16, page 153–163, Goslar, DEU, 2016. Eurographics Association.

[76] Mohammad Shabaz, Parveen Singla, Malik Mustafa Mohammad Jawarneh, and Himayun Mukhtar Qureshi. A novel automated approach for deep learning on stereotypical autistic motor movements. In *Advances in Medical Diagnosis, Treatment, and Care*, pages 54–68. IGI Global, 2021.

[77] Lei She, Mao Ye, Shuai Li, Yu Zhao, Ce Zhu, and Hu Wang. Single-image hdr reconstruction by dual learning the camera imaging process. *Engineering Applications of Artificial Intelligence*, 120:105947, 2023.

[78] Aarti Thakore, Andrea Kelly, Anna Ingeborg Petursdottir, and Morgan Stockdale. Evaluation of a treatment package for chronic, stereotypic hand mouthing of a child diagnosed with autism. *Behav. Anal. Pract.*, June 2024.

[79] Michael D. Tocci, Chris Kiser, Nora Tocci, and Pradeep Sen. A versatile hdr video production system. *ACM Trans. Graph.*, 30(4), jul 2011.

[80] C Y Andy Tse, C L Pang, and Paul H Lee. Choosing an appropriate physical exercise to reduce stereotypic behavior in children with autism spectrum disorders: A non-randomized crossover study. *J. Autism Dev. Disord.*, 48(5):1666–1672, May 2018.

[81] J. Tumblin, A. Agrawal, and R. Raskar. Why i want a gradient camera. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 103–110 vol. 1, 2005.

[82] Md. Zasim Uddin, Md. Arif Shahriar, Md. Nadim Mahamood, Fady Alnajjar, Md. Ileas Pramanik, and Md Atiqur Rahman Ahad. Deep learning with image-based autism spectrum disorder analysis: A systematic review. *Engineering Applications of Artificial Intelligence*, 127:107185, 2024.

[83] A. Vadivel, M. Mohan, Shamik Sural, and A. K. Majumdar. Segmentation using saturation thresholding and its application in content-based retrieval of images. In

Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 33–40, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[84] An Gia Vien, Seonghyun Park, Truong Thanh Nhat Mai, Gahyeon Kim, and Chul Lee. Bidirectional motion estimation with cyclic cost volume for high dynamic range imaging. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1182–1189, 2022.

[85] An Vien Gia and Chul Lee. Single-shot high dynamic range imaging via deep convolutional neural network. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1768–1772, 2017.

[86] Lin Wang and Kuk-Jin Yoon. Deep learning for hdr imaging: State-of-the-art and future trends. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8874–8895, 2021.

[87] Lin Wang and Kuk-Jin Yoon. Deep learning for hdr imaging: State-of-the-art and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8874–8895, 2022.

[88] Shangzhe Wu, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Deep high dynamic range imaging with large foreground motions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[89] Jun Xiao, Qian Ye, Tianshan Liu, Cong Zhang, and Kin-Man Lam. Multi-scale sampling and aggregation network for high dynamic range imaging, 2022.

[90] Qingsen Yan, Dong Gong, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, Ian Reid, and Yanning Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[91] Qingsen Yan, Dong Gong, Pingping Zhang, Qinfeng Shi, Jinqiu Sun, Ian Reid, and Yanning Zhang. Multi-scale dense networks for deep high dynamic range imaging. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 41–50, 2019.

[92] Qingsen Yan, Song Zhang, Weiye Chen, Yuhang Liu, Zhen Zhang, Yanning Zhang, Javen Qinfeng Shi, and Dong Gong. A lightweight network for high dynamic range imaging. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 823–831, 2022.

[93] Qian Ye, Masanori Suganuma, Jun Xiao, and Takayuki Okatani. Learning regularized multi-scale feature flow for high dynamic range imaging, 2022.

[94] Faliu Yi and Inkyu Moon. Image segmentation: A survey of graph-cut methods. In *2012 international conference on systems and informatics (ICSAI2012)*, pages 1936–1941. IEEE, 2012.

[95] Gaocheng Yu, Jin Zhang, Zhe Ma, and Hongbin Wang. Efficient progressive high dynamic range image restoration via attention and alignment network. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1123–1130, 2022.

[96] Yue Yu, Sally Ozonoff, and Meghan Miller. Assessment of autism spectrum disorder. *Assessment*, 31(1):24–41, 2024. PMID: 37248660.

[97] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[98] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

[99] Hang Zhao, Boxin Shi, Christy Fernandez-Cull, Sai-Kit Yeung, and Ramesh Raskar. Unbounded high dynamic range photography using a modulo camera. In *2015 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, 2015.

[100] Zhong Zhao, Zhipeng Zhu, Xiaobin Zhang, Haiming Tang, Jiayi Xing, Xinyao Hu, Jianping Lu, and Xingda Qu. Identifying autism with head movement features by implementing machine learning algorithms. *J. Autism Dev. Disord.*, 52(7):3038–3049, July 2022.