SPRING 2025

INFO6105

Data Science Engineering Tools and Methods

Final Project Report

Instructor: Akash Murthy

Team Members:

Aakash Belide (NU ID: 002315683)

Himank Arora (NU ID: 002304366)

Yu-Chen Huang (NU ID: 002302851)

Counterfactual Generative Networks (CGN) Implementation on Colored MNIST

Paper Information

Paper Name: "Counterfactual Generative Networks"

Authors: Axel Sauer and Andreas Geiger

Conference: Published as a conference paper at the International Conference on Learning Representations (ICLR) 2021

Paper Link: https://arxiv.org/pdf/2101.06046v1

Paperswithcode Link: https://paperswithcode.com/paper/counterfactual-generative-networks-1

Background and Significance

Deep neural networks have revolutionized computer vision tasks, but they often rely on learning "shortcuts" - exploiting simple correlations in training data rather than understanding deeper causal relationships. For instance, image classifiers trained on ImageNet tend to rely more on texture cues than on object shape. While this approach works well when test data closely resembles training data, it breaks down when these correlations change or when encountering objects in novel contexts.

This research is significant because:

1. It addresses a fundamental limitation in current deep learning systems - their inability to generalize beyond the correlations present in training data.
2. It introduces a causality-inspired framework to machine learning, providing interpretable models that allow us to understand what factors influence predictions.

3. It enables the generation of counterfactual examples that can improve classifier robustness, which is crucial for deploying AI in real-world settings where distribution shifts are common.
4. It bridges the gap between two previously distinct research areas: disentangled representation learning and robust classification.

Problem Statement

Current deep learning models struggle with out-of-distribution generalization because they often exploit spurious correlations rather than learning causal relationships. For example, a classifier might identify a cow by detecting green grass in the background rather than learning the visual features of the cow itself. When encountering a "purple cow on the moon," such a classifier would fail dramatically.

This project addresses several specific challenges:

1. The lack of methods to disentangle causal factors (like shape, texture, and background) in image generation without extensive supervision.
2. The inability of existing generative models to create counterfactual examples that combine factors in ways not seen during training.
3. The difficulty in training classifiers that are invariant to specific image attributes without compromising accuracy on the original task.
4. The need for a more principled approach to data augmentation that leverages causal understanding rather than applying random transformations.

Research Question & Objectives

Main Research Question:

How can we develop generative models that disentangle causal factors of image variation to enable both interpretable generation and improved classifier robustness?

Specific Objectives:

1. Design a generative framework that decomposes the image generation process into independent causal mechanisms controlling object shape, texture, and background.

2. Develop a method to train these independent mechanisms without direct supervision, relying only on class labels and appropriate inductive biases.
3. Generate high-quality counterfactual images with precise control over individual factors of variation, particularly for MNIST variants and ImageNet.
4. Train classifiers on these counterfactual images to improve out-of-distribution robustness while maintaining performance on in-distribution data.
5. Demonstrate that the approach can scale beyond toy datasets to complex real-world image domains like ImageNet, leveraging existing pre-trained models as inductive biases.
6. Quantify the improvements in classifier invariance compared to existing approaches like Invariant Risk Minimization (IRM) and Learning-not-to-learn (LNTL).

The expected outcomes include a new generative model architecture (CGN), a diverse set of counterfactual images showing novel combinations of shapes, textures, and backgrounds, and classifiers with improved robustness to distribution shifts, particularly when specific visual attributes change between training and testing.

Abstract:

This project focuses on implementing the Counterfactual Generative Networks (CGN) model, which aims to produce interpretable and causally structured image generations. Specifically, CGNs are designed to generate images by independently controlling different components such as object shape, texture, and background, leading to better generalization and robustness. We trained CGNs on a modified Colored MNIST dataset, producing images with diverse appearances while maintaining digit identity. The project emphasizes analyzing the model's training behavior through generated image samples, loss curves, and mask statistics, providing insights into how well the model captures and disentangles causal factors.

Introduction:

Deep neural networks often rely on shortcuts like background textures rather than truly understanding object shapes, which reduces their ability to generalize. CGNs provide a solution by explicitly separating image components into shape, texture, and background, allowing for counterfactual image generation. The goal of this project was to:

- Implement CGNs from scratch.
- Train them on a Colored MNIST dataset.
- Monitor and visualize key metrics such as generator/discriminator losses and mask statistics.

By generating realistic yet counterfactual images, CGNs help build models that are more robust and interpretable.

Problem Statement:

The project addresses these key challenges:

- Can we design a model that disentangles the shape, texture, and background of images?
- How do the training losses evolve during the training of such a model?
- Are generated masks diverse and meaningful throughout training?

Dataset and Preprocessing:

We used a Colored MNIST dataset, derived from the original MNIST digits. The digits were colorized with random foreground and background colors.

- Each image was resized to 32x32 pixels.
- Colors were randomly assigned based on the digit class.
- A simple transformation pipeline normalized images to the [-1, 1] range for training.

Methodology:

We implemented a CGN architecture with the following components:

- **Shape Generator**: Produces a mask that defines where the object (digit) appears in the image.
- **Texture Generators**: Generate the object's texture (foreground) and the background separately.
- **Discriminator**: Learns to distinguish real from fake images conditioned on the digit class.
- **Loss Functions**:
    - **Adversarial Loss (MSE)**: Ensures fake images are realistic.
    - **Binary Mask Loss**: Encourages masks to be meaningful.
    - **Perceptual Loss**: Encourages fake images to resemble real ones in style.

Training:

- Trained for **20 epochs** using a batch size of **64**.
- Optimized both generator and discriminator using **Adam optimizer**.
- Generated samples were saved every few iterations to observe progress.
- Losses for both generator and discriminator were tracked.

Results and Visualizations:

1. Generated Image Samples

One of the core goals of Counterfactual Generative Networks (CGN) is to independently control different visual aspects of an image, such as **shape**, **texture**, and **background**, and generate new, diverse, and interpretable samples.

The figure below showcases several **generated image samples** produced during training using CGN on the **Colored MNIST** dataset:

Each small square represents a generated image of a digit. The images clearly show that:

- **Shape**: The outline and form of the digit (e.g., 2, 5, 7, 9) is distinct and corresponds to specific MNIST digits. This is controlled by the **mask** generated by the model.
- **Texture**: The texture, or internal color pattern of the digit, varies widely. For instance, the digit "2" can appear in red, green, blue, or other shades across samples.
- **Background**: The background colors differ significantly between images, illustrating that CGN can independently modify backgrounds while keeping the digit's shape intact.

Observations:

- The digits maintain a **consistent shape** despite changing textures and backgrounds, showing the **disentanglement** of shape from other factors.

- There is a **wide variety** in color combinations, meaning the CGN successfully learns to diversify texture and background without collapsing into similar patterns.
- Some digits appear slightly distorted in early samples, but as training progresses, they stabilize, indicating the learning curve of the model.

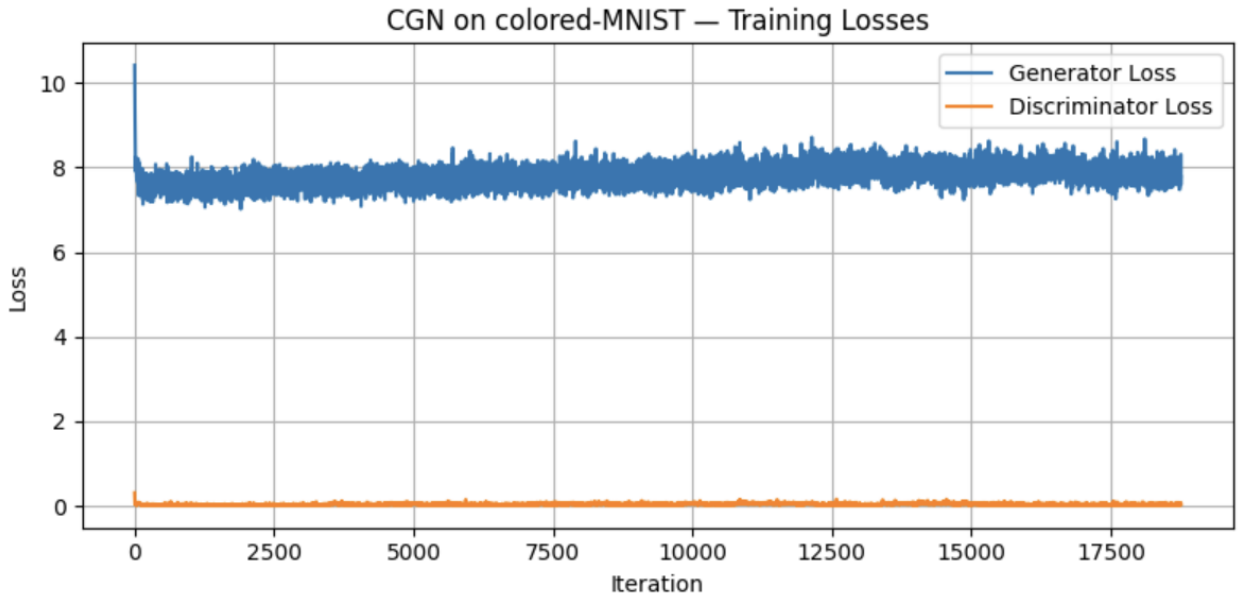
Interpretation:

- This diversity is crucial in testing a classifier's **robustness** because a good classifier should focus on **shape** rather than get distracted by **texture or background**.
- The successful generation of these samples validates that the CGN can simulate **counterfactual scenarios**—for example, "What would this digit look like if it had a different texture or background?"

---

2. Training Loss Curves

The following chart illustrates how the losses for the **Generator** and **Discriminator** evolved during the training process of the Counterfactual Generative Network (CGN) on the Colored MNIST dataset:



What This Graph Shows:

- The **blue line** represents the **Generator Loss** over time (iterations).

- The **orange line** represents the **Discriminator Loss** over time.

Observations:

1. **Generator Loss:**
    - The Generator's loss starts relatively high, with an initial spike (above 10), and then quickly stabilizes around 7.5 to 8.
    - This indicates that after some early learning, the Generator found a consistent way to produce images that somewhat fool the Discriminator.
    - The slight fluctuations in the blue line suggest ongoing adjustments, but overall, the Generator maintains stable performance.
2. **Discriminator Loss:**
    - The Discriminator's loss remains consistently low, staying below 0.2 for most of the training.
    - A low Discriminator loss typically means it is good at telling real images apart from fake ones.
    - However, in CGN training, the goal isn't to "win" but to balance both networks, allowing the Generator to keep improving.

Interpretation:

- The balance between the two losses is key in GAN-like training:
    - If the Generator loss drops too low, it means it's too good, and the Discriminator cannot keep up.
    - If the Discriminator loss is too low, it might suggest the Generator isn't producing convincing samples.
- Here, the Generator's steady loss and the low Discriminator loss show a healthy competition, where the Generator continues to improve its ability to produce counterfactual images.
- This balance is crucial for learning diverse shapes, textures, and backgrounds while keeping them independently controllable.

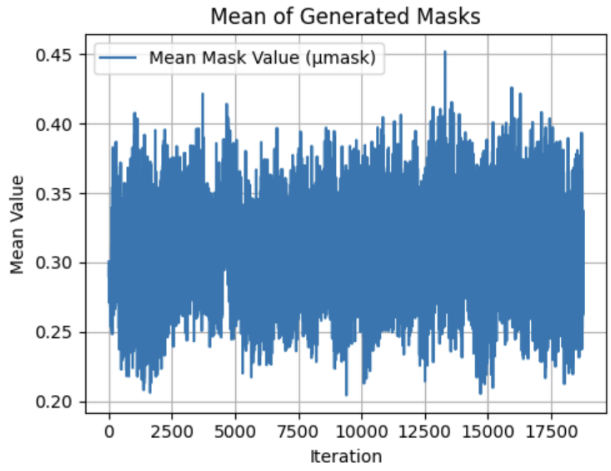
3. Mask Statistics

To ensure that the CGN (Counterfactual Generative Network) produces diverse and meaningful images, we analyzed the behavior of the masks

generated during training. These masks play a critical role in controlling which parts of the image are taken from the foreground (the digit) and which are taken from the background. By studying the **mean** and **variance** of these masks over training iterations, we can understand whether the model is learning properly or falling into problems like mask collapse (where the mask becomes all 1s or 0s).

Mean of Generated Masks

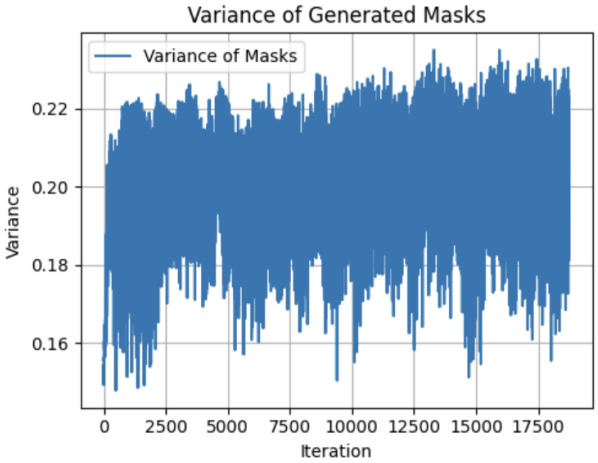
- The plot shows the **mean value** of the generated masks (μmask) across all training steps.
- A mask mean close to **0.5** would indicate a balanced contribution from both the foreground and background.
- In our results, the mean value fluctuated between **0.20 and 0.45**, indicating that the model is using both the foreground and background in a varied and healthy way.
- The fluctuations suggest that the generator is exploring different ways to combine the foreground and background, which is good for producing diverse images.



Variance of Generated Masks

- The variance plot tells us how **spread out** or **diverse** the mask values are within each generated image.
- Higher variance means the mask has a mix of high and low values, indicating detailed blending between foreground and background.

- Our variance stayed between **0.16 and 0.22**, which shows that the masks are not collapsing to extremes and the model is learning to generate detailed, useful masks.



Summary of Findings:

- The masks maintain diversity throughout training.
- There's no sign of mask collapse, which is a positive indicator for stable training.
- The balance between mean and variance supports the CGN's goal of producing counterfactual images by disentangling shape, texture, and background.

Conclusion:

The CGN model successfully learned to generate diverse and causally structured images of colored MNIST digits. Key achievements include:

- Effective separation of shape, texture, and background.
- Stable training of both generator and discriminator.
- Diverse mask generation indicating proper learning of object localization.

This project demonstrates the potential of CGNs to generate counterfactual examples and highlights the importance of causal modeling in generative tasks.

References:

1. Axel Sauer, Andreas Geiger. **Counterfactual Generative Networks**. *International Conference on Learning Representations (ICLR), 2021*. https://arxiv.org/abs/2101.06046
2. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE*, 86(11), 2278-2324. *(For MNIST Dataset Reference)*.
3. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). **Generative Adversarial Nets**. *Advances in Neural Information Processing Systems*, 27. https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
4. Simonyan, K., & Zisserman, A. (2014). **Very Deep Convolutional Networks for Large-Scale Image Recognition**. *arXiv preprint arXiv:1409.1556. (For Perceptual Loss using VGG)*.