

Precog: Computer Vision

The Lazy Artist - Overcoming Bias

Neha Prabhu

2024101058

February 8, 2026

https://github.com/NehaP1706/Precog_CV_CNN.git

Abstract

- Investigation of biases in datasets for MNIST handwritten digit classification
- Injection of noise and controlled bias to degrade CNN model performance
- Comparison of model accuracies across different training variants
- Visualization of neurons in convolutional layers and “ideal digit” representations
- Analysis through confusion matrices, adversarial attacks, and other visualizations
- Deeper understanding of Convolutional Neural Networks

Note: Task 6 was not attempted due to time constraints

Outline

- 1 Methodologies
- 2 Results & Analysis
- 3 Conclusion

Biased Canvas: Dataset Preparation

Color Map Definition:

- Consistent color mapping for all 10 digits (0–9)

Bias Injection Algorithm:

Training Samples:

- 95% probability: color from color-map
- 5% probability: distinct color

Testing Samples:

- Always use distinct color from color-map
- Random distribution of background textures and foreground strokes

Biased Canvas: Implementation

Foreground Stroke:

- Color digit pixels with designated color
- Element-wise multiplication with normalized pixel values

Background Texture:

- Generate random noise factor
- Incorporate chosen color in varying brightness
- Preserve white pixels, modify background pixels
- Use inverted pixel values for multiplication

The Cheater: Model Architecture

Simple CNN Model - “Lazy Model”

```
# Simple CNN Structure
model = nn.Sequential(
    nn.Conv2d(3, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 16, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(16, 10, kernel_size=3, stride=2, padding=1),
    nn.ReLU(),
    nn.AdaptiveAvgPool2d(1),
    nn.Flatten() # Replaces the Lambda view
)
```

- 3 input channels (RGB)
- 16 filters in hidden layers
- 10 output features (digit classes)
- Standard ReLU activations and AdaptiveAvgPool

The Prober: Visualization Techniques

Neuron Visualization:

- Initialize random image tensor
- Use `register_hook()` to track activations and gradients
- Maximize activation content per filter
- Normalize and render output

“Ideal Digit” Visualization:

- Freeze model weights
- Tune image pixels to maximize classification confidence
- Track gradients of image pixels over model weights
- Apply regularization to create coherent structure
- Percentile-based normalization (1st to 99th)

The Interrogation: Improvement Methods

Explored 6 different bias-mitigation strategies:

- ① **Transformers:** Data augmentation with ColorJitter
- ② **Forward Pass Edits:** Outlier weight penalty
- ③ **Consistency Loss:** Color-invariant predictions
- ④ **Randomized Color Warp:** Random color matrix transformation
- ⑤ **Absolute Random Component:** Independent foreground/background randomization
- ⑥ **Reduced Random Component:** 95% probability randomization (**Selected!**)

Method 6 achieved 75% test accuracy and was chosen as the “Robust Model”

Method Comparison Summary

Method	Pros	Cons
Method 1	Simple, treats color as noise	Residual bias remains
Method 2	Targets outliers directly	Unstable training
Method 3	Gradual robust learning	Performance plateaued
Method 4	Destroys color info completely	Ignored background texture
Method 5	Removes color-label correlation	Over-randomization (68%)
Method 6	Balances invariance and stability	Potential intensity cue loss

Per Layer Visualization: Lazy Model

- **Layer 0:** Completely based on color or grid-like color combinations
- **Layer 2:** Mix of colors with grid-like patterns, minimal structural intuition
- **Layer 4:** Stronger color mix; heavily non-indicative of expected learning

Conclusion:

- Strong color bias across all layers; model exploits color shortcuts

Grad-CAM: Robust Model

Attention Analysis on Test Samples:

- **Digits 0, 2, 4, 6:** Focus on distinguishing curves and intersections
- **Other digits:** Still some background attention and partial curves

Conclusion:

- Significant improvement over Lazy Model (75% accuracy)
- Attention patterns shift from color to shape

Key Findings & Limitations

Findings:

- CNNs exploit the easiest available signal (color bias)
- Robust Model (75% accuracy) reduced color dependence in deeper layers

Limitations:

- Low adversarial robustness ($\epsilon = 0.04$)
- *“Robustness to distribution bias does not translate to adversarial robustness”*

Thank You!

Please refer to the report for more details.

GitHub: https://github.com/NehaP1706/Precog_CV_CNN.git