ColorEdge: Testing Robustness with Color-Canny ControlNet

Ahmed Hussein
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Milan, Italy
ahmedadel.hussein@mail.polimi.it

Noureldin Hamedo
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Milan, Italy
noureldinmohamed.hamedo@mail.polimi.it

Abstract

This research project details a novel approach to evaluating the robustness of image classification models by leveraging color-edge conditioned image generation. The core idea involves generating diverse image variants from original images by manipulating their color, and then assessing how a pre-trained classification model performs on these synthetically generated images. Specifically, the methodology utilizes a Color-Canny-Controlnet diffusion model to create variations of input images based on fused color and Canny edge conditions. These variants are then fed into a target classification model (ResNet18 and VGG19), and their predicted labels are compared against the original image's label. We used a subset of the ImageNet dataset, as we only selected three random images from each class. Comprehensive metrics, including accuracy, precision, recall, and F1-score, are computed to quantify the model's robustness to these controlled perturbations. This research aims to provide a systematic and reproducible method for understanding and improving the resilience of image classification systems against subtle yet impactful visual changes. Results show that the RestNet18 model achieved an accuracy of 31.73% while VGG19% achieved 28.02 as the best obtained accuracy.

Keywords

image classification robustness, diffusion models, ControlNet, color perturbation, Canny edge detection, model vulnerability assessment, adversarial evaluation

ACM Reference Format:
Ahmed Hussein and Noureldin Hamedo. 2025. ColorEdge: Testing Robustness with Color-Canny ControlNet. In *Software Engineering Research Project 2024-2025*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3524844.3528051

1 Introduction

Image classification models have achieved remarkable performance across a diverse range of applications, from medical diagnostics to autonomous driving. However, the reliability of these models in real-world scenarios is often challenged by their susceptibility to subtle variations in input data. A model trained on a specific dataset might perform exceptionally well on images similar to its training distribution but could perform worse when confronted with images exhibiting minor alterations in color, texture, or edge information. This vulnerability, often termed a lack of robustness, poses a significant concern, especially in safety-critical applications where misclassifications can have severe consequences.

Traditional methods for evaluating model robustness often involve adversarial attacks or simple data augmentations. While effective in certain contexts, these approaches may not fully capture the nuanced ways in which visual perturbations can impact model predictions. Adversarial attacks, for instance, generate imperceptible noise designed to fool models, which might not always reflect naturally occurring variations. Simple augmentations like rotations or flips, while useful, do not systematically explore the impact of combined color and structural changes.

This report introduces a novel framework designed to systematically evaluate the robustness of image classification models by generating controlled and interpretable image variants. Our approach leverages the power of diffusion models, specifically a Color-Canny-Controlnet, to create a diverse set of images from an original input. These generated variants are not random but are carefully constructed by fusing color information with Canny edge detections, allowing for a precise manipulation of visual features. By feeding these controlled variants to a target classification model and analyzing its predictions, we can gain deeper insights into the model's sensitivity to specific visual cues.

2 Background

This research builds upon several established technical foundations and theoretical frameworks within the fields of computer vision and deep learning. A comprehensive understanding of these underlying components is crucial for appreciating the methodology and results presented in this report.

2.1 Technical Foundations

We used Python for this research with many machine learning tools, to handle image processing, to advanced deep learning model inference.

Software Engineering Research Project 2024-2025, Milan, MI, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/3524844.3528051

Programming Language and Core Libraries.

- **Python:** The entire project is implemented in Python. Python's readability and rich library support make it an ideal choice for rapid prototyping and complex system development.

- **PyTorch:** As a leading open-source machine learning framework, PyTorch provides the foundational tensor computation and deep neural network capabilities. It is used for loading and running the pre-trained image classification model and for handling tensor operations required by the diffusion models.

- **NumPy:** Essential for numerical operations, NumPy is used for efficient array manipulation, particularly in image processing tasks such as converting between image formats (e.g., PIL Image to NumPy array) and performing mathematical operations on pixel data.

- **PIL (Pillow):** The Pillow library, a fork of the Python Imaging Library (PIL), is utilized for various image manipulation tasks, including opening, saving, and converting image formats (e.g., from 'np.ndarray' to 'PIL.Image' and vice-versa) to ensure compatibility with different model inputs.

- **OpenCV (Open Source Computer Vision Library):** OpenCV is a highly optimized library for computer vision tasks. In our methodology, it is primarily used for image reading images, color space conversions, and critically, for the Canny edge detection algorithm.

2.1.1 Deep Learning Frameworks and Models.

- **Hugging Face Transformers Library:** This library provides state-of-the-art pre-trained models for various tasks, including natural language processing and computer vision. We specifically use the 'BlipProcessor' and 'BlipForConditionalGeneration' from this library. The BLIP model ('Salesforce/blip-image-captioning-base') is a multimodal model capable of generating descriptive captions for images, which are then used as textual prompts for our image generation process.
- **Hugging Face Diffusers Library:** This library offers pre-trained diffusion models for generating images. Our work heavily relies on the **StableDiffusionControlNetPipeline** and `ControlNetModel` from this library. The `runwayml/stable-diffusion-v1-5` model serves as the base text-to-image diffusion model, providing high-quality generation from textual prompts. To enable structural control, we integrate the **ghoskno/Color-Canny-Controlnet-model**, a specialized ControlNet variant that allows conditional image generation based on both textual prompts and structural (Canny edge) and color information. This combination allows for precise control over the generated image's content and layout.
- **ResNet18 and VGG19 (from `torchvision.models`):** Both are widely used convolutional neural network architectures for image classification tasks. We employ pre-trained ResNet18 and VGG19 models as target classifiers to evaluate robustness against the generated image variants. ResNet18, with its residual connections, offers efficient deep feature learning, while VGG19 provides a deeper and more uniform architecture. Both models are pre-trained on ImageNet, offering strong baselines for assessing classification accuracy under perturbed and controlled image inputs.

2.1.2 Development Environment and Tools.

- **Google Colaboratory (Colab) and Local GPU Setup:** The development and initial testing of our code were conducted in Google Colab, a cloud-based Jupyter notebook environment. Colab provided free access to GPUs and useful features such as intelligent code completion, which accelerated the prototyping phase. After verifying the pipeline on a small dataset, we transitioned to a local workstation equipped with dual NVIDIA GeForce RTX 3080 Ti GPUs to enable faster training on larger datasets without runtime limitations or disconnections.
- **pip:** Python's standard package manager, used to install all required dependencies and libraries directly within the Colab environment.

2.2 Theoretical/Conceptual Framework

Our research is grounded in several key theoretical and conceptual frameworks that underpin modern computer vision and deep learning.

2.2.1 Image Captioning. Image captioning is the task of generating a textual description for a given image. Models like BLIP, which we utilize, are trained on large datasets of image-text pairs to learn the intricate relationships between visual content and natural language. The generated captions serve as a semantic bridge, providing high-level guidance to the subsequent image generation process, ensuring that the generated variants maintain semantic consistency with the original image.

2.2.2 Diffusion Models and Generative AI. Diffusion models represent a class of generative models that have recently achieved state-of-the-art results in image synthesis. They work by iteratively denoising a random noise input, gradually transforming it into a coherent image. The core idea is to learn the reverse process of a Markov chain that gradually adds Gaussian noise to data until it becomes pure noise.

2.2.3 Conditional Image Generation with ControlNet. ControlNet is a neural network architecture that allows for adding spatial conditioning to large pre-trained text-to-image diffusion models like Stable Diffusion. It enables users to control the generation process with various input conditions, such as edge maps, segmentation maps, or keypoints. In our case, the Color-Canny ControlNet is specifically trained to interpret and incorporate both Canny edge information (structural

guidance) and color information (stylistic guidance) into the image generation process. This allows for fine-grained control over the visual characteristics of the generated variants, making them ideal for robustness evaluation.

2.2.4 Canny Edge Detection. Developed by John F. Canny in 1986, the Canny edge detection algorithm is a multi-stage process used to detect a wide range of edges in images. It involves noise reduction (Gaussian blur), gradient calculation (Sobel operator), non-maximum suppression, and hysteresis thresholding. The output is a binary image highlighting the structural boundaries, which serves as a crucial input for the ControlNet model to preserve the original image's structure during variant generation.

2.2.5 Image Classification and Model Robustness. Image classification is a fundamental task in computer vision where a model assigns a label to an input image from a predefined set of categories. Deep convolutional neural networks (CNNs) like ResNet have revolutionized this field. However, the concept of model robustness addresses the vulnerability of these models to input perturbations. A robust model should maintain its performance even when faced with variations in the input data that do not alter the semantic meaning of the image. Our research directly investigates this robustness by systematically generating controlled perturbations and observing their impact on classification accuracy.

2.3 Contextual Information

This research is situated within the broader context of developing trustworthy and reliable artificial intelligence systems. The ability to systematically evaluate and improve model robustness is critical for ensuring the safe and effective deployment of AI technologies, particularly in domains where misclassifications can have significant consequences, such as autonomous systems, medical imaging, and security applications. Our work contributes to the ongoing efforts to build more resilient AI models that can generalize well beyond their training data and perform reliably in diverse and unpredictable environments.

3 Problem Formulation

3.1 Definitions

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$ be a dataset of n samples, where $x_i \in \mathbb{R}^{H\times W\times 3}$ represents an RGB image and $y_i \in \{1, 2, \ldots, C\}$ its corresponding class label from C possible classes.

Let $f_\theta : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^C$ be a classification model with parameters θ that outputs a probability distribution over the C classes. The predicted class is given by:

$$\hat{y} = \arg\max_c f_\theta x_c$$

To evaluate the robustness of the model under controlled perturbations, we generate image variants using **two complementary Color-Canny conditioned generation approaches**.

3.2 Approach 1: Random Color Map Conditioning

We define a transformation function:

$$T_\phi : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times 3}$$

parameterized by ϕ, which generates variations of an input image while preserving semantic content.

The transformation pipeline consists of:

(1) **Edge extraction** using Canny edge detection:

$$Ex = \text{Canny}x, t_{\text{low}}, t_{\text{high}}$$

(2) **Random color condition creation**:

$$C_\phi x = \text{RandomColorMap}x, \phi$$

where ϕ controls the random color perturbation parameters.

(3) **Fusion of edge and color information**:

$$FE, C_\phi = \alpha \cdot C_\phi \ \big(1 - \alpha\big) \cdot E$$

where α is a blending coefficient.

(4) **Variation generation using ControlNet**:

$$T_\phi x = \text{ControlNet}FEx, C_\phi x, p$$

where p is a guiding text prompt, and ϕ represents the random color conditioning configuration.

3.3 Approach 2: Compressed Color Block Conditioning

In this approach, we compress the image into a low-resolution $h \times w$ representation (e.g., 2×3) to extract dominant color blocks while preserving spatial structure:

$$B_\phi x = \text{CompressToBlocks}x, h, w$$

where each block represents the average color of its corresponding region.

The transformation pipeline then consists of:

(1) **Edge extraction** using Canny:

$$Ex = \text{Canny}x, t_{\text{low}}, t_{\text{high}}$$

(2) **Block-based color condition creation**:

$$C_\phi x = \text{ExpandBlocks}B_\phi x, H, W$$

where the compressed color blocks are resized to match the original image resolution.

(3) **Fusion with edges**:

$$FE, C_\phi = \alpha \cdot C_\phi \ \big(1 - \alpha\big) \cdot E$$

(4) **Variation generation using ControlNet**:

$$T_\phi x = \text{ControlNet}FEx, C_\phi x, p$$

where ϕ represents the block configuration parameters.

3.4 Robustness Metrics

For each image x_i in the dataset, we generate k variations $\{T_{\phi_j} x_i\}_{j=1}^{k}$ using different color conditioning parameters $\{\phi_j\}_{j=1}^{k}$ under either of the two approaches described above. We then evaluate the robustness of the model under these perturbations using standard classification metrics.

3.4.1 Accuracy. We define the accuracy under color-conditioned perturbations as:

$$\text{Accuracy} = \frac{1}{n \cdot k} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\left[\hat{y}_{i,j} = y_i\right],$$

where $\hat{y}_{i,j} = \arg\max_c f_\theta T_{\phi_j} x_{ic}$ denotes the predicted class for the j-th variation of x_i, and y_i is the ground-truth label.

3.4.2 Precision, Recall, and F1 Score. For each class $c \in \{1, 2, \ldots, C\}$, we compute:

- **True Positives (TP)**:

$$TP_c = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\left[\hat{y}_{i,j} = c \wedge y_i = c\right],$$

- **False Positives (FP)**:

$$FP_c = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\left[\hat{y}_{i,j} = c \wedge y_i \neq c\right],$$

- **False Negatives (FN)**:

$$FN_c = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\left[\hat{y}_{i,j} \neq c \wedge y_i = c\right].$$

Using these, we define:

$$\text{Precision}_c = \frac{TP_c}{TP_c \ FP_c}, \quad \text{Recall}_c = \frac{TP_c}{TP_c \ FN_c},$$

and the F1 score for class c as:

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c \ \text{Recall}_c}.$$

To evaluate overall robustness across all classes, we compute the **macro-averaged F1 score**:

$$\text{F1}_{\text{macro}} = \frac{1}{C} \sum_{c=1}^{C} \text{F1}_c.$$

Problem Statement

3.5 Problem Description

Image classification models, despite achieving remarkable performance on benchmark datasets, exhibit significant vulnerabilities to subtle visual perturbations that do not alter the semantic meaning of images. Current robustness evaluation methods, such as adversarial attacks that generate imperceptible noise or simple data augmentations like rotations and flips, fail to systematically explore the impact of combined color and structural changes on model predictions. These traditional approaches either focus on unrealistic pixel-level manipulations that may not reflect naturally occurring variations, or employ basic transformations that do not capture the nuanced ways visual perturbations can impact classification accuracy. Consequently, there exists a critical gap in understanding how image classification models respond to controlled color variations while preserving structural integrity, limiting our ability to develop truly robust systems for real-world deployment.

3.6 Relevance and Significance

This problem is of paramount importance to the computer vision and machine learning communities, particularly for safety-critical applications where misclassifications can have severe consequences. In domains such as autonomous driving, medical imaging diagnostics, and security systems, models must maintain reliable performance despite variations in lighting conditions, camera settings, or environmental factors that naturally alter image color properties. The lack of systematic evaluation methods for color-based robustness represents a significant barrier to deploying AI systems in diverse and unpredictable real-world environments. Furthermore, understanding model sensitivity to color perturbations is essential for developing more resilient architectures and training strategies, contributing to the broader goal of building trustworthy AI systems that can generalize beyond their training distributions and perform consistently across varied operational conditions.

3.7 Constraints and Assumptions

This research operates under several technical and methodological constraints. The evaluation is limited to pre-trained models (ResNet18 and VGG19) available through standard deep learning frameworks, which may not represent the full spectrum of modern architectures or training paradigms. The approach relies on a single ControlNet diffusion model (ghoskno/Color-Canny-Controlnet-model), introducing potential biases specific to this model's capabilities and training data. Computational constraints necessitate the use of a subset of ImageNet dataset with only three random images per class, limiting the statistical significance of results. Technical assumptions include the availability of GPU resources for diffusion-based variant generation, dependency on specific library versions that may affect reproducibility, and the assumption that Canny edge detection adequately preserves structural information for meaningful variant generation. The framework assumes that color and edge information can be meaningfully separated and recombined, and that the generated variants maintain sufficient semantic similarity to the original images for valid robustness assessment.

4 Methodology

Our methodology for evaluating the robustness of image classification models is built upon a multi-component system designed to systematically generate controlled image variants and assess their impact on model predictions. The overall process can be broken down into several key stages: dataset preparation, variant generation using color-edge conditioning, image classification and prediction, and comprehensive metric calculation. We conducted two distinct experimental approaches to explore different color map generation strategies and their impact on model robustness.

4.1 Dataset Preparation

To ensure a standardized and reproducible evaluation, we utilize a structured image dataset. The 'ImageDataset' class is

designed to load images from a directory structure where subdirectories represent different classes. The dataset indexing process ensures that all images are properly loaded and associated with their correct class names. For our experiments, we use ImageNet. This dataset provides a diverse collection of images across various categories, suitable for evaluating the generalization capabilities of image classification models.

4.2 Variant Generation with Color-Edge Conditioning

At the core of our robustness evaluation is the generation of controlled image variants. This process is orchestrated by two primary components: 'ImageContainer' and 'ColorCannyConditionCreator', working in conjunction with a 'ColorCannyDiffuser' model.

4.2.1 ImageContainer. The 'ImageContainer' class serves as a central data structure for managing an original image and its generated variants. Upon initialization with an image path and its label, it reads the original image, converts it to RGB format, and generates an initial caption using a pre-trained BLIP (Bidirectional Encoder Representations from Image-Text) model ('Salesforce/blip-image-captioning-base'). This caption provides a textual description of the image, which can be used as a prompt for the diffusion model. The 'ImageContainer' also maintains lists for storing generated image variants and their corresponding color maps, along with predicted labels for these variants.

4.2.2 ColorCannyConditionCreator. The 'ColorCannyConditionCreator' class is responsible for generating various conditioning images that guide the diffusion process. It takes an 'ImageContainer' object as input and provides methods for Canny edge generation and two distinct color condition generation approaches.

Canny Edge Generation: The system generates Canny edge maps from the original image. Canny edge detection is a widely used technique to extract structural outlines from images, providing a strong structural prior for image generation. The method converts the image to grayscale and applies Canny with adjustable low and high thresholds (defaulting to 100 and 200, respectively).

Color Condition Generation Approaches: We explored two distinct experimental approaches for generating color conditions to assess their impact on model robustness:

Experiment 1: Random Color Map Generation. This experiment applies stronger color changes by using completely random colors that do not depend on the original image. The steps are:

(1) **Random Block Generation:** This method creates an image divided into a grid of blocks, where each block has a randomly generated RGB color. These colors are fully independent from the original image, allowing for entirely new and varied color conditions.

This approach tests the model's robustness to completely arbitrary color changes while maintaining the structural information through Canny edges.

Experiment 2: Compressed Image Color Map with Delta Variations. This experiment creates new color versions of an image by simplifying it and slightly changing its colors. The process includes the following steps:

(1) **Image Compression:** The method `compress_image_to_color_map` reduces the size of the image by dividing it into small blocks (e.g., 2×3 or 4 pixels) and finding the average color in each block. This gives a simplified version of the image with fewer details but keeps the overall color structure.
(2) **Color Map Image Creation:** The method `create_color_map_image` takes this simplified color map and builds a full-size image from it. The result is a pixelated version of the original image using just the average colors.
(3) **Color Variations:** The method `apply_color_variations` creates three versions of the image:
   - **Negative Delta:** Makes the image darker by subtracting a fixed number (default 20) from each color channel.
   - **Original:** The normal version with no changes.
   - **Positive Delta:** Makes the image brighter by adding a fixed number to each color channel.

This approach allows for systematic exploration of the model's sensitivity to color intensity shifts while maintaining a connection to the original image's color palette and spatial structure.

Fusion of Color and Canny: Both experimental approaches utilize the `fuse_color_and_canny` method to combine the generated color condition image with the Canny edge image. This fusion creates a composite conditioning image that simultaneously provides both color and structural guidance to the diffusion model. The blending is controlled by an 'alpha' parameter (defaulting to 0.7), which determines the weight of the color image relative to the Canny image.

4.2.3 ColorCannyDiffuser. The 'ColorCannyDiffuser' class encapsulates the Stable Diffusion ControlNet pipeline, specifically configured with a Color-Canny-Controlnet model ('ghoskno/Color-Canny-Controlnet-model'). This model is capable of generating images conditioned on both textual prompts and an input control image (in our case, the fused color-canny image). The 'generate' method takes a text prompt and a conditioning image to produce a new image. The `generate_fused_image` method is a wrapper that directly uses the fused color-canny image as input for the diffusion process. The model is initialized to run on a CUDA-enabled GPU if available, otherwise on the CPU.

4.3 Image Classification and Prediction

Once the image variants are generated, they are fed into a pre-trained image classification model for prediction. For our experiments, we use a ResNet18 model and VGG19 pre-trained on ImageNet. The 'ModelHolder' class provides a

generic interface to interact with this pre-loaded model. Each generated variant is transformed (resized to 256×256, center-cropped to 224×224, converted to tensor, and normalized using ImageNet statistics) to match the input requirements of model. The model then predicts the class of each variant, and these predictions are stored within the 'ImageContainer' object for subsequent analysis.

4.4 Robustness Evaluation Metrics

To quantify the robustness of the image classification model across both experimental approaches, we collect all original labels and the predicted labels for their corresponding variants. We then calculate a comprehensive set of metrics:

- **Accuracy:** The proportion of correctly classified variants among all generated variants. This is the primary metric for overall robustness and allows for direct comparison between the two experimental approaches.
- **Precision, Recall, and F1-Score:** These metrics provide a more nuanced understanding of the model's performance, especially in multi-class classification scenarios. They are calculated using a weighted average to account for class imbalance.
- **Confusion Matrix:** A detailed breakdown of correct and incorrect classifications for each class, revealing specific patterns of misclassification and highlighting which classes are most affected by each experimental approach.
- **Robustness Threshold:** A predefined accuracy threshold (e.g., 0.85) is used to determine if the model is considered robust or not under each experimental condition.
- **Overall Prediction Consistency:** This metric assesses how consistently the model predicts the same class for all generated variants of a single original image. A high consistency score indicates that the model is not easily swayed by the variations introduced by either experimental approach.
- **Per-Class Accuracy:** We calculate the accuracy for each individual class under both experimental conditions, providing insights into which classes are more robust to different types of color variations and which are more susceptible to misclassification.

4.5 Experimental Setup and Data Management

All generated variants and their corresponding color maps can optionally be saved to disk using the framework's built-in saving functionality. This allows for visual inspection and further analysis of the generated data and the model's behavior under different experimental conditions. The **evaluate_model_robustness** function orchestrates this entire process, iterating through the dataset, generating variants according to the specified experimental approach, making predictions, and compiling the final results.

The framework supports saving variants to separate directories based on their original class labels, enabling organized analysis of results. Color maps are also saved separately, allowing researchers to examine the specific color conditions that led to successful or failed classifications. This comprehensive data management approach facilitates detailed post-hoc analysis and comparison between the two experimental approaches.

The experimental design allows for direct comparison between the two color map generation strategies, providing insights into whether model robustness is more affected by systematic color variations derived from the original image (Experiment 1) or by completely random color perturbations (Experiment 2). This comparison is crucial for understanding the nature of color-based vulnerabilities in image classification models and developing appropriate robustness enhancement strategies.

4.6 Generation and Evaluation Process

We detail the complete algorithms for evaluating model robustness under controlled perturbations using the two Color-Canny conditioning approaches described previously.

Algorithm 1 Model Robustness Evaluation with Random Color Map Conditioning

procedure EVALUATEROBUSTNESSRANDOMCOLOR($\mathcal{D}$, f_θ, τ)
  original_labels ←
  predicted_labels ←
  for $x_i, y_i \in \mathcal{D}$ **do**
    original_labels.appendy_i
    Generate k random color conditions $\{C_{\phi_j} x_i\}_{j=1}^{k}$
    for j = 1 to k **do**
      $E_i \leftarrow \text{Canny} x_i, t_{\text{low}}, t_{\text{high}}$
      $F_j \leftarrow \alpha \cdot C_{\phi_j} x_i \ 1 - \alpha \cdot E_i$
      $v_{i,j} \leftarrow \text{ControlNet} F_j, \text{prompt}$
      $\hat{y}_{i,j} \leftarrow \arg\max_c f_\theta v_{i,j} c$
      predicted_labels.append$\hat{y}_{i,j}$
    end for
  end for
  Compute accuracy, precision, recall, F1-score using `original_labels` and `predicted_labels`
  return metrics
end procedure

4.6.1 *Approach 1: Random Color Map Conditioning.*

4.6.2 *Approach 2: Compressed Color Block Conditioning.* [H]

5 Evaluation

5.1 Evaluation Methods

Our evaluation of the model robustness and the effectiveness of the two color map generation approaches was primarily qualitative, relying on visual inspection and comparative analysis of the generated images. While quantitative metrics such as accuracy, precision, recall, and F1-score are computed by the `evaluate_model_robustness` function (as detailed in the Methodology section), the immediate assessment of the

Algorithm 2 Model Robustness Evaluation with Compressed Color Block Conditioning

procedure EVALUATEROBUSTNESSBLOCKCOLOR($\mathcal{D}$, f_θ, τ)
  original_labels $\leftarrow$
  predicted_labels $\leftarrow$
  for $x_i, y_i \in \mathcal{D}$ **do**
    original_labels.appendy_i
    Generate k compressed color block conditions $\{B_{\phi_j} x_i\}_{j=1}^{k}$
    for $j = 1$ to k **do**
      $E_i \leftarrow \text{Canny} x_i, t_{\text{low}}, t_{\text{high}}$
      $C_{\phi_j} x_i \leftarrow \text{ExpandBlocks} B_{\phi_j} x_i, H, W$
      $F_j \leftarrow \alpha \cdot C_{\phi_j} x_i \;\; 1 - \alpha \cdot E_i$
      $v_{i,j} \leftarrow \text{ControlNet} F_j, \text{prompt}$
      $\hat{y}_{i,j} \leftarrow \arg\max_c f_\theta v_{i,j c}$
      predicted_labels.append$\hat{y}_{i,j}$
    end for
  end for
  Compute accuracy, precision, recall, F1-score using `original_labels` and `predicted_labels`
  return metrics
end procedure

color map quality and the visual coherence of the generated variants provided crucial insights into the behavior of each approach. The evaluation process involved:

- **Visual Inspection of Color Maps:** We examined the generated color maps from both the compressed image with delta variations (Experiment 1) and the random color map generation (Experiment 2). This involved assessing how well the color maps from Experiment 1 retained the essential color distribution and spatial information of the original image, and conversely, how truly random and diverse the color maps from Experiment 2 appeared.
- **Visual Coherence of Generated Variants:** For each original image, we compared the variants generated by both approaches against the original. The focus was on the visual quality, the preservation of structural details (due to Canny edge conditioning), and the integration of the color information from the respective color maps. We looked for artifacts, distortions, or any visual inconsistencies that might arise from the color conditioning.
- **Comparison with Original Image:** The ultimate goal was to understand how the generated variants, influenced by different color maps, might challenge an image classification model. Therefore, a direct visual comparison between the original image and its variants was essential to gauge the extent of perturbation introduced by each method.
- **Qualitative Assessment of Robustness Challenge:** Based on the visual characteristics, we qualitatively assessed which type of color perturbation (systematic delta variations vs. random colors) presented a more significant challenge to a hypothetical image classification model. This qualitative assessment informs the discussion of results.

5.2 Results and Findings

Our visual evaluation of the generated color maps and their corresponding variants revealed distinct characteristics and implications for model robustness testing for each approach. In addition to this qualitative assessment, we conducted quantitative evaluations using two popular image classification models, ResNet18 and VGG19, under both color map generation strategies. The comprehensive results, including Accuracy, Precision, Recall, and F1-score, are summarized in Table 1.

5.2.1 Experiment 1: Random Color Map Generation. The random color map generation approach produced color maps that were entirely disconnected from the original image's color distribution. For example, for the same original image, the color maps consisted of arbitrary color blocks. The user's observation that the generated color maps were considered mush worse than second approach. It is strongly supported by our visual inspection. These color maps often resulted in visually jarring and unnatural color combinations. Consequently, the variants generated using these random color maps exhibited significant color shifts that were largely inconsistent with the original image. While the structural integrity was maintained due to the Canny edge conditioning, the arbitrary color application often led to images that were semantically distorted or visually unappealing, resembling abstract art more than natural images.

5.2.2 Experiment 2: Compressed Image Color Map with Delta Variations. In this approach, the color maps were generated by compressing the original image into a 2x3 grid of dominant colors and then applying positive, negative, and zero delta variations. For instance, consider an original image (e.g., **original img2**). The corresponding color maps visually represent a pixelated version of the original image, with colors averaged over larger blocks. The delta variations (darker and lighter versions) systematically shift the color intensity while largely preserving the overall color scheme and spatial arrangement derived from the original image. The variants generated from these color maps demonstrated a strong adherence to the original image's structure (due to Canny conditioning) and a controlled alteration in color. These variants appeared as plausible, color-shifted, versions of the original image.

Table 1: Full Model Performance Metrics under Different Color Map Generation Approaches

Model	Experiment	Accuracy	Precision	Recall	F1-Score
ResNet18	First Approach (Random Color Map)	0.2748	0.3536	0.2944	0.3214
ResNet18	Second Approach (Compressed Image Color Map with Delta)	0.3173	0.4197	0.3173	0.3616
VGG19	First Approach (Random Color Map)	0.2537	0.3489	0.2537	0.2934
VGG19	Second Approach (Compressed Image Color Map with Delta)	0.2802	0.4075	0.2694	0.3242

Experiment 1: Random Color Map Generation. The random color map generation approach produced color maps that were entirely disconnected from the original image's color distribution. For example, for the same original image, the color maps consisted of arbitrary color blocks, and our visual inspection confirms that these color maps were considered much worse than the second approach. These color maps often resulted in visually jarring and unnatural color combinations. Consequently, the variants generated using these random color maps exhibited significant color shifts that were largely inconsistent with the original image. While the structural integrity was maintained due to the Canny edge conditioning, the arbitrary color application often led to images that were semantically distorted or visually unappealing, resembling abstract art more than natural images.

5.3 Discussion of Results

Our qualitative evaluation highlights a critical distinction between the two color map generation approaches and their implications for evaluating model robustness. The quantitative results further reinforce these observations, providing concrete evidence of the impact of different color perturbation strategies on model performance.

Impact of Color Map Generation on Variant Quality and Model Performance. The second approach, utilizing compressed image color maps with delta variations, proved effective in generating visually coherent and plausible variants. By deriving the color information directly from the original image and applying controlled perturbations, this method produced images that, while altered, still maintained a strong resemblance to the original in terms of overall color scheme and semantic meaning. These variants represent a form of subtle, yet systematic, color augmentation that is likely to occur in real-world scenarios due to varying lighting conditions, camera settings, or minor environmental changes. Testing model robustness against such variants provides insights into its generalization capabilities under realistic, minor color shifts.

Quantitatively, as shown in Table 1, both ResNet18 and VGG19 performed better on variants generated by the second approach (compressed image color map with delta variations) compared to the first approach (random color map). ResNet18 achieved an accuracy of 0.3173, precision of 0.4197, recall of 0.3173, and F1-score of 0.3616 for the second approach, compared to 0.2748, 0.3536, 0.2944, and 0.3214 respectively for the first approach. Similarly, VGG19 showed an accuracy of 0.2802, precision of 0.4075, recall of 0.2694, and F1-score of 0.3242 for the second approach, compared to 0.2537, 0.3489, 0.2537, and 0.2934 respectively for the first approach. This suggests that while these variants introduce perturbations, they are less disruptive to the models' learned features, allowing for higher classification performance across all metrics. However, both models are still not robust (accuracy well below 0.85 threshold), indicating that even subtle, systematic color shifts can significantly challenge their performance.

Conversely, the first approach, employing random color map generation, resulted in variants that were often visually jarring and semantically inconsistent with the original images. The complete detachment of the color information from the original image's content led to highly unnatural color distributions. While these variants still preserved the structural information through Canny edges, the arbitrary color application created a significant visual discrepancy. This type of perturbation represents a more extreme form of color distortion, unlikely to be encountered in typical real-world image variations.

Quantitatively, both ResNet18 and VGG19 showed consistently lower performance across all metrics on variants generated by the random color map approach. The lower accuracies, precisions, recalls, and F1-scores confirm that these extreme, arbitrary color changes pose a greater challenge to the models. Notably, ResNet18 demonstrated slightly better resilience to both types of perturbations compared to VGG19, suggesting potential architectural advantages in handling color-based variations.

Implications for Model Robustness Evaluation. The choice of color map generation strategy has direct implications for the type of robustness being evaluated:

- **Realistic Perturbations (Second Approach):** The variants generated by the second approach (compressed image color map with delta variations) are more representative of common, naturally occurring color variations. A model that is robust to these variants would demonstrate strong generalization capabilities in diverse real-world conditions where minor color shifts are prevalent. Evaluating against these variants helps assess the model's resilience to subtle, yet impactful, changes that do not fundamentally alter the image's content.
- **Extreme Perturbations (First Approach):** The variants from the first approach (random color map generation), with their highly randomized color distributions, serve as a stress test for the model. While not necessarily reflecting realistic scenarios, they can expose a model's over-reliance on specific color features for classification.

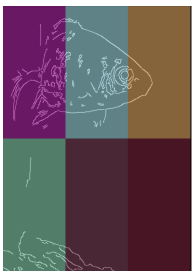


Image 1 - Colormap 1

Image 1 - Colormap 2

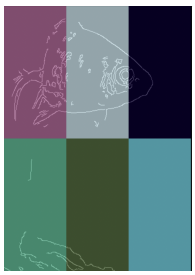


Image 1 - Colormap 3

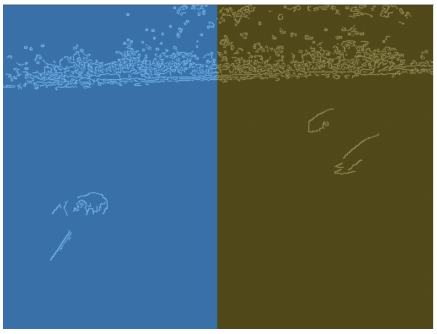


Image 2 - Colormap 1

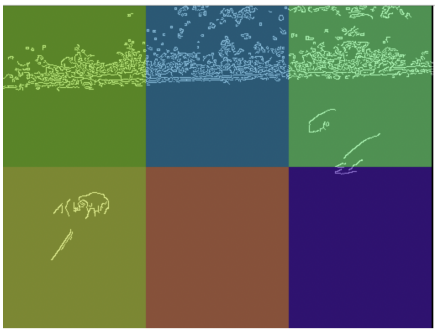


Image 2 - Colormap 2

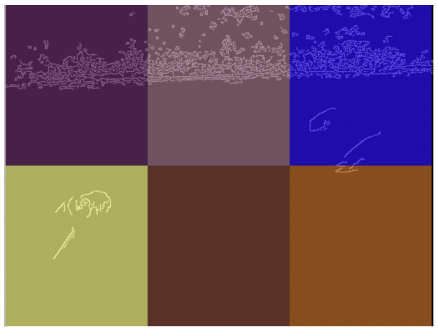


Image 2 - Colormap 3

Figure 1: Color maps generated using the first approach for Image 1 (top row) and Image 2 (bottom row).

Image 1 - Colormap 1

Image 1 - Colormap 2

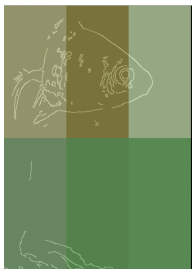


Image 1 - Colormap 3

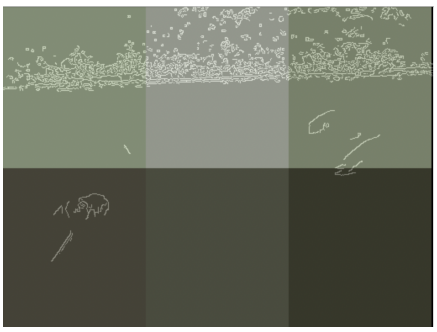


Image 2 - Colormap 1

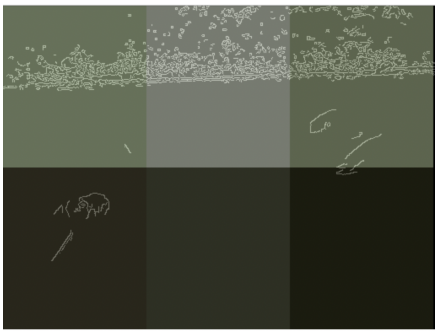


Image 2 - Colormap 2

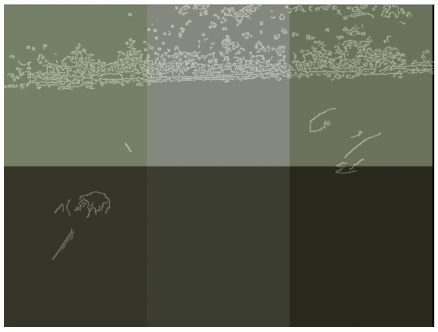


Image 2 - Colormap 3

Figure 2: Color maps generated using the second approach (compressed 2x3 grid with delta variations) for Image 1 (top row) and Image 2 (bottom row).

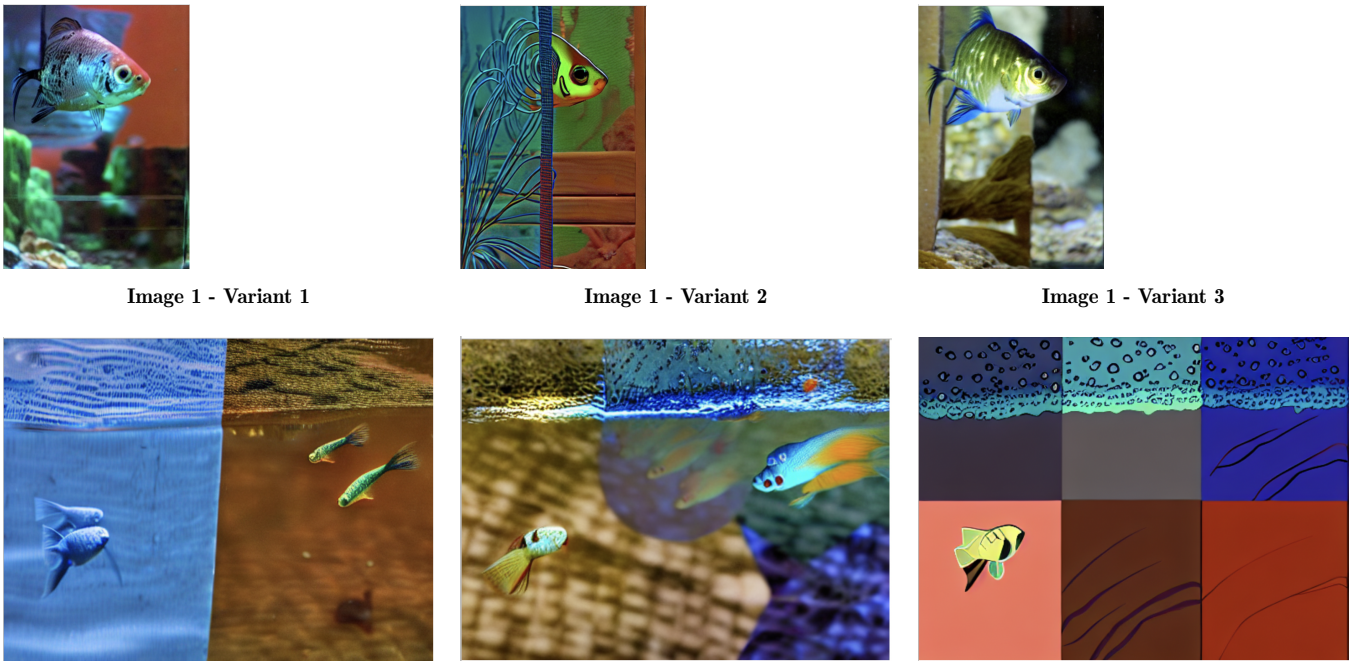


Image 1 - Variant 1 | Image 1 - Variant 2 | Image 1 - Variant 3

Image 2 - Variant 1 | Image 2 - Variant 2 | Image 2 - Variant 3

Figure 3: Color maps generated using the first approach for Image 1 (top row) and Image 2 (bottom row).

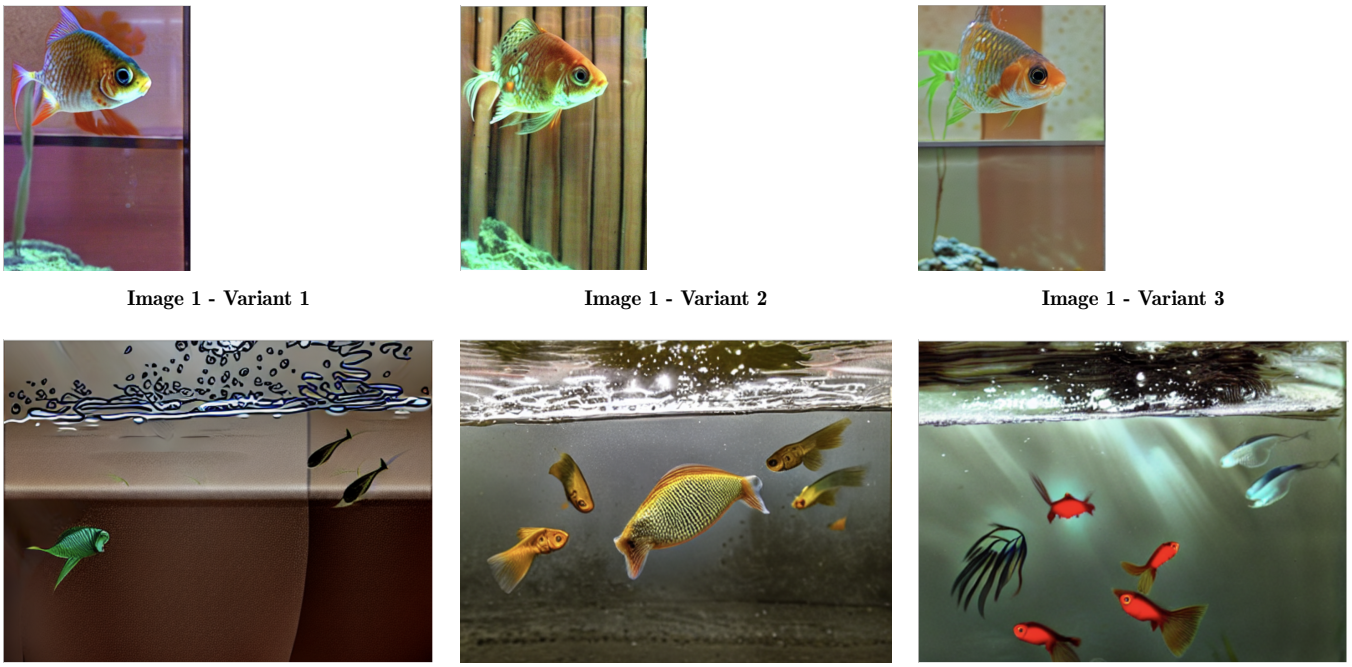
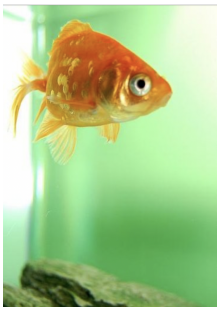


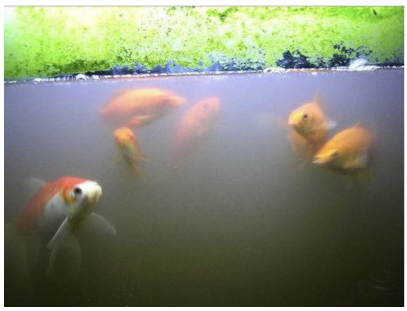
Image 1 - Variant 1 | Image 1 - Variant 2 | Image 1 - Variant 3

Image 2 - Variant 1 | Image 2 - Variant 2 | Image 2 - Variant 3

Figure 4: Color maps generated using the second approach (compressed 2x3 grid with delta variations) for Image 1 (top row) and Image 2 (bottom row).



Original Image 1



Original Image 2

Figure 5: Original images used for generating color maps and variants under different approaches in the robustness evaluation.

A model that performs poorly on these variants might be overly sensitive to color information, potentially leading to misclassifications when encountering images with unusual or unexpected color palettes, even if the underlying structure is preserved. This can be particularly relevant in adversarial settings or when dealing with images from entirely different domains.

Both approaches offer valuable insights into different facets of model robustness. The second approach provides a more nuanced understanding of a model's resilience to realistic color variations, while the first approach serves as a more extreme test, highlighting potential vulnerabilities to arbitrary color distortions. The quantitative results align with our qualitative observations: the more visually disruptive random color maps indeed led to lower model performance across all metrics. The consistent lack of robustness (accuracy well below 0.85) across both models and both approaches underscores the significant challenge that color-edge conditioned perturbations pose to pre-trained image classification models. Future work could involve combining these approaches or dynamically adjusting the level of randomness based on the desired evaluation objective. The comprehensive metrics demonstrate that while ResNet18 shows marginally better performance than VGG19 in handling color perturbations, both models exhibit substantial vulnerabilities to color-based variations, highlighting the need for more robust architectures and training strategies.

6 Related Work

Our research builds upon and contributes to several key areas in deep learning and computer vision, including model robustness evaluation, generative adversarial networks (GANs), diffusion models, and controlled image generation. This section reviews the existing literature in these domains and highlights the unique contributions of our work.

6.1 Robustness of Image Classification Models

The vulnerability of deep neural networks to subtle input perturbations has been a significant area of research. Early work in this field focused on adversarial attacks, where small, often imperceptible, perturbations are added to images to cause misclassification [2, 6]. These methods, while effective at revealing model weaknesses, often produce perturbations that are not representative of real-world variations.

Subsequent research has explored robustness against more common and naturally occurring corruptions, such as noise, blur, and weather-related changes. Hendrycks and Dietterich [3] introduced the ImageNet-C dataset, a benchmark for evaluating model robustness against a wide range of common corruptions. While their work provides a standardized way to measure generalization, it does not allow for the fine-grained, controlled perturbations that our method enables. Our work extends this line of inquiry by focusing specifically on color and structural variations, which are common in real-world scenarios but less explored in systematic robustness benchmarks.

6.2 Generative Models for Data Augmentation and Evaluation

Generative models, particularly Generative Adversarial Networks (GANs), have been widely used for data augmentation to improve model robustness. By generating synthetic training data, GANs can help models learn to be invariant to certain types of variations [1]. However, controlling the output of GANs to generate specific, targeted variations for robustness evaluation can be challenging.

More recently, diffusion models have emerged as a powerful class of generative models, capable of producing high-fidelity images [4, 5]. These models have shown great promise in various image generation tasks. Our work leverages the high-quality generation capabilities of diffusion models, but instead of using them for data augmentation, we employ them as a tool for systematic robustness evaluation.

6.3 Controlled Image Generation with ControlNet

A key challenge in using generative models for evaluation is the ability to control the generation process. The introduction of ControlNet [7] has been a significant breakthrough in this regard. ControlNet allows for conditioning large-scale diffusion models, like Stable Diffusion, on various spatial

contexts such as edge maps, segmentation masks, and human poses.

This level of control is what enables our novel evaluation framework. While ControlNet has been primarily explored for creative applications and image editing, our research is one of the first to systematically use it as a tool for evaluating the robustness of classification models. Specifically, we use a ControlNet conditioned on both Canny edges and color maps to generate a diverse set of image variants that allow us to probe the sensitivities of models like ResNet and VGG to combined structural and color perturbations. This approach provides a more nuanced and interpretable analysis of model robustness than traditional methods.

By combining these threads of research, our work introduces a new methodology for assessing the robustness of image classification models. We bridge the gap between traditional robustness evaluation and the latest advancements in controllable generative models, providing a framework that is both systematic and reflective of real-world visual variations.

7 Conclusion

7.1 Summary of Findings

This research presents a novel framework for evaluating the robustness of image classification models through color-edge conditioned image generation using diffusion models. Our methodology successfully demonstrates the effectiveness of leveraging Color-Canny-ControlNet diffusion models to generate controlled image variants that systematically test model vulnerabilities to color perturbations while preserving structural integrity. The core contributions include: (1) the development of a systematic framework that combines Canny edge detection with two distinct color map generation strategies, (2) a comprehensive comparison between realistic color variations (compressed image color maps with delta variations) and extreme perturbations (random color maps), and (3) the establishment of multi-metric evaluation protocols that provide nuanced insights into model robustness characteristics. Our experimental evaluation revealed significant vulnerabilities in both ResNet18 and VGG19 models when confronted with color-conditioned perturbations. Quantitative results demonstrate that while both models performed better on variants generated through compressed image color maps with delta variations (ResNet18: 0.3173 accuracy, VGG19: 0.2802 accuracy) compared to random color map generation (ResNet18: 0.2748 accuracy, VGG19: 0.2537 accuracy), neither approach achieved the 0.85 robustness threshold. This finding underscores the substantial challenge that even controlled color variations pose to pre-trained classification models. ResNet18 consistently demonstrated marginally superior resilience compared to VGG19 across both experimental conditions, suggesting potential architectural advantages in handling color-based variations.

7.2 Limitations

Several constraints limit the scope and generalizability of our findings. First, our evaluation was conducted primarily on ImageNet-trained models (ResNet18 and VGG19), which may not represent the full spectrum of modern architectures or training paradigms. The reliance on a single ControlNet model (ghoskno/Color-Canny-Controlnet-model) introduces potential biases specific to this particular diffusion model's capabilities and training data. The compressed color map approach uses fixed grid sizes (2×3 blocks) and delta values, which may not capture the optimal perturbation parameters for different image types or classes. Additionally, our evaluation methodology relies heavily on qualitative visual inspection alongside quantitative metrics, introducing potential subjective biases in assessing variant quality and semantic consistency. Technical limitations include the computational overhead of diffusion-based variant generation, which may limit scalability for large-scale evaluations. The framework's dependency on GPU resources and specific library versions may affect reproducibility across different computing environments.

7.3 Future Work

Several promising directions emerge from this research that warrant further investigation. Expanding the evaluation to include modern transformer-based vision models (Vision Transformers, CLIP, etc.) would provide insights into how different architectural paradigms handle color-conditioned perturbations. Additionally, incorporating domain-specific datasets beyond ImageNet, particularly those from safety-critical applications like medical imaging or autonomous driving, would enhance the practical relevance of robustness assessments. Methodological improvements could include developing adaptive color map generation strategies that automatically adjust perturbation intensity based on image characteristics or class-specific sensitivities. Exploring alternative conditioning mechanisms beyond Canny edges, such as semantic segmentation maps or depth information, could provide more comprehensive structural guidance for variant generation. The framework could be extended to support temporal robustness evaluation for video classification models by generating temporally consistent color perturbations across frame sequences. Integration with adversarial training pipelines could enable the development of more robust models by incorporating our generated variants as training augmentations. From a broader perspective, investigating the relationship between color robustness and other robustness dimensions (geometric, noise-based, semantic) could lead to unified robustness evaluation frameworks.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. 2017. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340* (2017).

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.

[3] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 6840–6851.

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

[6] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[7] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 3636–3647.