University of Massachusetts Dartmouth

Department of Computer and Information Science

Optimizing Rash Datasets for Lyme Disease Detection

A Thesis in

Computer Science

by

Melanie G Thibodeau

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

January 2026

We approve the thesis of Melanie G Thibodeau

Date of Signature

Iren Valova
Professor, Department of Computer and Information Science
Thesis Advisor

Gokhan Kul
Associate Professor, Department of Computer and Information Science
Thesis Committee

Firas Khatib
Associate Professor, Department of Computer and Information Science
Thesis Committee

Adnan El-Nasan
Graduate Program Director, Computer Science

Haiping Xu
Chairperson, Department of Computer and Information Science

Robert Griffin
Dean, College of Engineering

Tesfay Meressi
Associate Provost for Graduate Studies

Abstract

Optimizing Rash Datasets for Lyme Disease Detection

by Melanie G Thibodeau

This thesis focuses on optimizing image datasets through augmentation methods for the detection of Lyme disease. Lyme disease often is accompanied by an erythema migrans rash, but other types of rashes that may appear similar and have their identification mistaken.

Training a model to accurately recognize subtle differences between rash types requires a large quantity of images. However, there is a lack of publicly available datasets containing Lyme disease rashes, which results in the thesis using a smaller dataset for its foundation. Using a public crowdsourced dataset, "Lyme Disease Rashes", by Edward Zhang, the objective of this study is to improve the accuracy of YoloV7 through image enhancements and augmentations. The study applies a combination of data preprocessing techniques, including CLAHE, photometric transformation, elastic deformation, and MixUp to improve image quality and address dataset imbalances.

YoloV7, an object detection model, was trained on the enhanced dataset to accurately differentiate Lyme disease-related rashes from other dermatological conditions. The results favored the CLAHE pre-processing results over the others. This work contributes to the development of more reliable, automated diagnostic tools for individual users.

Results indicate a significant improvement in detection accuracy, demonstrating the potential of optimized rash datasets in the early identification of Lyme disease.

Acknowledgments

Thank you to Professor Valova for guiding me through this thesis and corresponding project. Also, thank you for all the lessons you provided me with within the classroom and through research. I enjoyed the walks we had together and the talks we had.

Thank you to the professors in my thesis committee for my defense: Professor Khatib and Professor Kul. Thank you to Edward Zhang, who provided the dataset. Without him, this thesis would not have been possible. Finally, thank you to my family and friends who supported me through all of this and more.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background:

Each year, approximately 476,000 Americans are diagnosed and treated for Lyme disease according to the Center for Disease Control and Prevention (CDC) [2]. Lyme disease is a deer tick-borne illness caused by the bacterium Borrelia burgdorferi. It is commonly found in the Northeast and Midwest regions of the United States, although cases do occur in other parts of the country, albeit less frequently [2]. Some of the early symptoms are erythema migrans rash, fever, chills, and headache. It is important to catch the disease in the early stages, as left untreated it can cause arthritis, nerve pain, heart palpitations, facial palsy, as well as other chronic symptoms [3, 7]. Treatment options for Lyme disease include antibiotics and nonsteroidal anti-inflammatory drugs [10].

One of the most common symptoms of Lyme disease is a uniformly red or oval rash that is over two inches in size, which occurs in approximately 70 to 80 percent of infected individuals [2]. Obviously, not all rashes are indicators of Lyme disease. Rashes can also be caused by other bug bites, bad reactions to medications, ringworm, chemicals, shingles, poison ivy, and other illnesses. Doctors must know the difference between them to make the correct diagnosis.

1.2 The Goal of this Thesis:

The ideal state envisioned after this thesis and the original goal was to develop an application for Lyme disease rash detection. Our investigation pointed out that classifying an existing public dataset obtains less than satisfactory results, aligned our goal to optimize the dataset to improve the classification performance of the Lyme disease rash detection.

1.2.1 Ideal State:

The goal of the thesis is to create an application that can identify Lyme disease rashes from a photo of a rash. The user will open the application and upload their rash image. The model will generate results to identify the rash and the confidence it has in the prediction. While not a medical diagnosis, the results can serve as a prompt to schedule a doctor visit for a test. This would reduce the number of unnecessary visits to the hospital or prompt individuals to follow up with a professional medical visit.

1.2.2 Current State:

Optimization of medical datasets is necessary to train an artificial intelligence (AI) for accurate results for patients. The project “Medical Diagnosis at a Snap of the Camera” [14] aims to provide a recommended correct medical diagnosis for Lyme disease based on an image of a rash. In the ideal form of the thesis, stated above, it would help patients to identify possible causes of rashes and recommend patients to schedule a doctor’s visit. As its extension, this thesis explores the data from “Lyme Disease Rashes” by Edward Zhang [21]. Its different versions include using image enhancements and augmentations to achieve a better precision of identification. As part of the data optimization process, adjustments were applied to the dataset's inclusion criteria, and the results were gathered for analysis. An example of this process is testing Lyme disease to evaluate whether the system would recognize the illness’s rash more accurately compared to other illnesses. The current state of the project is analyzing the adjusted data to learn what results in the optimal dataset for the object-detection model.

1.3 Relevant Work:

Similar works have been proposed in the past which include applications such as Ada [1] and First Derm [5]. Ada is a symptom-based application for all medical problems. The tool asks a series of questions to compile a list of symptoms the patient is experiencing to make an estimated guess what the patient has. First Derm, also called Teledermatology, is an AI-based tool that helps in the diagnosis of skin conditions. According to First Derm's website, 20% of all primary care visits are skin related disorders that are only correctly identified 50% of the time [5]. First Derm also claims that there is about a 32-day waiting period to see a dermatologist. By going through First Derm, the patient gets results within 48 hours. First Derm also uses AI which recognizes 43 skin diseases with 90% accuracy [5]. Like First Derm, this project aims to use images to speed up diagnosis of Lyme disease and help ensure the diagnosis is correct. Like Ada and First Derm, it will encourage individuals to get help when needed.

Chapter 2: Dataset

This chapter reviews and analyzes the dataset that originated from Kaggle and expanded upon with usage of images from public databases. Kaggle is a platform where individuals can share datasets, knowledge, collaborate with each other, or join machine learning competitions. The dataset contains rashes: Lyme disease, ringworm, fixed-drug rashes, and pityriasis rosea. Additionally, this chapter reviews the lifecycle of an updated version of this dataset that was not utilized. It concludes by addressing the removal of some of the images that did determine the visibility of the rashes or artifacts contained.

2.1 Dataset Analysis:

Table 1: Shows the sources of the images within the dataset and the images

	Kaggle: [20]	DermNet:[11]	VisualDX: [16]	SkinSight: [13]	Kaggle:[19]
Fixed Drug Rashes	48	14	78	15	95
Lyme Disease Rashes	181	8	61		
Pityriasis Rosea	104	50	72	24	
Ringworm	105	57	88		

The dataset, curated by Edward Zhang and hosted on Kaggle, comprises 438 images classified into four categories: Lyme disease, fixed-drug rash, ringworm, and pityriasis rosea [17]. Developed to address the lack of publicly available Lyme disease datasets available for public usage. Additional images were included from DermNet, VisualDx, and SkinSight which are other public databases. These databases allow searches and filters for individual illness with all the images file names containing the illness within. Combining the images from these databases to the original dataset was critical to increase the class sizes to be larger. The final raw dataset contains 1000 images with 250 images in each class.

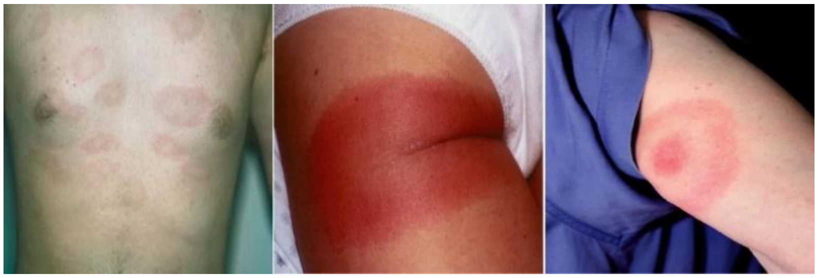

Figure 1: Lyme disease positive images.

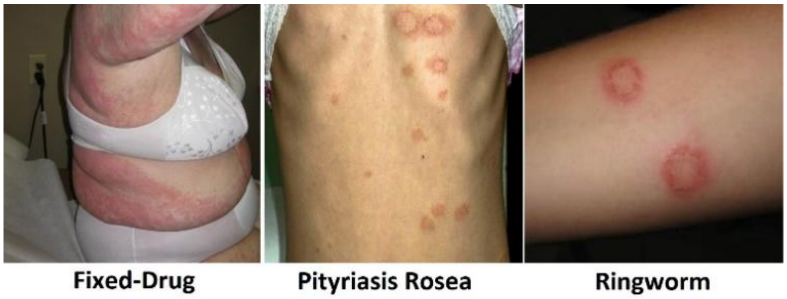

Figure 2: Other rashes from the datasets that are not Lyme disease.

The dataset consists of a variety of different images depicting Lyme disease and the other conditions which were all uploaded into Roboflow. Roboflow is an online dataset repository where you can store data and reformat it. Within Roboflow each image received bounding boxes to show where the rashes were and label them with a class. The dataset containing 1000 images was scaled to 640x640 pixels. Additionally, Roboflow applied auto-orientation enhancements to the dataset to have the objects within the bounding boxes more focused. This process involves discarding the image’s EXIF metadata, which contains image resolution, coloring scaling, and other camera-specific information. This does not directly affect YoloV7, it is beneficial for maintaining data consistency and preventing interference during the training process [4]. The data was subsequently divided into subsets, 70% for training, 20% for validation, and 10% for testing.

The pityriasis rosea in Figure 2 represents a milder version as compared to other examples used throughout model training. In contrast, the ringworm image aligns closely with the other ringworm images in the dataset. The fixed-drug rash is one of the many versions in the training set as drug rash appears different depending on the person.

2.1.1 Common Traits:

The dataset contains light skinned individuals without other features on the skin. Very few in the dataset contained tattoos, moles, birthmarks, or scars. The lighting within the dataset varies as they were taken by different cameras and in different environments. Prior to processing in Roboflow, the images were of varying sizes.

2.2 Dataset Lifecycle:

The latest version of “Lyme Disease Erythema Migrans Rashes” dataset by Edward Zhang was updated in February, 2023 [19]. It now contains more than 5000 images in total which is more than four times the size of the original size. The new dataset contains thirteen types of ailments seen on the skin level. On the Kaggle page both datasets are still available for download with the original version archived as RashData and the newer dataset called Lyme Full Statified.

The dataset has undergone significant changes from the original version as nine new classes were added. Notably, the updated version renamed the ringworm from the old dataset with its scientific name, tinea corporis. Additionally, it introduced several new classes, bacterial cellulitis, contact dermatitis, erythema marginatum, local reaction to arthropod bite, serum sickness-like reactions, shingles rash, spider bite rash, superficial erythema annulare centrifugum, and urticartia. Each of these classes contains around 343 images. Also, in the Lyme disease positive group there are now 941 images which are 760 images more than the previous version. Unlike the old dataset, all the images are now scaled 278x182.

2.3 Filtering Bad Images from Dataset:

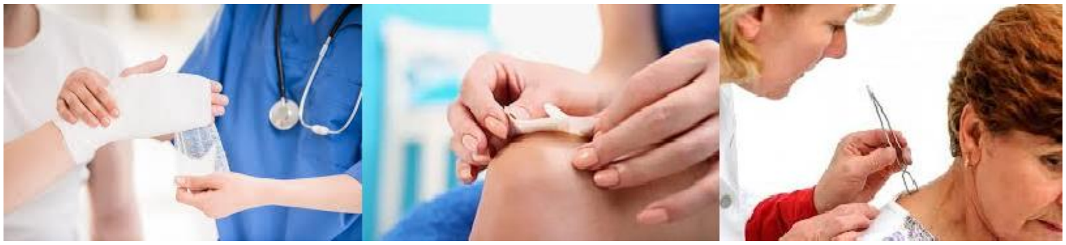

Figure 3: Images not showing any skin ailments.

While the dataset contains mostly good images, there are still images that had to be removed due to several reasons. There were images like Figure 3 in both the original dataset and the newer version, which do not depict skin rashes but instead display unrelated content.

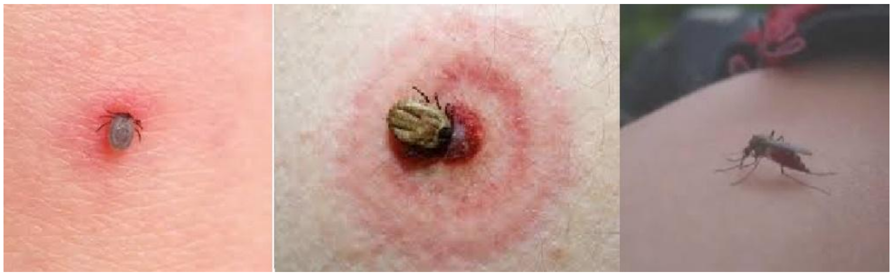

Figure 4: Some of the images removed from "local reaction to arthropod bite" as they had the artifact within it.

Within the dataset, some images were eliminated due to issues such as blurriness and others still showed a bug biting the person. Figure 4 provides examples of images that were excluded from the dataset. A small number of images contained only a bug without a rash which does not qualify as a skin condition as seen in the rightmost image within Figure 4.

Chapter 3: Methodology

This chapter provides an overview of the tools and methods employed throughout the thesis to enhance the dataset and train the artificial intelligence model. It introduces key parts such as CUDA, Anaconda, and Roboflow, detailing their roles within the dataset and the model. Additionally, it explores advanced augmentation techniques, including Regional MixUp, Photometric transformation, Contrast Limited Adaptive Histogram Equalization, and Elastic Deformation, which were implemented to improve the dataset's quality and quantity. Those augmentations were tested on You Only Look Once V7 (YoloV7)

3.1 Tools:

Tools used were CUDA toolkit, Anaconda and Roboflow. Anaconda is a distribution of the Python and R programming languages for scientific computing. Anaconda was used to create a virtual environment to host the pre-made model, YoloV7. The virtual environment allows for Python libraries with user specified versions to be used without interference of other installed libraries outside of the environment. Roboflow is an online dataset repository that stores datasets and reformats them. It was used to add bounding boxes to annotate the dataset and rescale the images to 640x640 pixels. Roboflow provides version control of the different datasets and allows multiple datasets with different augmentations to be uploaded. CUDA was created by Nvidia to be a developer tool to allow development, optimization, and deployment of applications on GPU-accelerated embedded systems. Fast training was possible because CUDA delegated offloading roughly half of the CPU work to the graphics card.

3.2 YoloV7:

This section draws exclusively from the paper authored by the developers of YoloV7 [14,15].

3.2.1 Introduction and Key Features:

The model was trained using YoloV7 which was released in July 2022 by Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolo stands for “You Only Look Once” and is a real-time object detector that can identify objects by classification. This model’s new features include trainable bag-of-freebies that provided techniques for reparameterization on the convolution layer and dynamic label assignment. These bag-of-freebies are optimized during training, without increasing resource usage. Extended efficient layer aggregation networks (E-ELAN) and model scaling are improvements to the Yolo architecture. Table 2 displays the weight variations that contribute towards scaling efforts. These enhancements have led to a very fast model with a 40% reduction in total parameters and a 50% reduction in computation, all while maintaining high accuracy.

Table 2: The table different YoloV7 weights and their parameters. [17,18]

Model	#Para	Flops	Size	FPS	AP^{test}/AP^{val}	AP_{50}^{Test}	AP_{75}^{Test}	AP_{S}^{Test}	AP_{M}^{Test}	AP_{L}^{Test}
YoloV7 -tiny	6.2M	13.8G	640	286	38.7% /38.7%	56.7%	41.7%	18.8%	42.4%	51.9%
YoloV7	36.9M	104.7G	640	161	51.4%/51.2%	69.7%	55.9%	31.8%	55.5%	65.0%
YoloV7 -X	71.3M	189.9G	640	114	53.1%/52.9%	71.2%	57.8%	33.8%	57.1%	67.4%

3.2.2 Architecture:

YoloV7’s architecture has evolved from previous versions of the Yolo series through the introduction of E-ELAN and models scaled for concatenation-based models to improve both accuracy and computational efficiency. The basic model contains the backbone, Feature Pyramid Network (FPN), and multiple heads. The E-ELAN is in the backbone's primary computational block. It enhanced the model capabilities to learn while maintaining the gradient flow of the expand, shuffle, and merge cardinality as seen in Figure 5 [17]. The expand cardinality expands the number of channels and cardinality. Continued to shuffle cardinality, which shuffles the channels into groups after the feature map is created. Conclusively, merge cardinality caused shuffled groups to merge to improve the dynamic model.

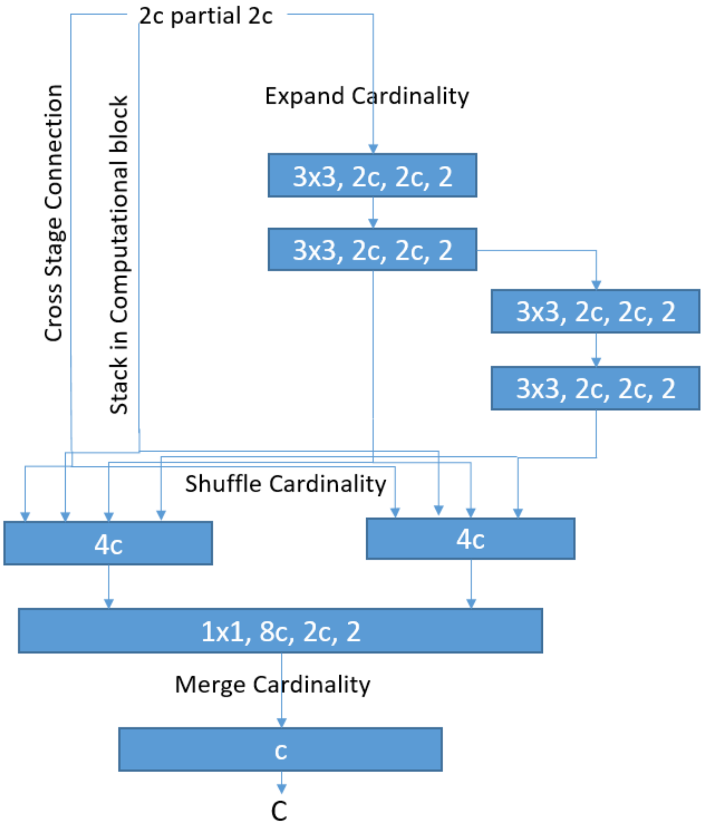

Figure 5: Diagram of the E-ELAN [17].

3.2.3 Understanding Metric Evaluation and Matrix:

YoloV7 training results in a series of graphs and a confusion matrix that summarize the model's training performance. The confusion matrix consists of the training classes, the false negative (FN) and false positive (FP) classes unless removed. The false negative and false positive are not a misclassification, rather it is a lack of detection. In this thesis, false negative is the model missing rashes within an image, and a false positive is mistaking something else as a rash. Additionally, the diagonal, top-left to bottom-right is the true positive (TP) which depicts the percentage of rash detection and classification that the model got correctly. Everything else on the matrix conveys the model that detected the rashes, but misclassified them.

The series the graphs resulting in a completed run contain precision-recall curve (PR), precision-confidence curve, recall-confidence curve, and F1 curve. The precision-recall curve is an overall graph that summarizes the balance between precision and recall. Precision is the proportion of the true positive detection the model finds during training. Recall is the proportion of the true positive out of all the detected objects, in these rashes.

3.2.4: Testing Methods of the Trained Datasets:

After training the augmented versions of the dataset, the testing script was modified to exclude the background classification and precision that YoloV7 automatically incorporates during model training. Since each image in the dataset contained only a single rash type, classification between bounding boxes was not a primary focus during testing. The modified test script employs a low confidence score threshold to eliminate the built-in background class. As a result, multiple bounding boxes were generated, which were then organized by their confidence levels. The bounding box with the highest confidence score was used to determine the image's label.

3.3: Augmentation Methods

Augmentation methods are employed to expand the dataset. These techniques increase the dataset size and help prevent model overfitting. The following data augmentation methods were implemented to ensure distinct and non-overlapping transformations. The additional benefit of these different methods is the ability to combine them, which enables the creation of even more images.

3.3.1: Regional Mixup:

Regional Mixup is a relatively new form of MixUp first published in 2024 by Saptarshi Saha and Utpal Garain [12]. Conceptually, it is combination between the traditional MixUp and CutMix augmentation, a method that uses the simple cut and paste approach between two images. The traditional MixUp overlayed two images with semi-transparency. It integrates portions from multiple images, instead of replacing entire patches. This method allows for a more realistic and wider variety of combinations. This focuses on creating new training images to train a more generalized and robust model. Additionally, Regional MixUp allows class balancing and dataset expansion while mitigating the worry of overfitting.

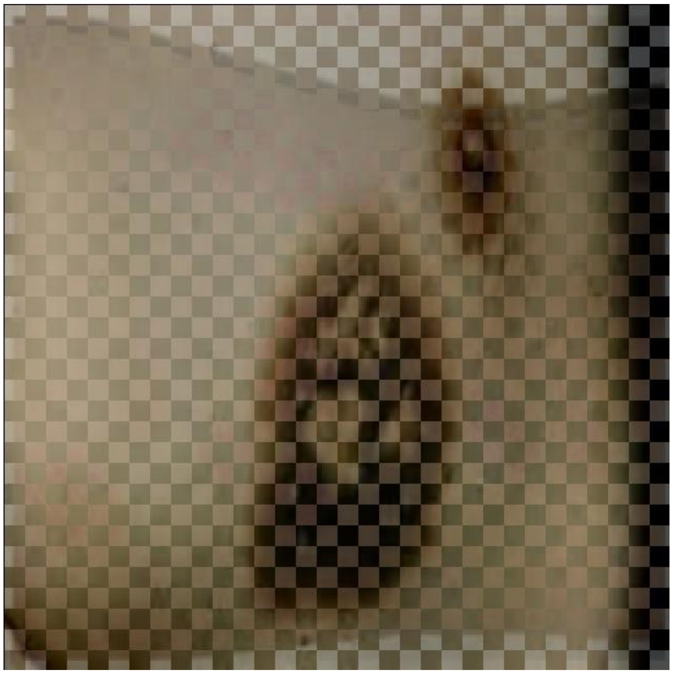

Figure 6: Example of Regional MixUp.

Regional MixUp, Figure 6, was implemented using the Python libraries: OS, OpenCV (cv2), PIL, Random, and NumPy. The algorithm contained a loop which selected two images from the original dataset during each iteration. It checked the image's size and adjusted if necessary to ensure the images being overlapped were the same size. Additionally, a suitability function was implemented to prevent unsuitable pairings. The function evaluated the two images' brightness and contrast to see if the pairing was reasonable. Next, the images are converted into NumPy arrays to allow for pixel manipulation. Afterwards, a grid is overlaid onto the images to divide them into structured regions based on the "region_height" and "region_width" parameters. The images are then alternately overlapped across the grid to create a blended image. If an edge does not match, the grid is adjusted by one pixel to prevent artifacts from occurring. The transparency of overlapped regions is controlled by the parameter α (alpha), which defines the balance between the two images in mixed regions. After the MixUp, the new image is saved to another folder designated for mixed images.

3.3.2: Photometric Dataset:

Photometric transformation involved applying a series of transformations to images in order to introduce additional variation to a dataset. The augmentation method included adjustments to brightness, contrast, and saturation, followed by the addition of Gaussian noise [21]. The use of the method simulated natural lights and color variations which the dataset could use. Gaussian noise simulates real-world noise which the model must learn to ignore. Another unique use of this method was to adjust the parameters to restore images of book pages [21]. This augmentation is a widely used combination augmentation method used in computer vision and image processing.

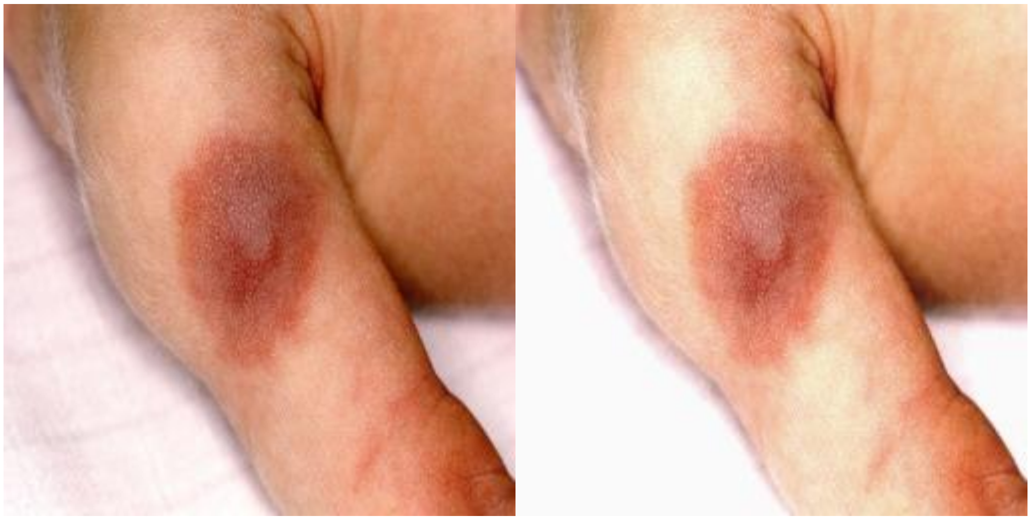

Figure 7: Left image is original without augmentation; right image contains the augmentation.

Photometric transformation, Figure 7, was implemented using the Python libraries: OS, CV2, Numpy, PIL, and random. Python Imaging Library (PIL) was used to support the handling and manipulation of images. Photometric transformation had a set list of parameters that the program applied to the image one at a time. The parameters for brightness, contrast, and saturation had a set range of 0.5 through 1.5 that were randomly generated for each image. This range was chosen to ensure the images retain a realistic visual characteristic while still introducing variability. Brightness of 0.0 turned the full image black, contrast of 0 removed any details in the image, and saturation of 0.0 removed all color and changed it to grayscale so the range of the random values were implemented. After those three parameters are applied the image is temporarily converted into a float32 NumPy array to alter the pixel values.

The gaussian noise parameters are as follows:

1. **Mean(μ)** determines the average value of the noise and was set to mean = 0.

2. **Standard deviation (σ)** determines the amount of noise and sometimes is referred to as sigma in codes. Standard deviation = 25 in the code. Setting the standard deviation high would mean more noise, which is not always preferred.
3. **X** represents the various values that the noise can take as it is distributed along the Bell curve of a Gaussian (normal) distribution. The shape and spread of this curve are defined by the standard deviation (σ) and the mean (μ).
4. **Formula for Gaussian Function (Probability Density Function):**

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

After the noise is applied, the NumPy array is converted back into a normal image. Once the augmentation is finished, the augmented images are saved into a different folder to preserve the original dataset.

3.3.3 CLAHE:

Contrast Limited Adaptive Histogram Equalization (CLAHE) is an augmentation method that enhances the contrast of the images within a dataset [22]. This augmentation method works well for medical images like X-rays or rashes because the contrasting makes details easier to control. It also produces good results for images with poor lighting or minor blurring as it adapts contrast enhancement to local regions rather than applying a uniform adjustment. This process extends beyond the normal contrasting, the augmentation method uses an idea of a grid within an image. The image is divided into smaller regions called tiles, enabling localized contrast enhancement. This ensures a more accurate and adaptive adjustment of contrast across different parts of the image without over saturation or adding too much extra noise.

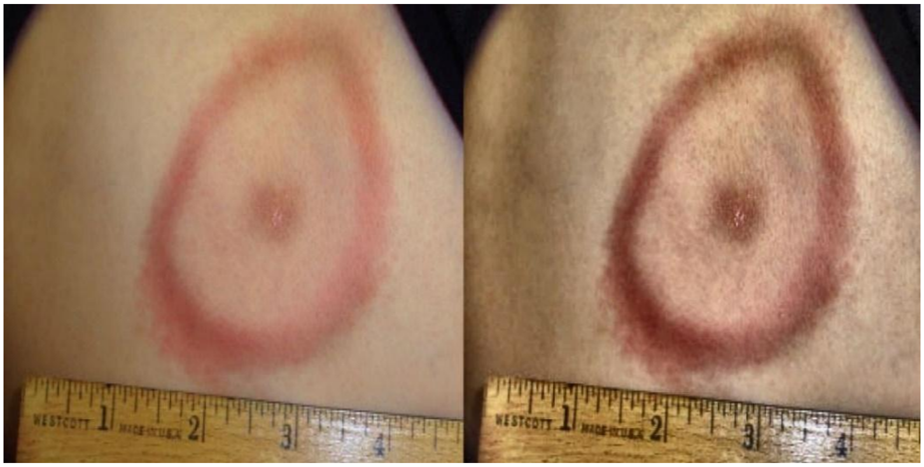

Figure 8: Left image is without CLAHE applied; the right image is with CLAHE applied.

CLAHE, Figure 8, was implemented using the Python library OpenCV (cv2), which is a tool for image processing. Libraries like OS and NumPy are supported and used in the workflow. OpenCV handles the image processing, while OS is used to access the dataset folder and manage input images. NumPy is essential for efficiently handling image arrays.

In some implementations of CLAHE, a grayscale conversion step can be applied before augmentation. However, this was not applied in this case, as the dataset remained in its original color format. Two primary parameters control CLAHE:

1. **clipLimit**: This parameter sets the threshold for contrast adjustment. The default value is 40.0, but it was lowered to 2.0 in this implementation to ensure a subtle enhancement without introducing excessive noise.

2. **tileGridSize**: This parameter determines the size of the tiles into which the image is divided. The default size of 8×8 was used here, meaning the image was split into grids of 8×8 tiles.

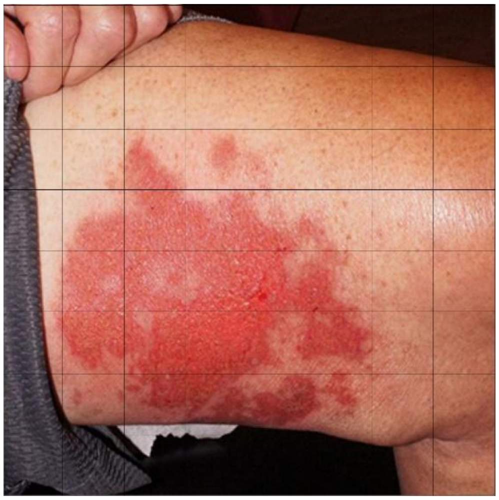

Figure 9: Demonstrating an image with the “tileGridSize” parameter applied.

After configuring these parameters, the CLAHE algorithm is applied to the images. Figure 9 presents a simplified representation of the image divided into a tile grid. The processed images are then saved into a separate folder to preserve the integrity of the original dataset.

3.3.4: Elastic Deformation:

Elastic deformation is a term that describes the temporarily distorting of an object’s shape under stress until the stress is removed. The concept was eventually introduced to machine learning and computer vision was an augmentation method for images. The method takes images and introduces various amounts of stretching and twisting within a set range to the full image. For this dataset, the argumentation is to simulate different skin-type stretching and less of the twisting to simulate what the model could possibly see in the future. It also uses the Gaussian function to smooth the stretch and twists to ensure it looks natural.

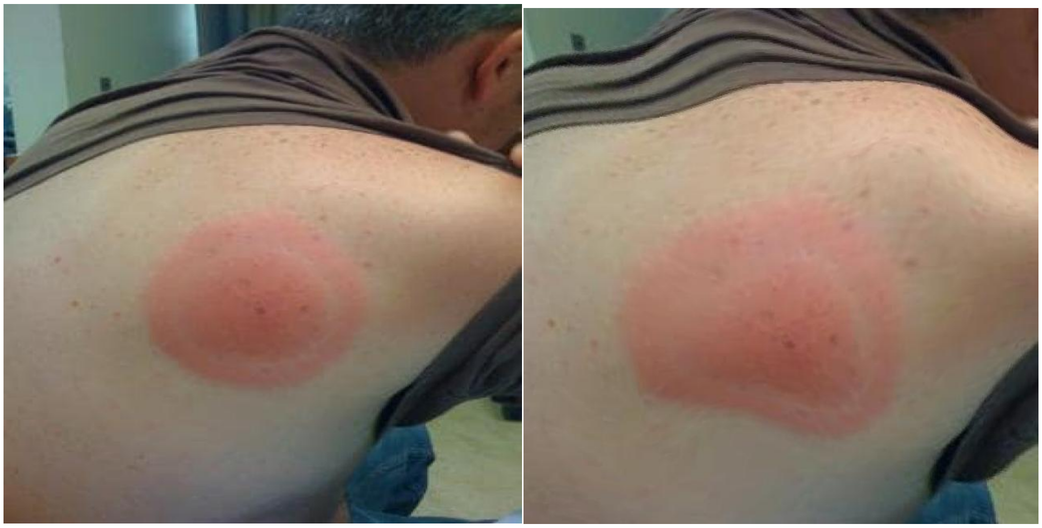

Figure 10: Left image is the original without the augmentation, the right has the augmentation.

This elastic deformation, Figure 10, was implemented using the Python libraries: OS, CV2, Numpy, SciPy, and random [15]. SciPy provides efficient tools for optimization, integration, linear algebra, signal processing, and statistics, all built on top of NumPy. The main parameters:

1. **Alpha**, the range that controls the amount of distortions. Determines how much the pixels are stretched and in what direction. Program's alpha = (55,70)
2. **Sigma**, controls the smoothness of the distortions via Gaussian function. If the value is too small the distortions will appear wavy while a bigger value makes clearer deformations. The program sigma = (20,40).

The parameters' values are randomly selected within the ranges provided. The first stage of the elastic deformation is stretching the pixels based on alpha. Next stage is the Gaussian function (see section 3.2.3.2) where any abruptions are smoothed out to make the image look more natural. Finally, this is all put into a mapping pixel process where the image's deformations are applied. The images are then saved into a different folder than the original dataset to preserve the original dataset.

3.3.5 Various Combinations Attempted:

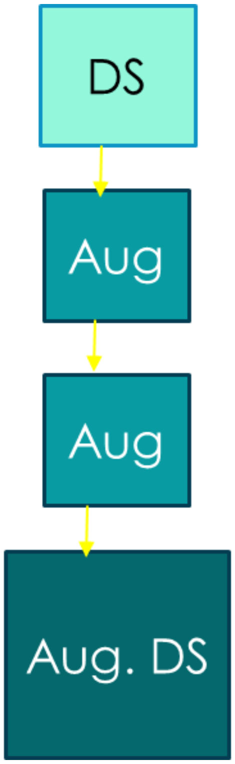

Figure 11: Combos: CLAHE and Photometric Transformation.

This method, Figure 11, sequentially layers two augmentation methods with the purpose to expand and improve the dataset. Initially, elastic deformation is applied to the raw dataset to create various distortions and variability. Subsequently, photometric transformations are then applied to the elastic deformed images with variations in lighting and color to simulate different environmental conditions. To conclude, the augmented images are then merged with the original dataset to form an augmented dataset, ensuring the resulting dataset is robust and diverse.

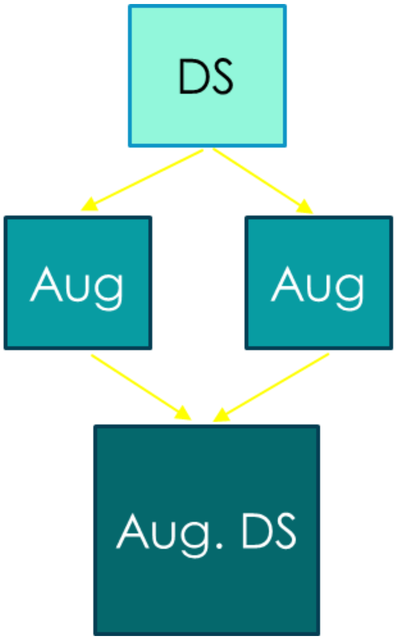

Figure 12: Combo: Photometric Transformation and Elastic Deformation.

This method, Figure 12, utilizes a combination of augmentations applied in parallel to get two new versions of augmented images. The CLAHE technique is employed to improve local contrast and emphasize finer image details, while photometric transformations introduce variations in lighting, color, and brightness to simulate diverse environmental conditions. Once these augmentations are applied, the resulting outputs are then merged into a unified augmented dataset with the original images. This integration ensures the final dataset benefits from the complementary enhancements introduced by both augmentation techniques, resulting in a more diverse and robust dataset.

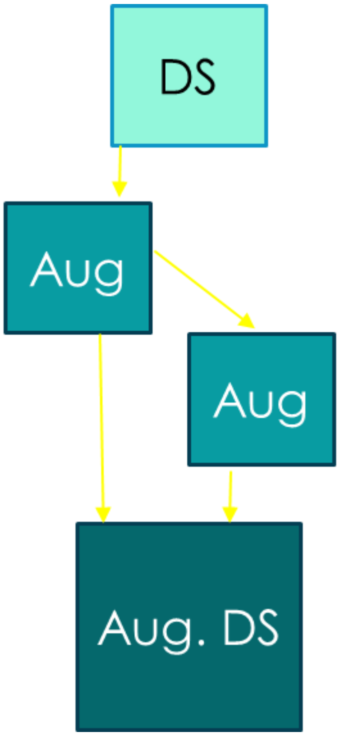

Figure 13: Combo: MixUp and CLAHE.

This method, Figure 13 employs a sequential and complementary augmentation strategy to the dataset. Initially, the MixUp augmentation technique is applied, blending pairs of images to create new ones. A copy of the MixUp-augmented dataset is then processed using the CLAHE technique, which enhances local contrast and emphasizes finer image details. In conclusion, the outputs from both augmentations are integrated back into a unified dataset. This approach leverages the strengths of MixUp and CLAHE to generate a new image and then improve the rashes' visibility.

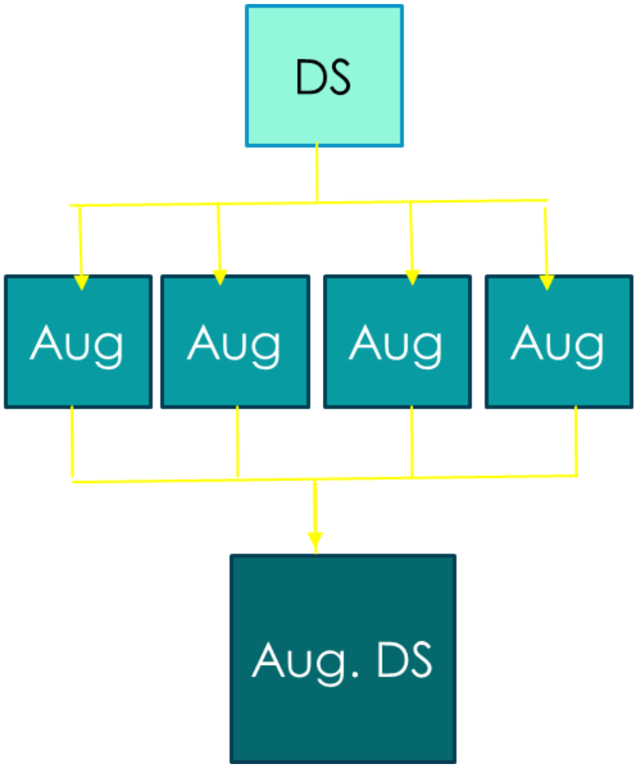

Figure 14: Funneled Combo.

This Funnel Combo, Figure 14, combination method applies the four augmentation methods to the original dataset independently of each other. This occurs by executing the augmentation methods one at a time, but no single image undergoes more than one transformation. Then the independently augmented images are combined with the original dataset to create a new dataset version. This method offers the largest number of images and variation amongst combinations, which is the most ideal for a training model. This version allowed the dataset to at least quadruple the images compared to the original dataset to increase training result potential.

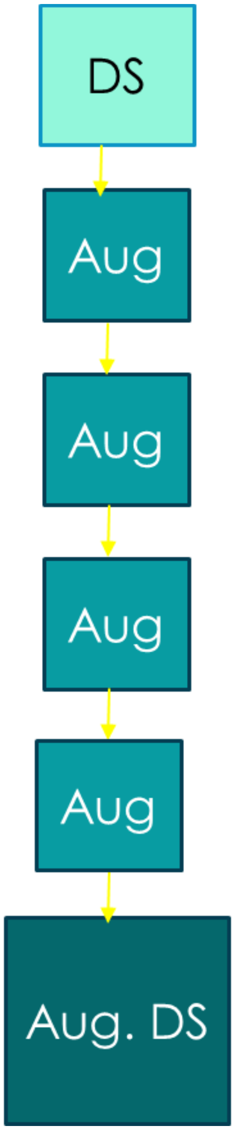

Figure 15: Combined Combo.

The Combined Combo, Figure 15, sequentially applied the augmentation methods in a linear fashion to the images. Initially, the process begins with MixUp to create new images which are merged with the original dataset images. The dataset containing the MixUp images are then augmented by CLAHE, the photometric transformation, and finally elastic deformation. The resulting augmented dataset from the multi-step process is then merged back into the original dataset to ensure maximum diversity within the combined images.

Chapter 4: Results

This chapter provides an in-depth analysis of the results for each of the augmentations applied during this thesis. The Precision and Recall graphs are the validation datasets as the end of the training process in YoloV7. The confusion matrix consists of the test dataset's results and is a separate process. The graphs measure the precision and classification of the bounding boxes whereas the confusion matrix is verifying the classification within the images is correct. Since all the images in the dataset solely contained one rash type, the test was able to verify the rash type rather than the different boxes. (See section 3.2.4 for more information)

4.1 Raw Data:

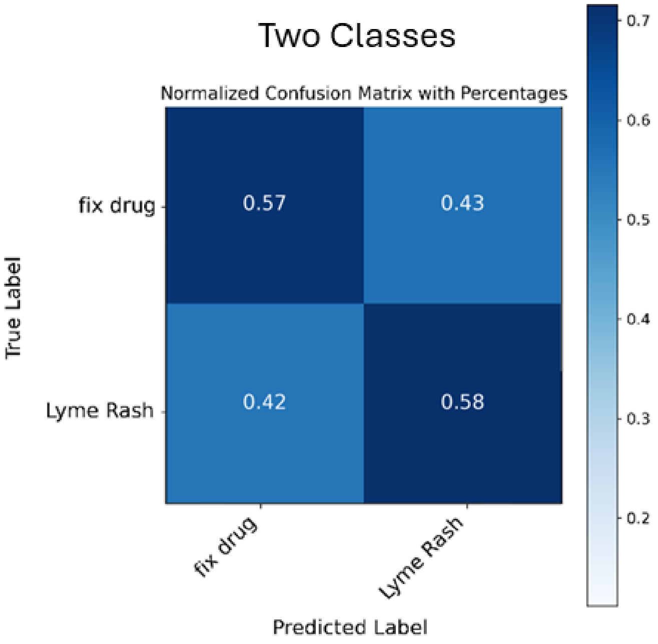

Figure 16: Raw dataset's matrix from the classification test which contained two classes.

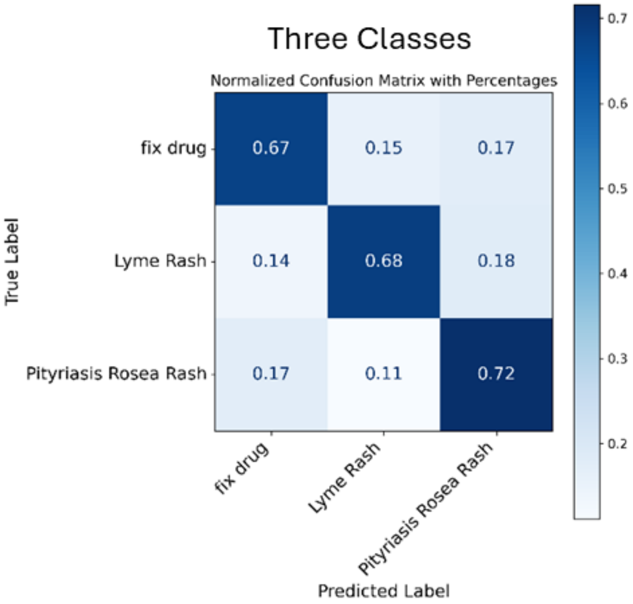

Figure 17: Raw dataset’s matrix from the classification test which contained three classes.

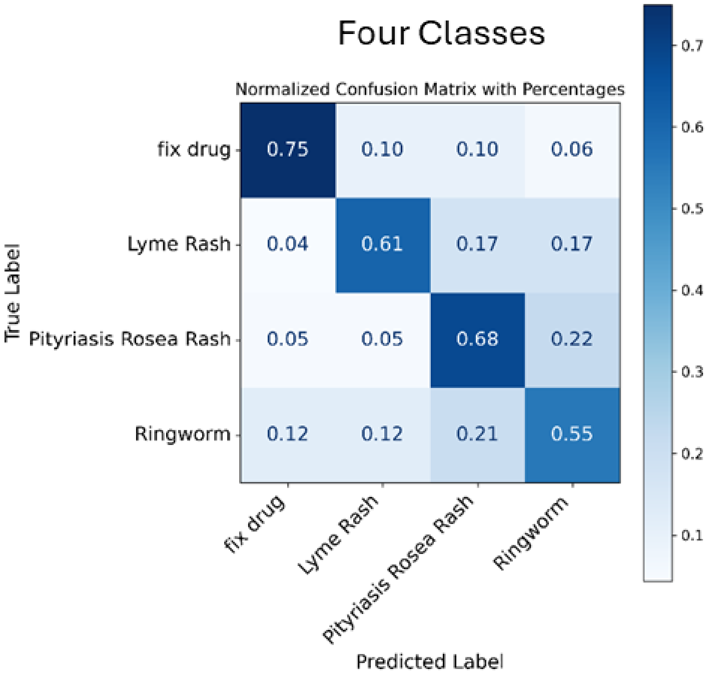

Figure 18: Raw dataset’s matrix from the classification test which contained four classes.

The result matrices from the tests, Figures 16-18, show that as the number of classes increases the classification accuracy rate fluctuates. Figure 16, and three classes, Figure 17, showed notable improvements as the classification accuracy increased by 10% each. This could be due to ringworm and Lyme disease sharing many similarities and causing confusion. Figure 18, which contained all four classes, was slightly less accurate compared to Figure 17 when the ringworm rashes were introduced to the training. It improved the classification of the fixed-drug rashes by 18%, while the other classes were lower. Once more images were added, this allowed the computer to observe some details that differentiate these two classes, improving its ability to classify them correctly.

Raw Data Precision vs Recall Curves:

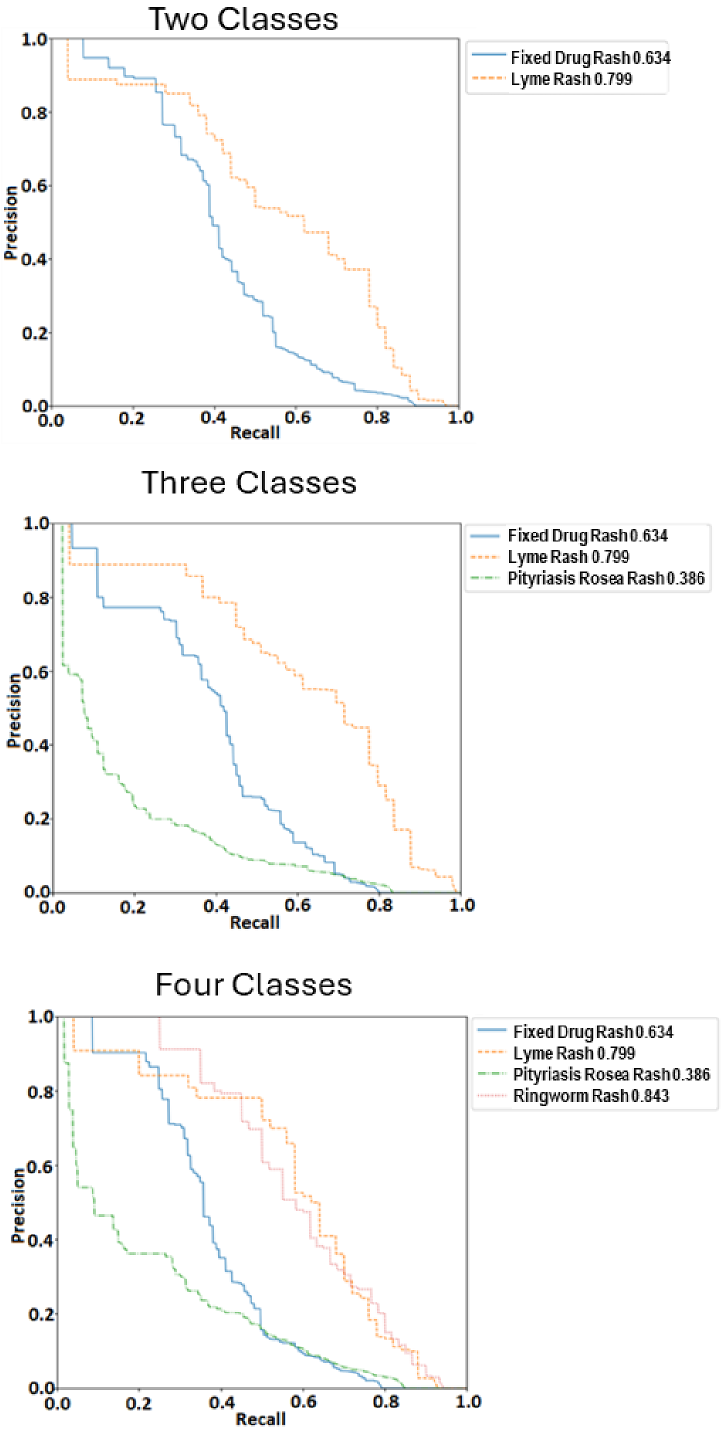

Figure 19: Precision and Recall Curves from the training and validation of the raw dataset. In order the graphs are two classes, three classes, and four classes.

The precision-recall curve for the raw datasets, Figure 19, from the validation dataset performed overall better than the testing according to the confusion matrices. Despite the curve for Lyme disease displays 79.9%, it does not display that way in the test matrices.

4.2 Augmentations :

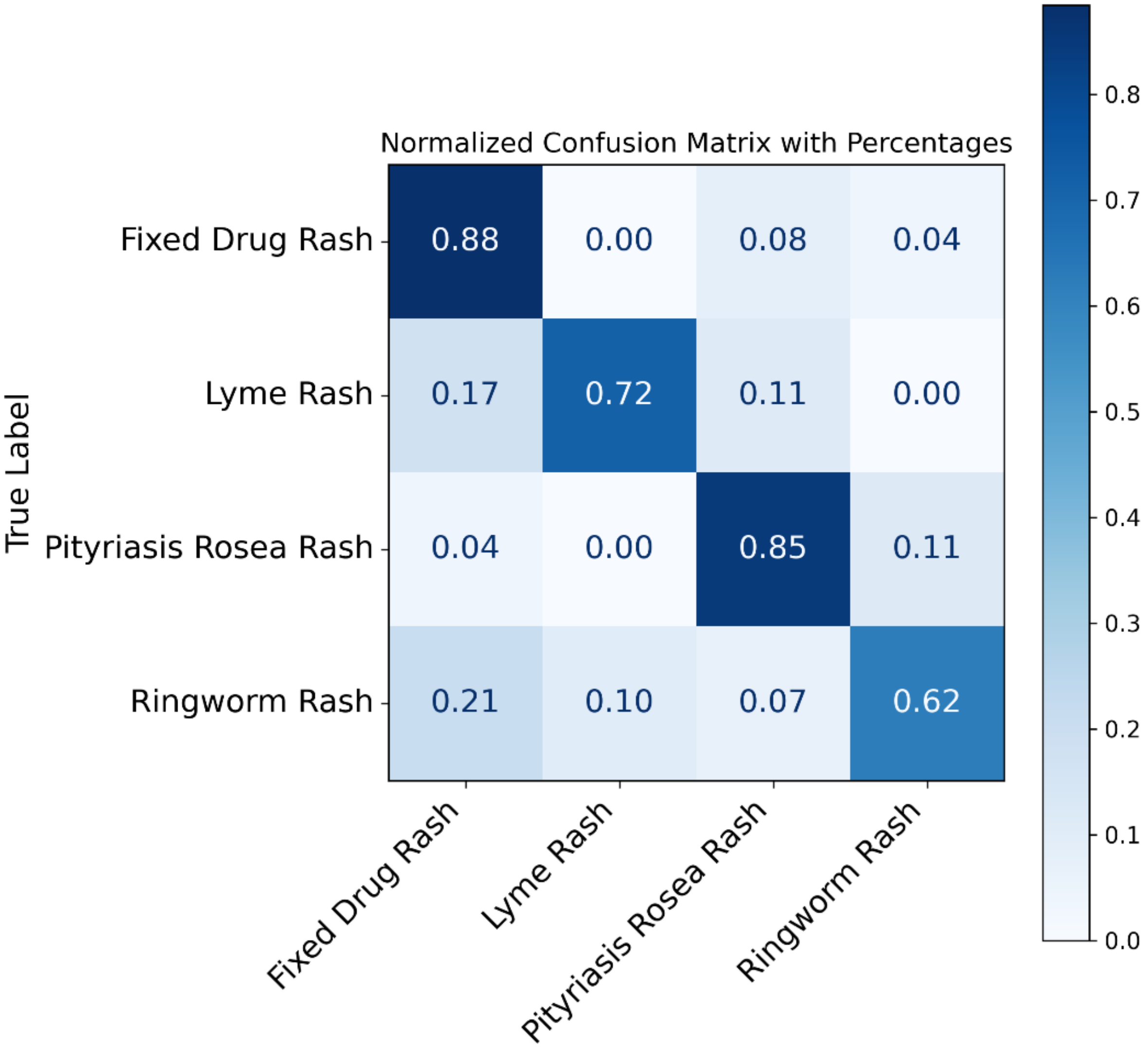

Figure 20: Regional Mixup’s test matrix that measured classification.

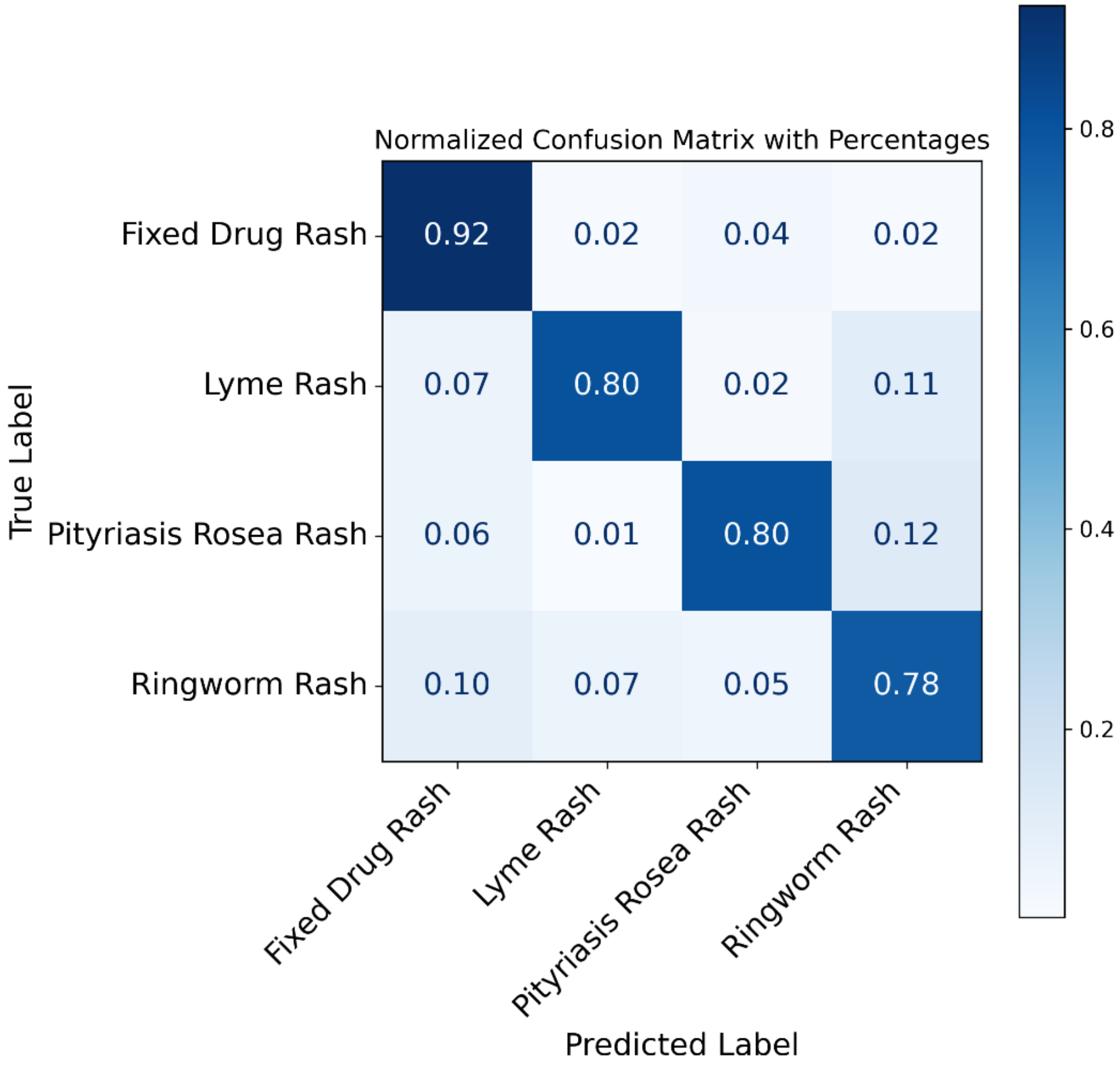

Figure 21: Photometric Transformation Matrix's test matrix that measured classification.

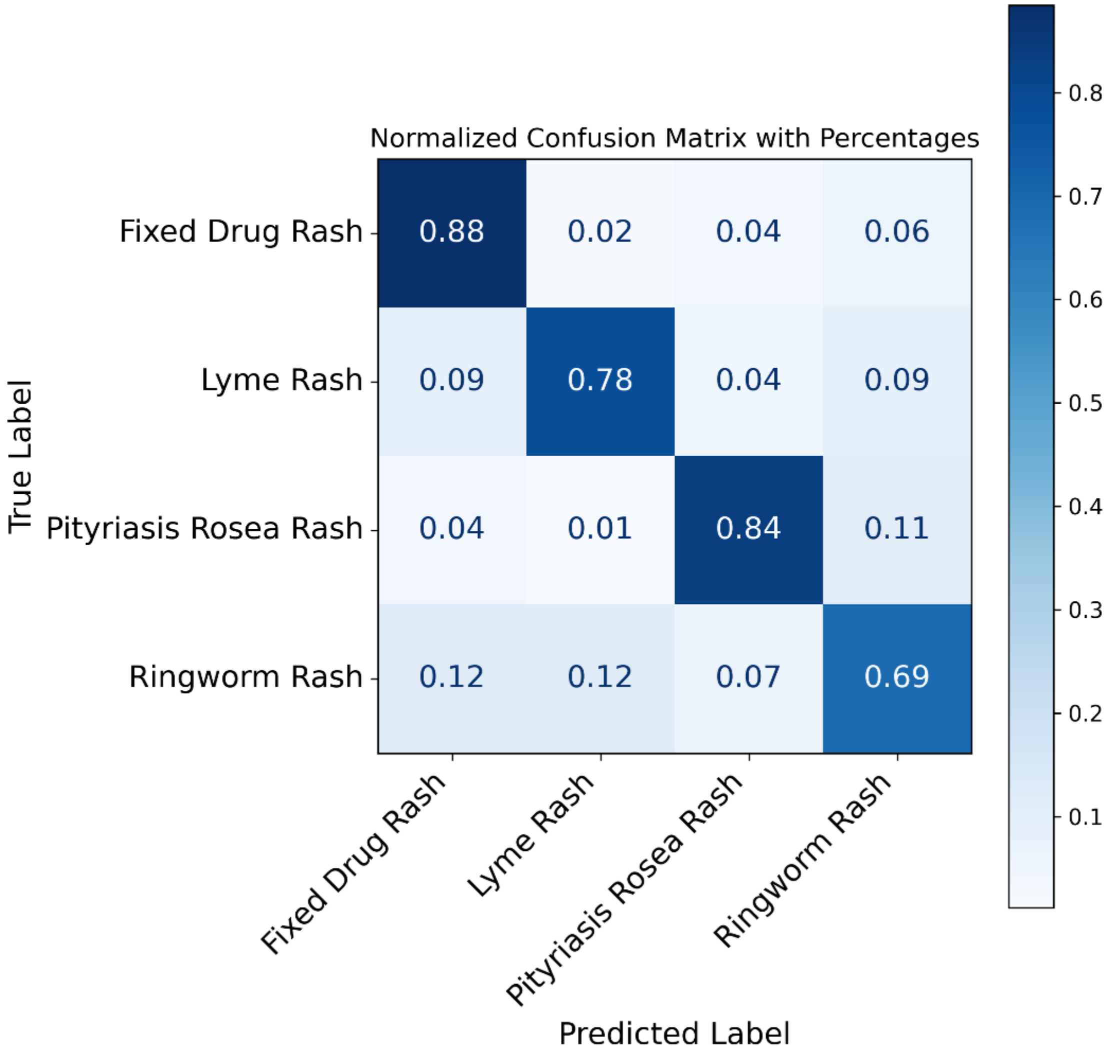

Figure 22: Elastic Deformation Matrix's test matrix that measured classification.

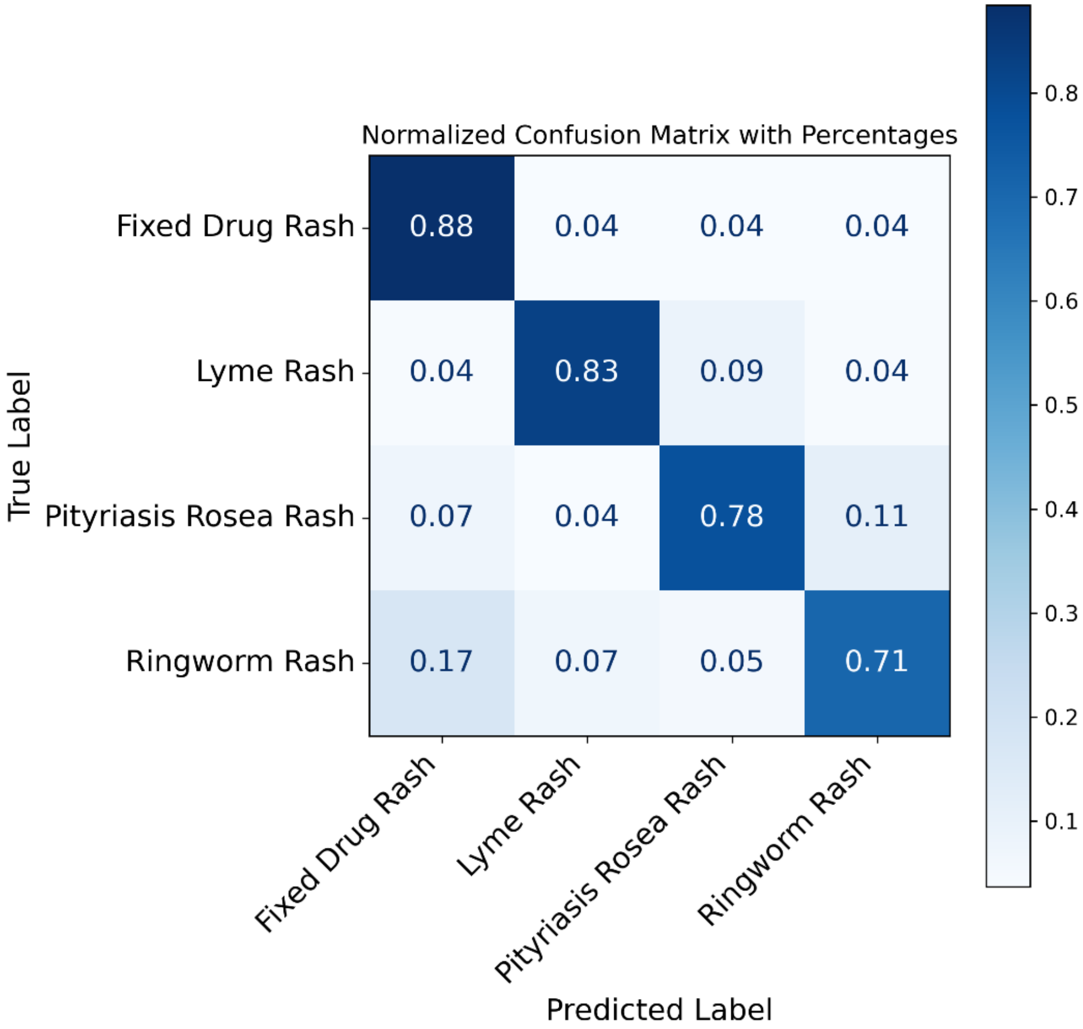

Figure 23: CLAHE Matrix's test matrix that measured classification.

The classification performance of the different augmentation methods varies across rash types, as displayed in the figures above. Photometric Transformation, Figure 19, has the highest classification for drug rashes which is 92%. CLAHE, Figure 21, has the highest classification for Lyme disease rash, 83%, while Regional MixUp, Figure 18, has the lowest classification with 72%. Regional MixUp, Figure 18, has the highest classification for Pityriasis Rosea rash 85%, while CLAHE has the lowest classification with 78%. Photometric Transformation, Figure 19, has the highest classification for Ringworm, 78%, while MixUp had the lowest classification with 62%.

Elastic Deformation, Figure 20, did not fail any class greatly, but also did not achieve the highest classification accuracy in any class. Overall, Photometric Transformation exhibits the best performance with highest classification accuracy within two classes.

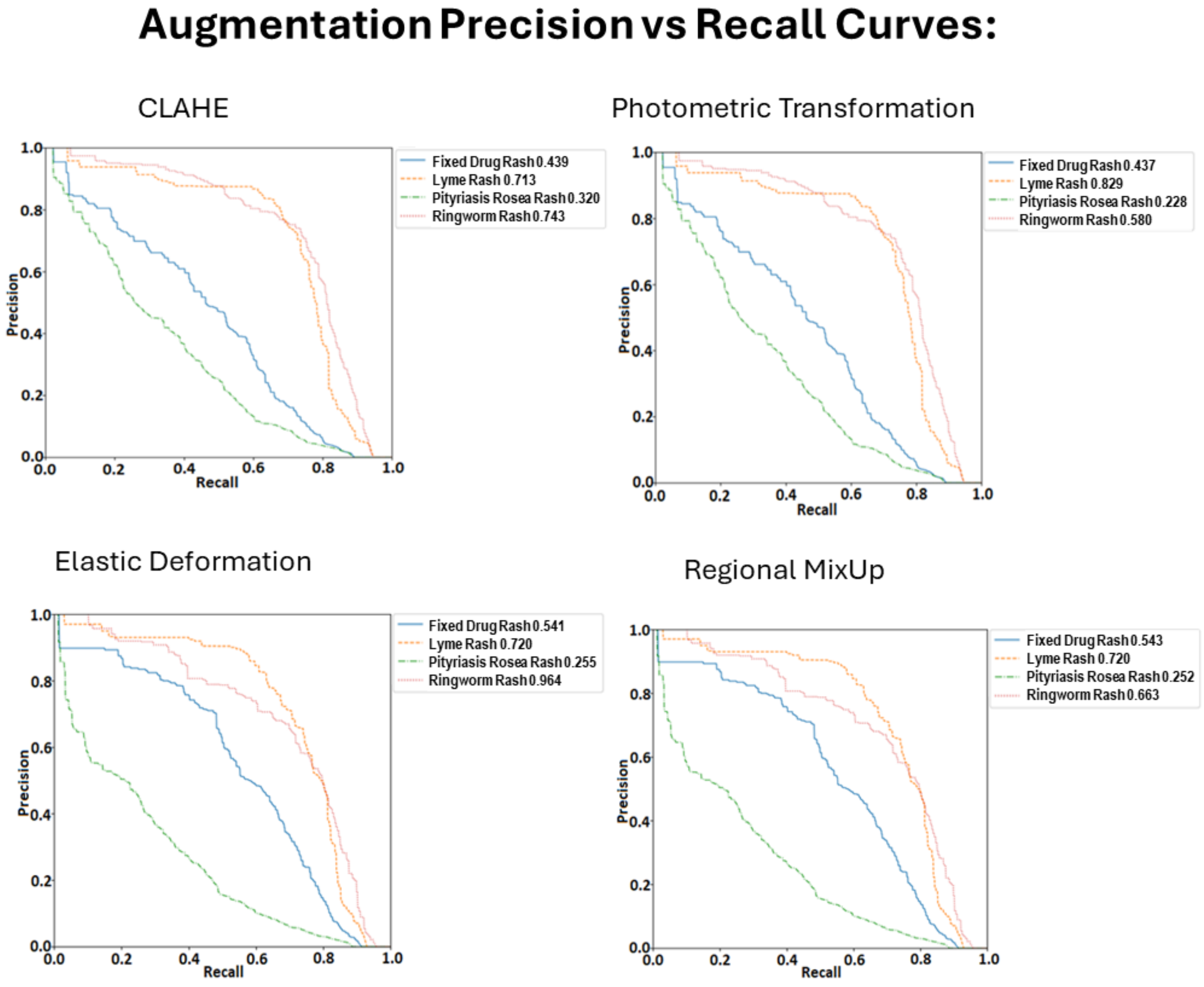

Figure 24: Precision and Recall Curves from training and validation of the different augmentation methods.

The precision-recall curves, Figure 24, showed similar patterns to each other during validation which is somewhat visible within the test matrices as well. Compared to the matrices, the precision-recall graph shows persistent struggle with the pityriasis rosea class. The curves showed the pityriasis rosea performing the lowest, between 22% to 32% compared to the other classes. Due to pityriasis rosea rashes being able to vary from a few spots on the back to a large break out, the model could be missing it all together, giving a partial reading or giving it a low confidence score. Across all the curves, the ringworm curve and the Lyme disease curve both performed similarly. Ringworm's lowest performing curve was 58% within photometric transformation. In the matrices, however, the confidence was turned down and the overall image is scored not just the box, but in the full image.

4.3 Combinations of Augmentations:

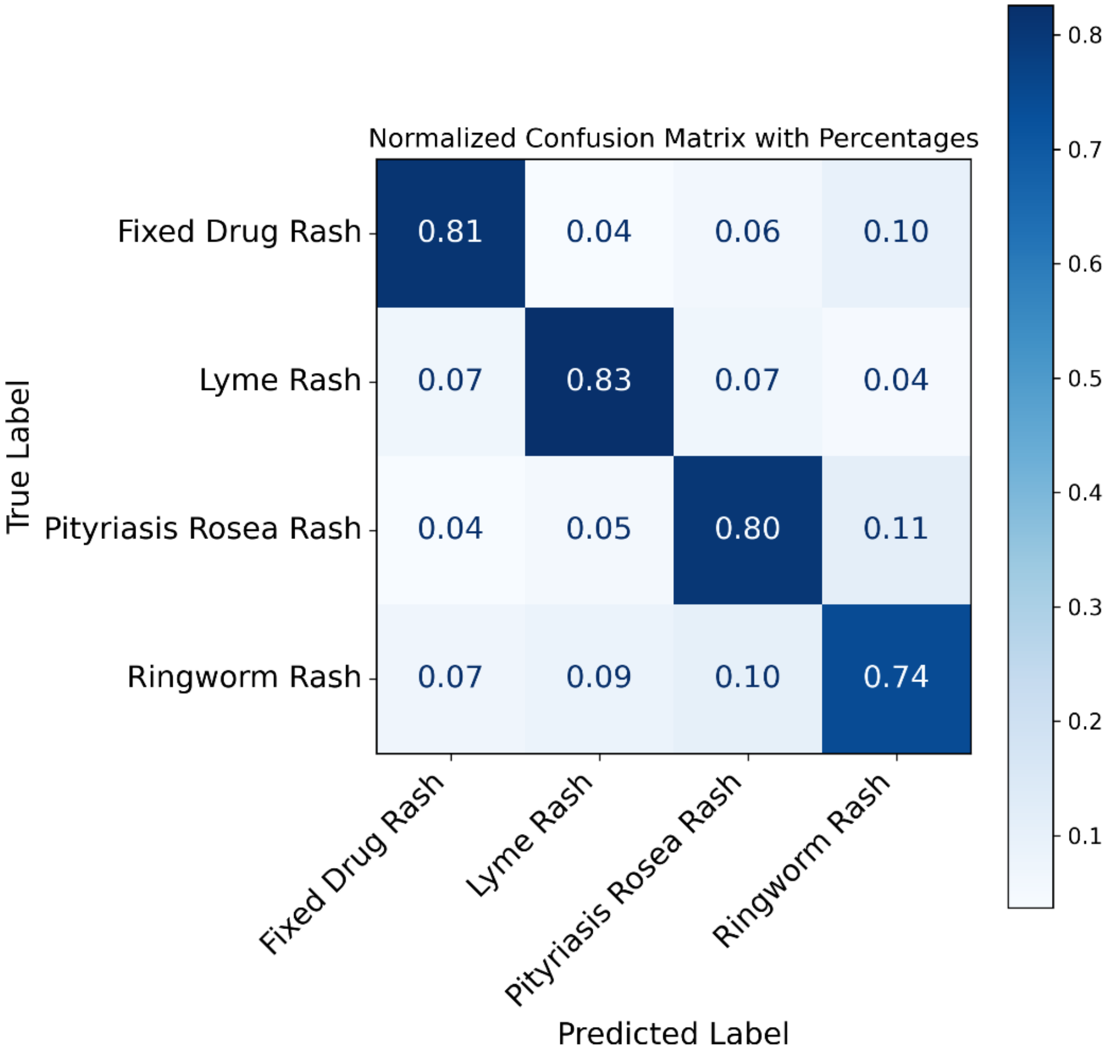

Figure 25: CLAHE and Photometric Transformation's test matrix that measured classification.

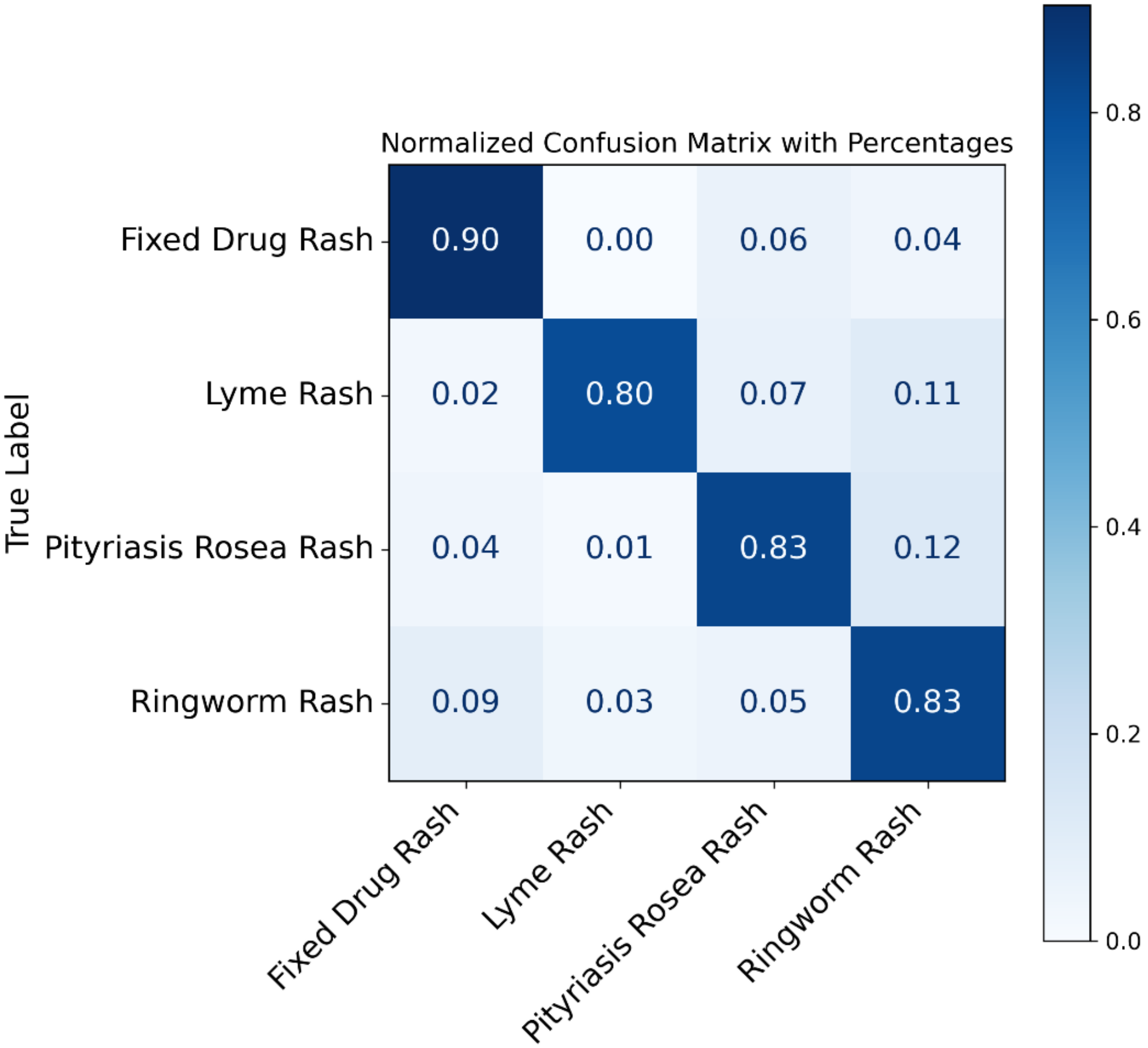

Figure 26: Photometric Transformation and Elastic Deformation's test matrix that measured classification.

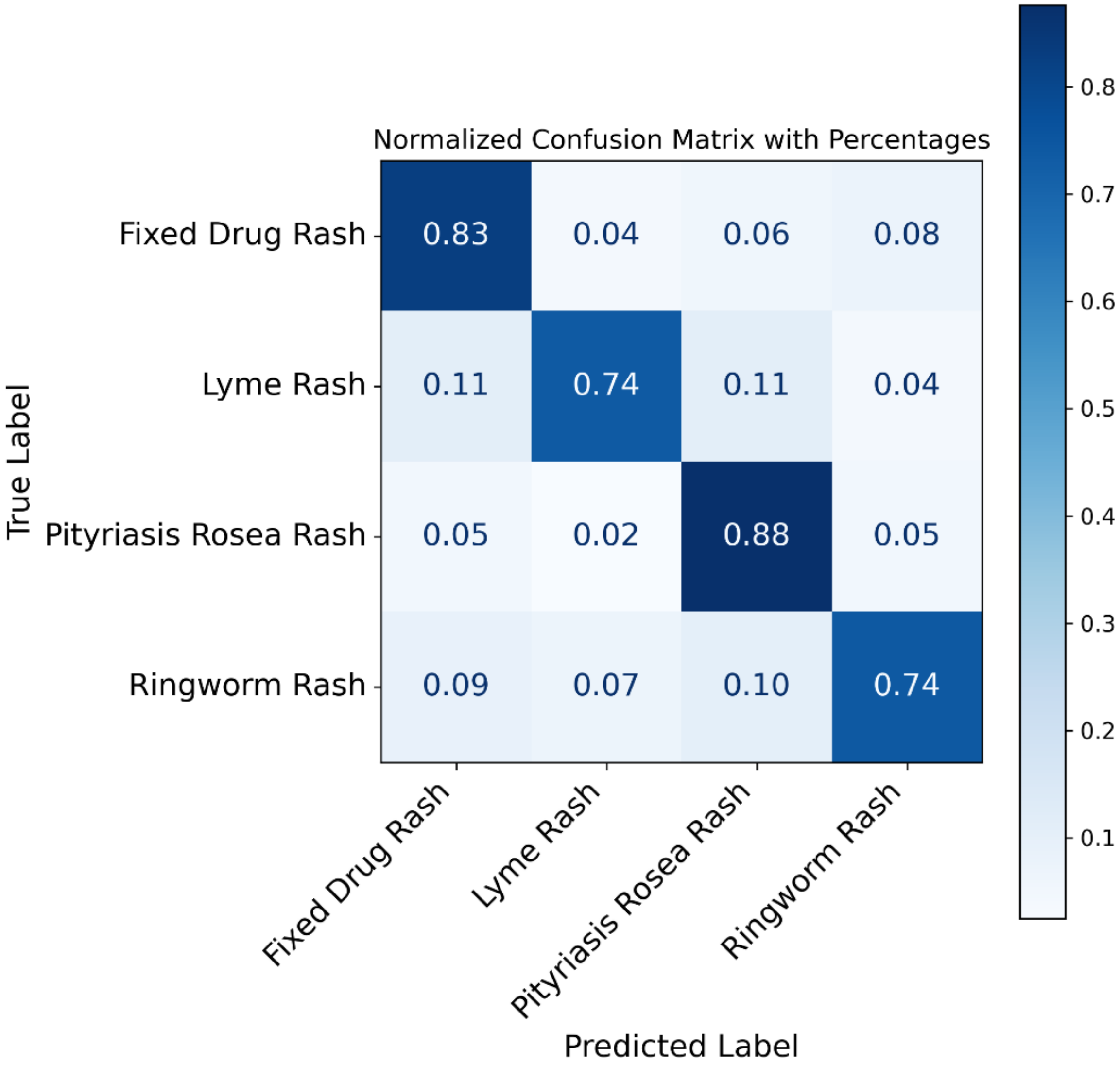

Figure 27: Regional MixUp and CLAHE's test matrix that measured classification.

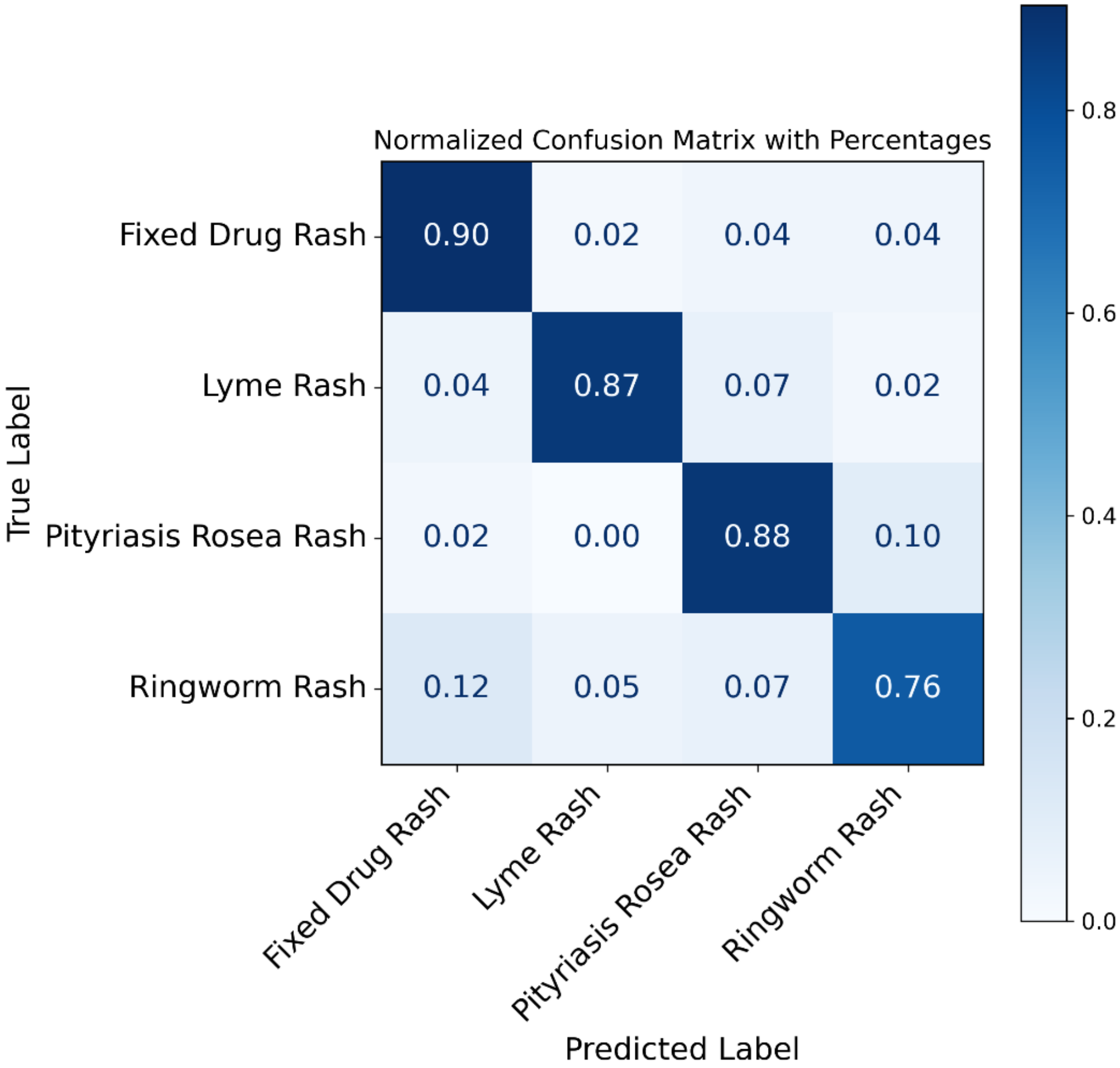

Figure 28: Funneled Combo’s test matrix that measured classification.

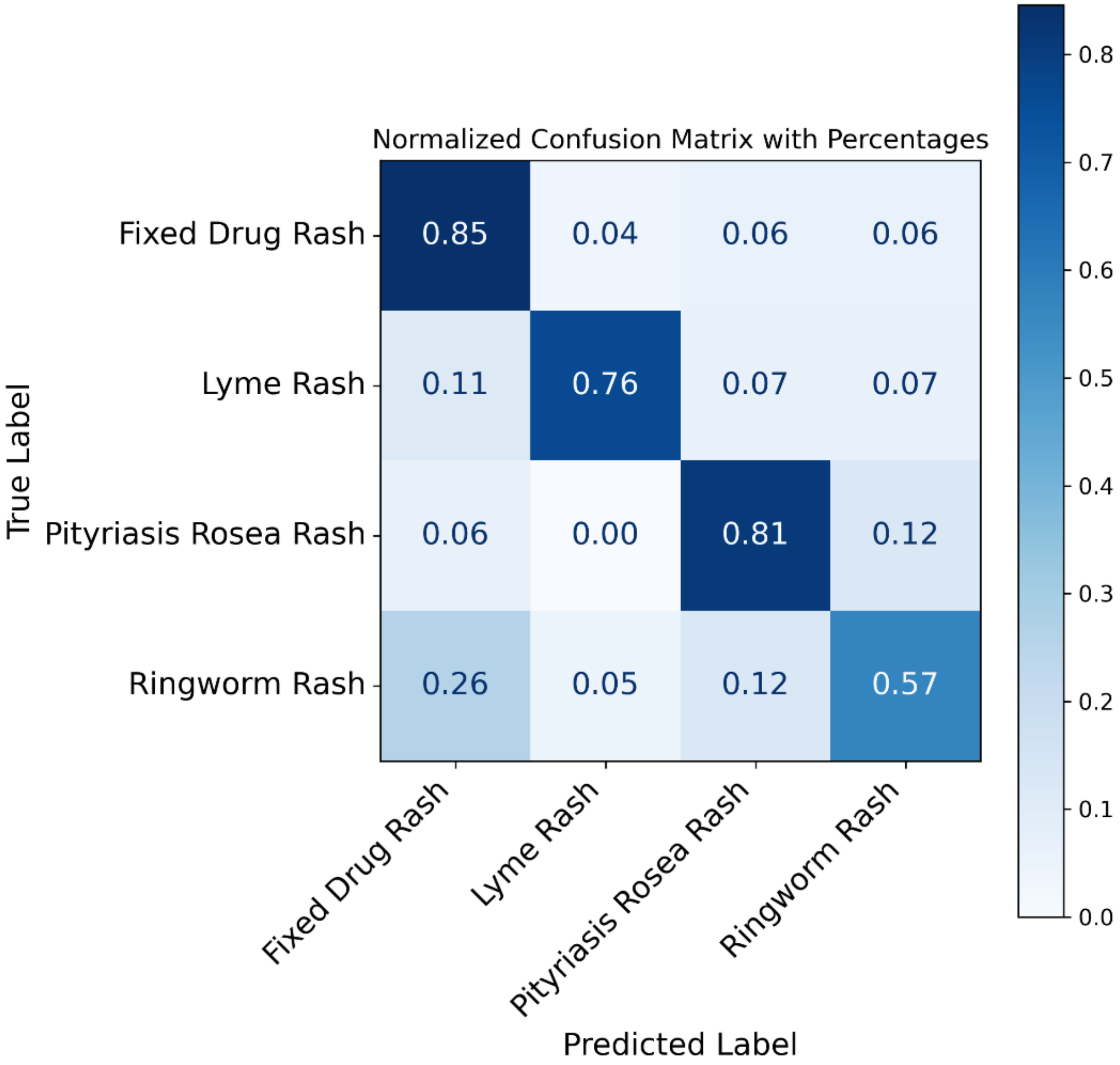

Figure 29: Combination Combo's test matrix that measured classification.

The classification performance of the combinations of the different augmentation methods varies across rash types, as displayed in the Figures 23-27 above. The Funnel Combo, Figure 26, contained the highest classification for Fixed Drug Rashes, 90%. The CLAHE and photometric transformation combination had the lowest classification rate for fixed drug rashes with 81%. The Combination Combo has the highest classification for Lyme disease rashes with 88% and the MixUp and CLAHE combination has the lowest classification with 74%. The Funneled combination and the MixUp and CLAHE combination consists of the highest classification score of 88% for pityriasis rosea rash. The CLAHE and photometric transformation had the lowest classification percent with 80%. The Funnel Combo and the Combination Combo both contained the highest classification for ringworm rashes with 76%. The lower classification score for ringworm was photometric transformation and elastic deformation with 69%. Overall, either combination containing all four augmentation methods performed the best. The worse performing combination was the CLAHE and photometric transformation as it got two of the lowest scores.

Combo Precision vs Recall Curves:

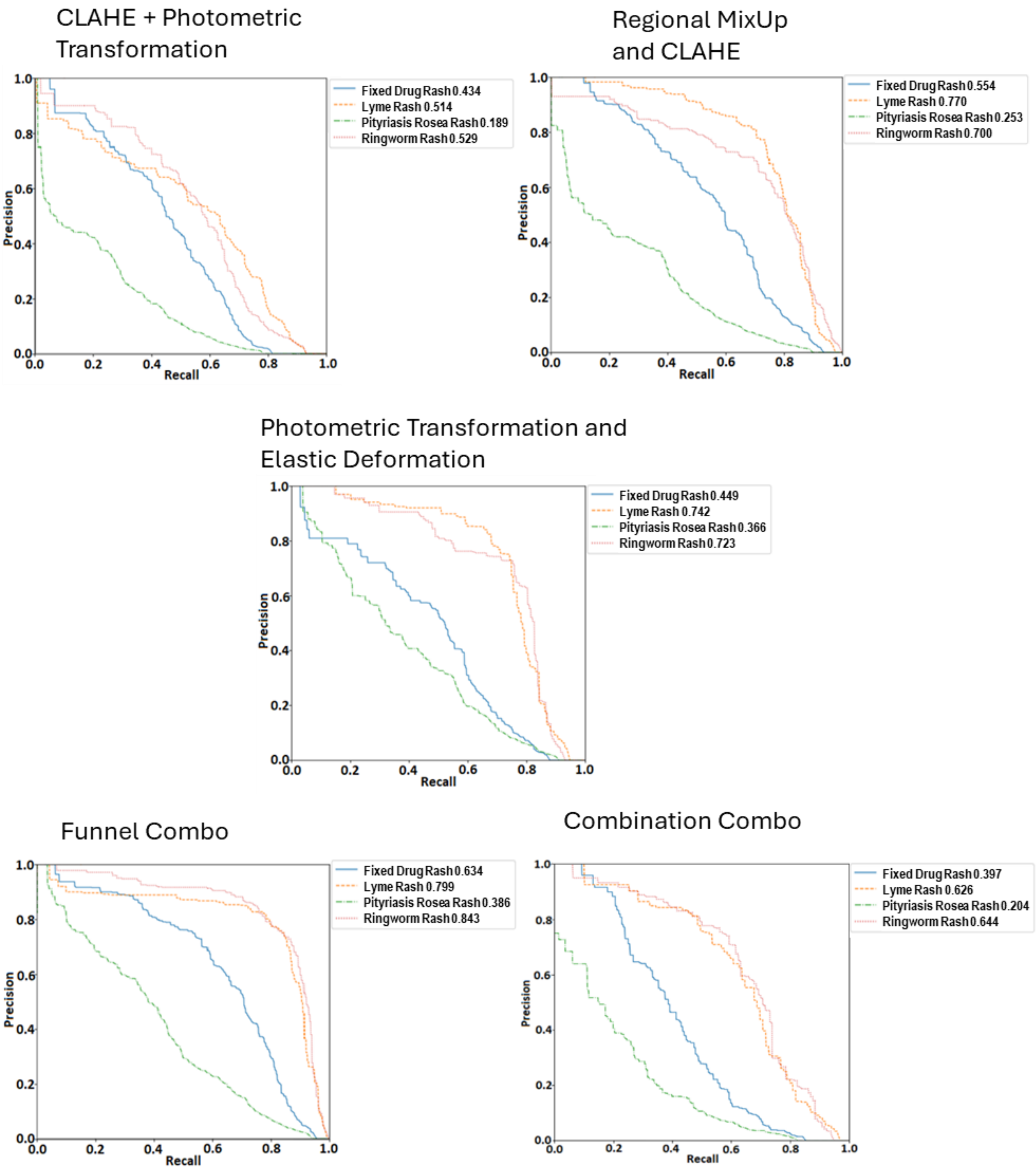

Figure 30: Precision and Recall Curves from the training and validation of the five different augmentation combinations.

The precision-recall curve graphs for the validation dataset, Figure 30, compare similarly to the tests' matrices, Figures 25-29. The funnel combo overall did the best compared to the other combination methods as seen in Figure 30. This result is also seen in the test matrix. Similar to the test, photometric transformation and elastic deformation were the next best. Although Figure 25 and Figure 27 display similar results, the precision-recall curves show that the Regional MixUp and CLAHE combination performed slightly better in all classifications. The combination combo's precision-recall curve showed the lowest performance of all combos trained.

4.4: Cross Validation Disclaimer:

The results in the sections above were cross evaluated to verify validity. Due to limited data, only one other test dataset was utilized in the evaluation. The results could exhibit variations in other test sets due to image quality and other factors. Therefore, generalization of the model to a completely new dataset cannot be guaranteed to achieve similar or high performance. The second test dataset contained similar biases, which were presented within the original dataset as well. In conclusion, the results support the findings within Chapter 4, but the results could vary and be expanded in future datasets with wider image diversity.

Chapter 5: Discussion

This chapter provides a comprehensive analysis of the challenges and opportunities in the future development of medical datasets and the training of artificial intelligence models for diagnostic purposes. It addresses the limitations of existing datasets, explores strategies for their enhancement, and considers the integration of advanced tools such as YOLOv8. Furthermore, it explores the idea of future community-driven data collection and innovative approaches to improve dataset diversity and model accuracy.

5.1 Analysis:

There are multiple training runs with different variables or an improved dataset, resulting in different outcomes (see Chapter 3: Methodology for more details). While reviewing the results of the 200 epochs, the model just reached convergence. The runs limited to 100 epochs or fewer yielded obscured results and failed to achieve full training potential. While runs containing 150 epochs generated clearer results but did not reach convergence and resulting less accurate. Values exceedingly more than 250 epochs had diminishing returns as the test and validation results did not improve significantly. This highlights the importance of finding the optimal training duration to obtain a balanced and effective model within YoloV7.

In the future version the unofficial version Yolov8 created by Ultralytics as it is the most up-to-date release. YoloV8 was not extensively documented nor fully released when the research started thus it was not utilized. At this point, YoloV8 is known to have a faster frames per second rate which would not apply value since the dataset is with images only [8].

5.2 Dataset Characteristics:

The dataset used was small which provided limited exposure to other elements to model while training. A future version should aim to broaden the dataset in size and image variety. One method is community-driven data collection which could be a reasonable method to gain new dataset components.

5.2.1 Future Dataset Enhancements:

Improving the dataset presents an opportunity to enhance its diversity and robustness by including a broader range of skin tones and age-related skin characteristics. The basic improvements would be to get a range of skin colors and varying levels of aging skin. With aging, loose muscles, and skin causes veins to become prominent. More complex images would contain skin markings such as scarring, tattoos, and other skin conditions such as vitiligo, the lost-of skin pigment, and eczema, a dry, itchy/painful, red patch on skin.

A way to collect data is if individuals using the app donate their images to a cloud base. Another way is to generate dataset images with a Generative Adversarial Networks (GAN) model or requesting them from a research center might be an option to expand the data greatly. GANs would use the original dataset to learn and generate new images to expand the original dataset. There are many ways to improve the dataset, and the model’s accuracy. Once the accuracy is at least 80%, the model is deployed with a simple frontend to make an application for a diagnosis application. The latest iteration, YoloV8, has been released and is the most recent advancement in the Yolo series.

5.2.2 Community-Driven Data Collection:

Citizen scientists could contribute to data collection and analysis of other data in the set. This approach not only increases the volume of available data but also fosters community engagement and collaboration in scientific endeavors.

There are many ways to interest citizen scientists. One way is through gamification which is a new expansion in computer science. It incorporates game-like elements into non-gaming contexts to encourage active contribution to data collection, analysis, and research through a fun means. Examples of gamification working are FoldIt[8] and StallCatchers[6]. Foldit is an in-depth computer game where players are challenged to find possible new proteins that will be tested in a lab. Stall catcher is a web-based game where individuals watch a video clip of the brain sending a signal and determine if the signal is interrupted at any point causing a stall. The goal is to learn more about the brain and eventually learn more about Alzheimer's disease.

5.3 Database Selection:

Apart from the dataset primarily utilized and the updated version compiled by the same author, there are no other public datasets on Lyme disease that are currently accessible. Different research groups requiring public datasets must either use the same pre-existing datasets due to limited availability or have to develop their own. The lack of public datasets could contribute to numerous challenges including regulatory constraints and limited resources. Limited resources could be financial constraints, time constraints or lack of data able to be collected. The number of high-quality medical images the hospitals can procure within a reasonable timeframe may not be enough to justify the dataset's development. Regulatory constraints may be preventing some of the access to medical data, however privacy is important to protect the patients and should be respected. Additional factors are that some institutions are not willing or able to release their privately developed dataset.

Chapter 6: Conclusion

This thesis explored multiple augmentation methods and combinations to best optimize a dataset for training artificial intelligence models using YoloV7 on Lyme disease rashes. The limited availability of high-quality medical datasets remains a challenge as they are often difficult to obtain. Nonetheless, this thesis demonstrated the possibilities that augmentations can offer to improve a lower-quality dataset. The augmentations increase the size of the dataset and offer a way to improve the images themselves.

While there are many augmentation and enhancement methods to choose from depending on the dataset type, this thesis applied CLAHE, Reginal MixUp, Photometric Transformation, and Elastic Deformation. Photometric Transformation performed the best as a single augmentation while Regional MixUp performed the worst. The combinations with all four augmentation types tended to do the best compared to the ones with just two augmentation methods combined. For Lyme disease the Funneled method achieved the highest performance compared to the other methods. One of the next steps is to expand the database's variety of skin types, other markings like tattoos, and expand to other conditions.

References

[1] ADA. 2024. Ada: Your health companion. Accessed May 13, 2025 from https://ada.com/

[2] Centers for Disease Control and Prevention. 2025. Lyme disease data and statistics. Accessed January 6, 2025, from https://www.cdc.gov/lyme/data-research/facts-stats/index.html

[3] Centers for Disease Control and Prevention. 2024. Signs and symptoms of untreated Lyme disease. Access March 13, 2025 from https://www.cdc.gov/lyme/signs-symptoms/index.html

[4] Dwyer, B. 2020. When Should I Auto-Orient My Images? Roboflow Blog. Accessed January 6, 2025 from https://blog.roboflow.com/exif-auto-orientation/.

[5] iDoc24 Inc. 2024. First Derm: Online Dermatology Service. Accessed May 13, 2025 from https://www.firstderm.com/

[6] Human Computation Institute. 2019. About Stall Catchers. Stall Catchers, Accessed March13, 2025, from https://stallcatchers.com/

[7] Johns Hopkins Lyme Disease Research Center. 2020. Lyme disease signs & symptoms. Accessed February 12, 2025 from Lyme Disease Symptoms : Johns Hopkins Lyme Disease Research Center

[8] Kleffner, R., Flatten, J., Leaver-Fay, A., Baker, D., Siegel, J. B., Khatib, F., & Cooper, S. 2017. Foldit Standalone: A video game-derived protein structure manipulation interface using Rosetta. Accessed November 12, 2025 from https://doi.org/10.1093/bioinformatics/btx283

[9] Lee, C. 2020. First Derm Launches Skin Disease Search Engine. American Spa. Accessed March 13, 2025 from https://www.americanspa.com/medical-spa/first-derm-launches-skin-disease-search-engine

[10]Mayo Clinic. 2023. Lyme disease: Diagnosis and treatment. Accessed March 13, 2025 from https://www.mayoclinic.org/diseases-conditions/lyme-disease/diagnosis-treatment/drc-20374655

[11]New Zealand Dermatological Society. 2025. *Image Library*. DermNet. Accessed July 20, 2025, from https://dermnetnz.org/images

[12] Saha, S., & Garain, U. 2024. Region MixUp. ArXiv. Accessed January 3, 2025 from

https://doi.org/10.48550/arXiv.2409.15028

[13]Skinsight, Rochester (NY): 2025. VisualDx; Accessed November 2024 from: https://skinsight.com/

[14] Thibodeau, 2023. Medical Diagnosis at a Snap of the Camera

[15] van Tulder, G. 2018. elasticdeform [Computer software]. GitHub. Accessed March 13, 2025, from https://github.com/gvtulder/elasticdeform

[16]VisualDx. 2025. *Diagnosis*. VisualDx. Accessed July 20, 2025, from https://www.visualdx.com/visualdx/diagnosis/

[17] Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696. Accessed January 3, 2025, from https://arxiv.org/pdf/2207.02696.

[18] Wong, K.-Y. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. GitHub repository. Accessed January 3, 2025, from https://github.com/WongKinYiu/yolov7.

[19] Zhang, Edward. 2022. "Lyme Disease Rashes." Kaggle, Accessed January 3, 2025. from https://www.kaggle.com/datasets/sshikamaru/lyme-disease-rashes.

[20] Zhang, Edward. 2022. "Lyme Disease Erythema Migrans Rashes." Kaggle, Accessed January 3, 2025. from https://www.kaggle.com/datasets/sshikamaru/lyme-disease-rashes/data

[21] Zhang, L., Yip, A. M., & Tan, C. L. 2007. Photometric and geometric restoration of document images using inpainting and shape-from-shading. Proceedings of the 22nd AAAI Conference on Artificial Intelligence, 1177–1182. Accessed January 7, 2025 https://aaai.org/Papers/AAAI/2007/AAAI07-178.pdf.

[22] Zuiderveld, K. J. (1994). Contrast Limited Adaptive Histogram Equalization. In P. S. Heckbert (Ed.), Graphics Gems IV (pp. 474–485). Academic Press. Accessed March 13, 2025 from https://www.cse.unr.edu/~bebis/CS474/StudentPaperPresentations/1994%20-%20CLAHE.pdf