INSTITUIÇÃO EXECUTORA
Centro de Informática UFPE
FADE UFPE
COORDENADORA
MCTI FUTURO
Softex
APOIO
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

ACORDO DE PARCERIA Nº 05/23 FADE/UFPE/SOFTEX - RESIDENCIAL IC13 (CONVÊNIO Nº 02/2023 UFPE) 23076.125530/2022-28

Lux.AI

Data pre-processing and augmentation on dermoscopy images for skin lesion classification

---

1. INTRODUCTION

In recent years, AI systems have been extensively researched to support medical professionals in diagnostic processes [1]. While state-of-the-art methods show promising results in experimental settings [2,3], their application in real-world clinical environments faces several challenges [4]. In clinical dermatology, visual features are crucial for assessing the risk of skin lesions. However, image-based classification methods encounter significant difficulties in clinical scenarios, primarily due to sample variance in captured images. Environmental factors, such as lighting conditions, can introduce noise and artifacts that affect the final representation [5]. Lighting variations, influenced by indoor/outdoor settings, time of day, and geographical location, can alter the perceived color of lesions—a critical feature for determining malignancy or benignity.

Color Constancy

As an alternative solution, **color constancy** is the ability to measure object colors independently of the light source, ensuring consistent color representation across varied lighting conditions. This consistency is critical in clinical dermatology, as variations in captured lesion colors can impact key features used to assess malignancy. Despite its importance, most studies based on deep learning either do not utilize color constancy pre-processing or fail to disclose its use, although it is considered a promising approach for handling variations in image captures.

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

As Goyal et al. [6] indicate, publicly available clinical and dermoscopic datasets often contain images acquired under diverse settings, and inconsistencies in scene illumination can reduce AI algorithm performance. Techniques like shades of gray or max-RGB [7,8] are examples of color constancy methods that can enhance algorithmic accuracy. **Figure 1** illustrates how color constancy can normalize the digital representation of lesions, improving the reliability of diagnostic models.

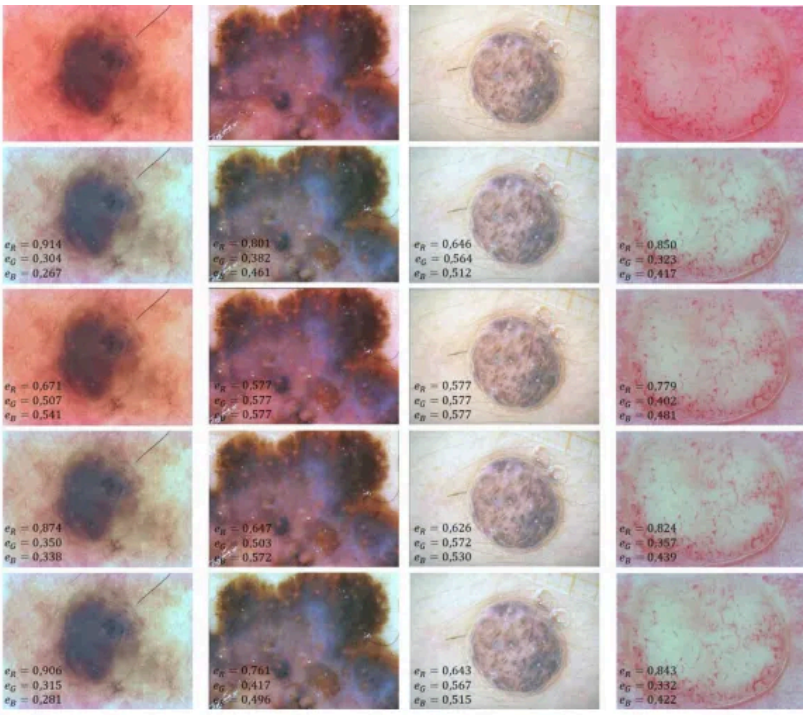


Figure 01 - Shadow of gray pre-processing is used to normalize the illumination and lighting effect on dermoscopic skin lesion images. Figure obtained from [1].

2. RELATED WORKS

To investigate how color generalization is applied in dermatology imagery, we conducted a brief literature review on preprocessing techniques, with a focus on color constancy algorithms that 3.6address lighting variations. Barata et al. [7] outline four primary color constancy techniques widely used in dermatology, serving as foundational references for studies employing traditional learning algorithms. In these methods, the illuminant components (e_c) of an **RGB** image (I) are defined as $e = [e_r, e_g, e_b]^T$, allowing the computation of pixel values (x) by adjusting each channel $c \in (r, g, b)$, based on the illuminant values, as described in the techniques below.

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

- **Gray world:** Assumes that, on average, the image's color should be a neutral gray if the illuminant were white, as shown in **Equation 1**.

$$\frac{\int I_c(x)dx}{\int dx} = ke_c \qquad (1)$$

- **max-RGB:** The brightness pixel in each color channel should ideally be a reflection of the illuminant color (**Equation 2**).

$$max_x I_c(x) = ke_c \qquad (2)$$

- **Shades of Gray:** Generalizes Gray World by using Minkowski norm parametrized by a value p (**Equation 3**).

$$\left(\frac{\int (I_c(x))^p dx}{\int dx}\right)^{\frac{1}{p}} = ke_c \quad (3)$$

- **General Gray World:** Extends the shades of gray by first smoothing the image with a Gaussian Filter (**Equation 4**).

$$\left(\frac{\int (I_c^{\sigma}(x))^p dx}{\int dx}\right)^{\frac{1}{p}} = ke_c \quad (4)$$

Using functions defined in Equations 1–4, we transform the entire image (I) by estimating the value of each illuminant component (e_c); Based on these estimated coefficients, we model the transformation using the von Kries diagonal model [9], adjusting each pixel to appear as it would under a canonical light source, assumed here to be ideal white light.

In contrast, [10] proposes that the average edge difference within a scene is achromatic, introducing the gray-edge and higher-order gray-edge algorithms. Unlike the previously described algorithms, which are computed based on the zero-order structure of images, gray-edge techniques rely on the derivative structure of images (as shown in **Equation 5**).

$$\left(\int \left|\frac{\partial^n f^{\sigma}(x)}{\partial x^n}\right|^p dx\right)^{\frac{1}{p}} = ke^{n,p,\sigma} \quad (5)$$

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

Here, n is the parameter defining the order of the image structure, distinguishing gray-world from gray-edge. Gray-world operates on RGB values, while gray-edge utilizes spatial derivatives of different orders. The parameter p represents the Minkowski norm, which determines the relative weights of various measurements used to estimate the final illuminant color. Additionally, the parameter σ represents the scale of local measurements. For first or higher-order estimations, this local scale is combined with a differentiation operation, typically computed using the Gaussian derivative.

Recent studies, such as [11], apply advanced heuristics to reduce variability in dermoscopy images while enhancing overall contrast. By utilizing learning algorithms like GAN Pix2Pix, the model combines generative capabilities with the ability to learn higher-order features specific to dermoscopic images. In this setup, the generator employs a ResNet architecture, while the discriminator uses a PatchGAN model (illustrated in **Figures 2a** and **2b**).

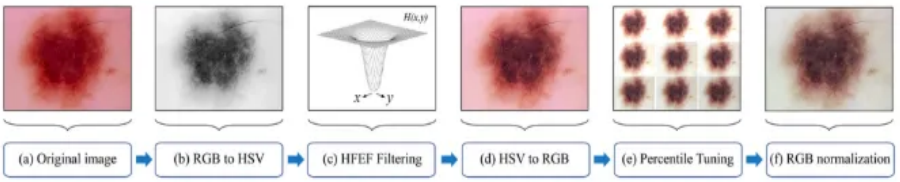


Figure 02a - Steps applied to normalize illuminant source. Figure obtained from [11]

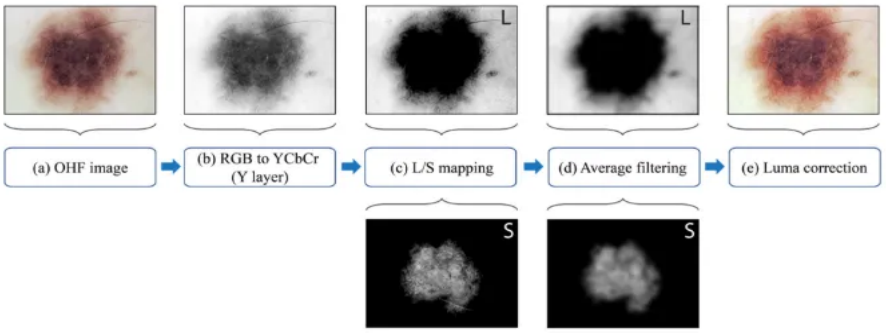


Figure 02b - Light/Shadow correction to optimize image brightness. Figure obtained from [11]

While color constancy has primarily been applied as a preprocessing method, Galdran et al. [12] introduced it as a dataset augmentation technique. After estimating the illuminant from a set of images, a random illuminant-corrected image can be adjusted using another randomly selected illuminant component, thereby generating a new variant. This process is illustrated in **Figure 3**.

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

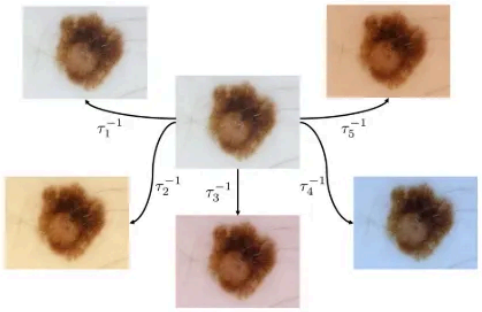


Figure 03 - Color-casting of a white-balanced image with different illuminants. Image obtained from [12]

3. METHODOLOGY

We can implement color constancy algorithms either as a **preprocessing** technique [7,8], applying transformations to all samples before loading them into the training batch, or as an **augmentation** technique [12], where transformations are randomly applied on the fly to diversify the dataset during training.

Since our scope includes clinical dermatology images, we aim to investigate how lighting conditions impact generalization across different datasets. We have chosen to assess color constancy as an augmentation rather than just a preprocessing step. We will evaluate the effects of incorporating color constancy as a complementary augmentation in our training pipeline, implemented in PyTorch[1].

Our investigation focused on five color constancy techniques commonly used in dermatology, as referenced in [7, 8, 10]: **Grey-World (1)**, **max-RGB (2)**, **Shades of Gray (3)**, **General Grey-World (4)**, and **Grey-edge (5)**. By integrating these methods, we can incorporate their functions into the training pipeline, which is based on torchvision.Compose[2]. This allows us to apply custom sequential augmentations[3] to the method's output before transforming it into torch tensors.

As a baseline, we used various **CNN** architectures trained on the **ISIC18** [13] dataset without applying color constancy, using only traditional augmentations as outlined in **Figure 4**. Next, we integrated color constancy with these augmentations for comparison.

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

```python
train_transforms = transforms.Compose([
    # Preprocessing
    transforms.Resize((255, 255)),
    transforms.CenterCrop((224, 224)),

    # Base augmentations
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomVerticalFlip(p=0.5),
    transforms.RandomRotation(degrees=30),
    transforms.ColorJitter(brightness=0.5, contrast=0.5),

    # Convert to tensor and normalize
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
valid_transforms = transforms.Compose([
    # Preprocessing
    transforms.Resize((255, 255)),
    transforms.CenterCrop((224, 224)),

    # Convert to tensor and normalize
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

Figure 04 - Augmentations used to train the BASE model variations. Image transformations from torchvision are used. We apply simple augmentation that does not alter semantic information of the images, mainly related to light and colors.

Although **ISIC18** is a **dermoscopic** dataset, primarily featuring images captured by specialized devices, it serves our purpose well by providing a controlled scenario for investigating the improvements color constancy can bring in addressing lighting artifacts.

We trained 10 different architectures which includes different structures **(i.e., convnext-tiny, convnext-small, convnext-base, densenet121, densenet161, densenet201, efficientNet-B0, ResNet101, ResNet152, InceptionV3**). Architectures trained without color constancy are referred to as **BASE_[M]** where **M** is the architecture used on training, while **COLOR_CONSTANCY_[M]** refers to the same architecture but trained on color constancy.

4. EXPERIMENTS AND DISCUSSIONS

Testing Implementation

We implemented the algorithms outlined in **Equations 1–5** [8,10]. To validate the effectiveness of these transformations, we plotted the color distributions for both the original and corrected images. Examples of the Gray-World and Shades of Gray transformations can be seen in **Figures 5a**, **5b**, and **5c**.

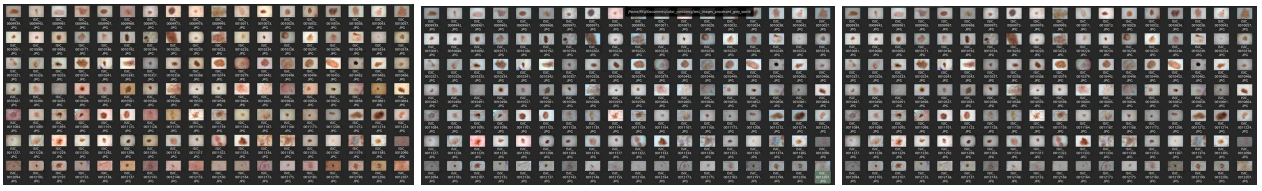


Figure 05a - Image samples obtained from original dataset, gray-world, and shades of gray from left to right, respectively.

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

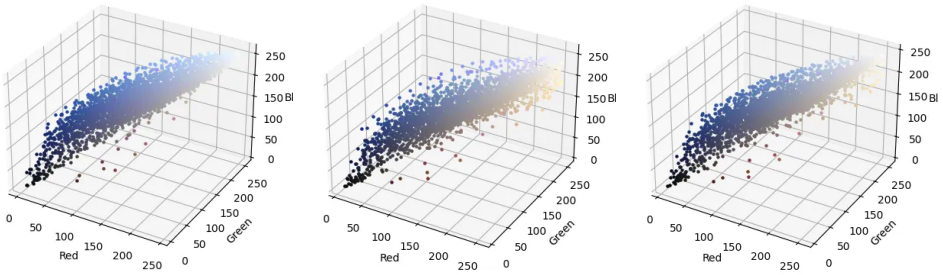


Figure 05b - Scatter plot for color distribution to the original dataset, gray-world, and shades of gray, respectively. Each image was quantized to 25 colors, which were all plotted in this scatter.

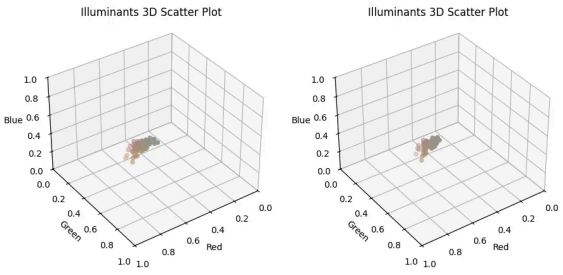
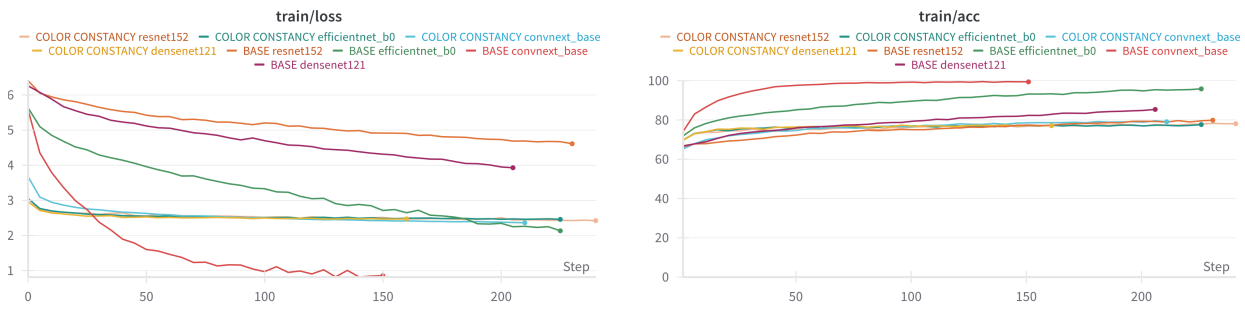


Figure 05c - Scatter plot for the estimated illuminants. Left shows the illuminant components for the gray world and the right for the shades of gray.

Implementing as augmentations

Since random augmentations are applied on the fly during training, we can effectively increase the number of images used by simply extending the number of epochs, as suggested by [12]. The augmentations implemented adhere to the methods outlined in **Figure 4**.



1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

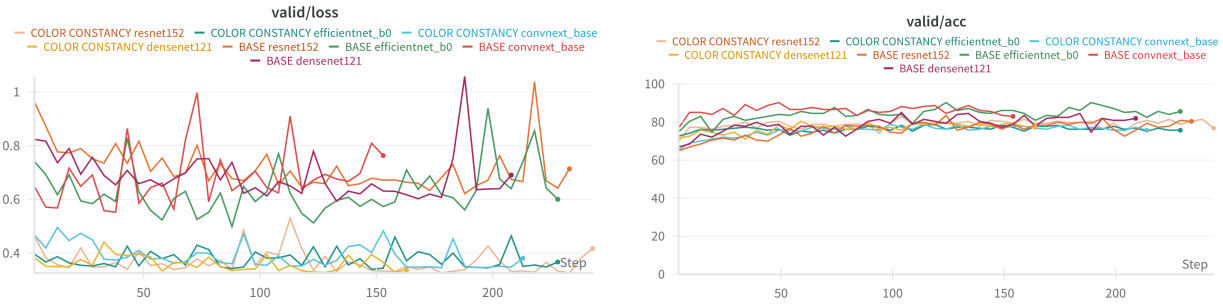


Figure 6 - The results from training and testing models are presented for both the baseline (**BASE**) and the models utilizing color constancy (**COLOR_CONSTANCY**). The top row displays the training loss and accuracy (from left to right), while the bottom row shows the validation loss and accuracy in the same order. For clarity in visualization, we have included only the most significant models that capture the overall experimental behaviors across different architectures. The results pertain to **ResNet152**, **EfficientNet-B0**, **ConvNext-base**, and **DenseNet121**.

As shown in the results (**Figure 6**), color constancy did not significantly improve any metrics; in fact, it produced worse outcomes than the previous augmentations for most models. While preprocessing may create the appearance of similar images, this can hinder the models' ability to generalize and may prevent them from learning essential features and causing overfitting , decreasing the loss value quickly but not improving accuracy scores. Although color constancy is widely used in classical learning algorithms, it negatively impacts the generalization capacity of neural network-based models.

Color constancy aims to normalize image attributes, which can be beneficial for hand-crafted features by adjusting them to align with the current distribution. However, when working with deeper models, we have less control over these adjustments. Our experiments confirm this observation. Consequently, alternative augmentation methods may be more suitable for clinical dermatology, as the color attributes in RGB images can significantly influence the models' accuracy.

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2

5. REFERENCES

[1] Pinto-Coelho, Luís. "How artificial intelligence is shaping medical imaging technology: A survey of innovations and applications." *Bioengineering* 10.12 (2023): 1435.

[2] Cai, Lei, Jingyang Gao, and Di Zhao. "A review of the application of deep learning in medical image classification and segmentation." *Annals of translational medicine* 8.11 (2020).

[3] Chan, Heang-Ping, et al. "Deep learning in medical image analysis." *Deep learning in medical image analysis: challenges and applications* (2020): 3-21.

[4] Wu, Eric, et al. "How medical AI devices are evaluated: limitations and recommendations from an analysis of FDA approvals." *Nature Medicine* 27.4 (2021): 582-584.

[5] Tschandl, Philipp. "Risk of bias and error from data sets used for dermatologic artificial intelligence." *JAMA dermatology* 157.11 (2021): 1271-1273.

[6] Goyal, Manu, et al. "Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities." *Computers in biology and medicine* 127 (2020): 104065.

[7] Barata, Catarina, M. Emre Celebi, and Jorge S. Marques. "Improving dermoscopy image classification using color constancy." *IEEE journal of biomedical and health informatics* 19.3 (2014): 1146-1152.

[8] Hua Ng, Jia, et al. "The effect of color constancy algorithms on semantic segmentation of skin lesions." *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Vol. 10953. SPIE, 2019.

[9] KRIES, VON. "Influence of adaptation on the effects produced by luminous stimuli." *handbuch der Physiologie des Menschen.* 3 (1905): 109-282.

[10] Van De Weijer, Joost, Theo Gevers, and Arjan Gijsenij. "Edge-based color constancy." *IEEE Transactions on image processing* 16.9 (2007): 2207-2214.

[11] Salvi, Massimo, et al. "DermoCC-GAN: A new approach for standardizing dermatological images using generative adversarial networks." *Computer Methods and Programs in Biomedicine* 225 (2022): 107040.

[12] Galdran, Adrian, et al. "Data-driven color augmentation techniques for deep skin image analysis." *arXiv preprint arXiv:1703.03702* (2017).

[13] Codella, Noel, et al. "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)." *arXiv preprint arXiv:1902.03368* (2019).

1. https://pytorch.org/
2. https://pytorch.org/vision/stable/index.html
3. https://medium.com/@sergei740/simple-guide-to-custom-pytorch-transformations-d6bdef5f8ba2