Reproducibility Report
Exploring Log RGB Data for Image Classification

Shubham Kukadiya - 202411066, Jaydeep Darji - 202411029
Khelan Bhatt - 202411025, Viren Bhalgamiya - 202411003

1 Scope of Reproducibility

This report aims to reproduce and validate the claims that different image representations affect the performance of convolutional neural networks (CNNs) in binary classification tasks. Specifically, the experiment compares image formats—JPG (sRGB), Linear RGB, Log RGB, and Pseudo-Log—and evaluates their impact on the classification accuracy between "Swedish Fish" and "No Fish" using models such as FishNet64, ResNet18 (modified), and DenseNet121.

2 2 Methodology

The methodology shows how different image representations affect classification performance using convolutional neural networks. The workflow was structured into three major stages: data preparation, model training, and performance evaluation.

2.1 2.1 Dataset Preparation and Image Conversion

We began with a image dataset organized into three subsets: `training`, `validation`, and `test`. Each subset included two categories—`SwedishFish` and `NoFish`—and all images were initially stored in the standard JPG (sRGB) format.

To explore the effect of image representation on model performance, we converted the original JPG images into two additional formats: Linear RGB and Log RGB.

- **Linear RGB conversion** involved reversing the sRGB gamma curve to approximate linear light intensity. This was done using a piecewise inverse gamma transformation and normalized back to an 8-bit scale.
- **Log RGB conversion** applied a logarithmic transformation to the image data to compress high-intensity values and enhance contrast in darker regions. Pixel values were adjusted to avoid undefined operations and re-normalized for storage and model compatibility.

These conversions were performed programmatically across all data splits and categories, preserving the original folder structure to maintain experimental consistency.

2.2 2.2 Model Training

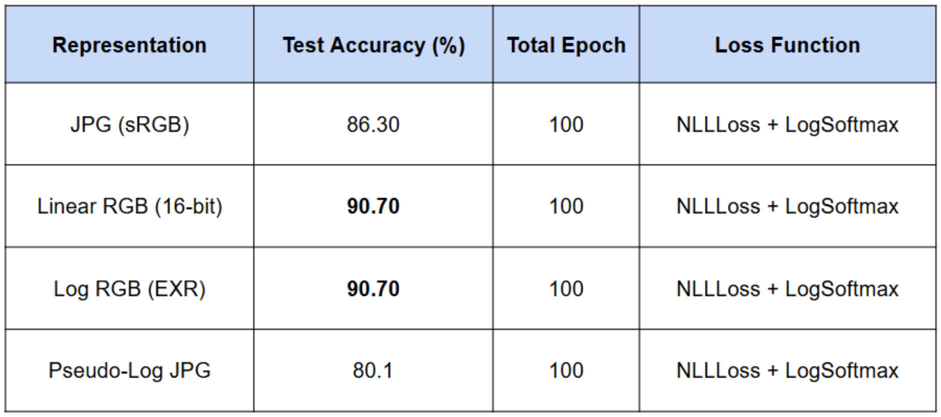
Once the datasets in JPG, Linear RGB, and Log RGB formats were ready, we proceeded to model training using three different CNN architectures. The goal was to compare how each model responded to the varying input representations.

- **FishNet64:** A custom, lightweight CNN designed specifically for 64×64 input images. It contains three convolutional layers followed by max-pooling, dropout, and fully connected layers. This model was trained for 100 epochs using the Adam optimizer and Negative Log-Likelihood loss with a LogSoftmax output layer.
- **ResNet18 (Modified):** A variation of the standard ResNet18 with additional dropout and batch normalization layers. The input images were resized to 224×224 pixels, and the model was trained for 30 epochs using CrossEntropyLoss and a learning rate scheduler.
- **DenseNet121:** A deeper architecture pretrained on ImageNet and fine-tuned for binary classification by replacing its final layer. Like ResNet18, it used 224×224 input size and was trained for 10 epochs with CrossEntropyLoss.

All models were trained independently on each representation type (JPG, Linear RGB, Log RGB), resulting in a total of nine experiments.

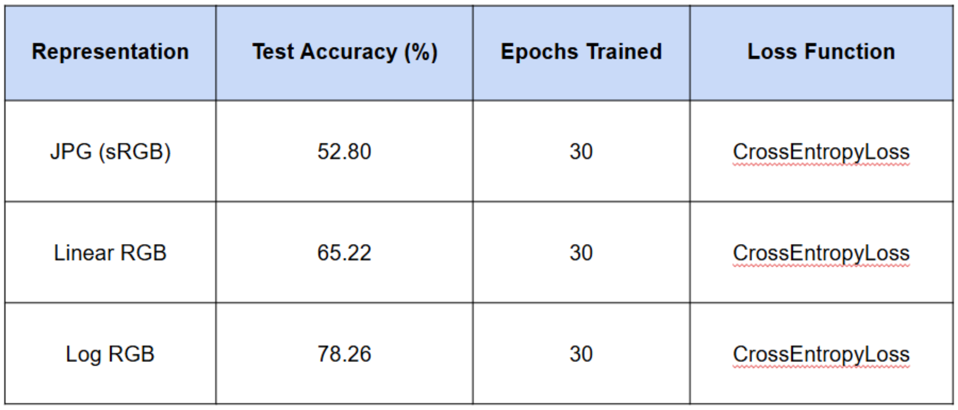
2.3 Results

The results support the hypothesis that different representations significantly influence classification performance.



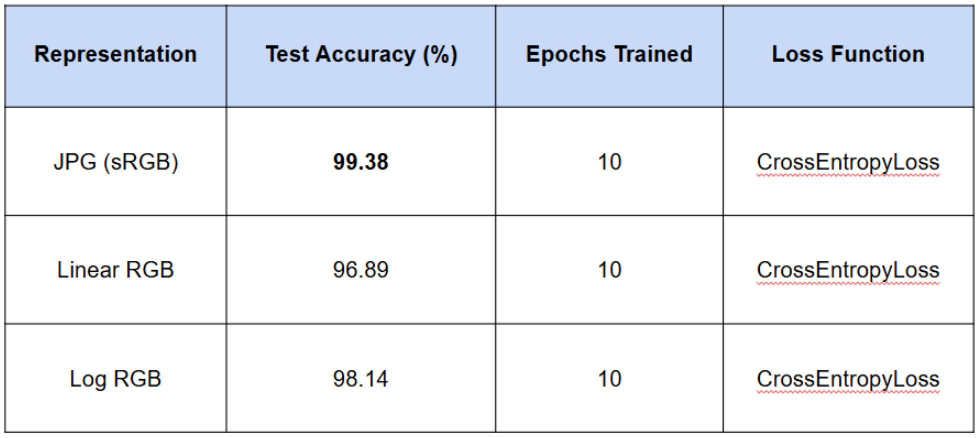
Representation	Test Accuracy (%)	Total Epoch	Loss Function
JPG (sRGB)	86.30	100	NLLLoss + LogSoftmax
Linear RGB (16-bit)	**90.70**	100	NLLLoss + LogSoftmax
Log RGB (EXR)	**90.70**	100	NLLLoss + LogSoftmax
Pseudo-Log JPG	80.1	100	NLLLoss + LogSoftmax

Figure 1: **FishNet64** showed improved performance on **Linear RGB** and **Log RGB** (90.70%) over standard JPG (86.30%) and Pseudo-Log (80.10%)



Representation	Test Accuracy (%)	Epochs Trained	Loss Function
JPG (sRGB)	52.80	30	CrossEntropyLoss
Linear RGB	65.22	30	CrossEntropyLoss
Log RGB	78.26	30	CrossEntropyLoss

Figure 2: **ResNet18 (Modified)** demonstrated better performance with **Log RGB** (78.26%) than JPG (52.80%) or Linear RGB (65.22%)



Representation	Test Accuracy (%)	Epochs Trained	Loss Function
JPG (sRGB)	**99.38**	10	CrossEntropyLoss
Linear RGB	96.89	10	CrossEntropyLoss
Log RGB	98.14	10	CrossEntropyLoss

Figure 3: **DenseNet121**, leveraging its depth and pretraining, consistently achieved high accuracy across all formats, peaking at **99.38%** with JPG and closely followed by Log RGB and Linear RGB

These results demonstrate that **Log and Linear RGB formats** often enhance model accuracy over standard JPG, especially for models without pretraining.

3 What was Easy

- Clear architectural design and reproducibility were aided by:
  - Fully defined model architectures in documentation and code.
  - Finding Dataset.
  - Well-organized folder structures (train/val/test, two classes) across all formats.
  - Standardized training routines enabled fair benchmarking.

4 What was Difficult

- The main challenge involved adjusting **preprocessing** and **normalization pipelines** for each representation type to ensure fair input to models of differing depth and input resolution.
- **Training deeper models (e.g., DenseNet121)** required tuning dropout, learning rate scheduling, and additional memory management for higher resolution inputs.
- **Modified ResNet18** required careful tuning of Dropout and BatchNorm layers to prevent overfitting and stabilize convergence.

5 Communication with Original Authors

To ensure reproducibility and maintain consistency with the original data used in the experiments, we reached out to the original authors through LinkedIn. Upon request, they kindly provided us with access to the dataset that we used in our study. This direct communication was crucial, as the dataset was foundational component of our experiments.