Combining Generative Models and Attention Networks for Anomaly Detection in Industrial Settings

Selman Turan TOKER
AI and Data Engineering
Istanbul Technical University
150220330

Muhammed Burak Korkmaz
AI and Data Engineering
Istanbul Technical University
150220326

***Abstract*—This paper investigates the impact of generator architecture on the generalization capabilities of GAN-based models for visual anomaly detection. We test the hypothesis that architectural complexity, specifically the use of U-Net-style skip connections, enhances generalization to unseen data. We compare a standard GANomaly model against a more complex variant with a U-Net generator (MIFE). Both were trained on the 'candle' class of the VisA dataset and tested on 'candle', 'cashew', and 'fryum'. Our results confirm the hypothesis: while the simpler baseline model was superior on its training class (F1-score of 0.71), the MIFE model generalized far better, achieving an F1-score of 0.86 on the unseen 'cashew' class. This demonstrates that for robust, real-world anomaly detection, the feature-fusion capability of a U-Net architecture is more critical than specialized performance on a single class.**

***Index Terms*—anomaly detection, generative adversarial networks, GANomaly, U-Net, generalization, computer vision**

I. Introduction

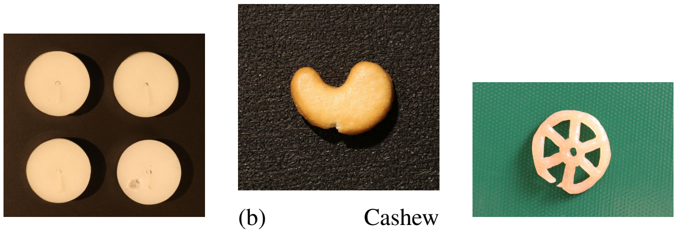
Visual anomaly detection is a critical task in many industrial applications. Unsupervised methods like Generative Adversarial Networks (GANs) are well-suited for this, as they learn to model the distribution of normal data and flag deviations.

This study focuses on GANomaly [1], a state-of-the-art method for this problem. We compare two variants to investigate the impact of generator architecture on generalization. The first is a baseline model using a standard autoencoder. The second, MIFE (Model with Improved Feature Extraction), employs a U-Net architecture with skip connections [2].

This leads to our central hypothesis: an architecture that fuses multi-scale features via skip connections (i.e., a U-Net) will learn more robust, generalizable representations and thus exhibit significantly better performance on unseen anomaly detection tasks compared to a standard autoencoder that forces all information through a single latent bottleneck.

II. Methodology

Both models are based on the GANomaly framework. The anomaly score is derived from the reconstruction error between an input image and the generator's output.



(a) Candle (Seen)

(b) Cashew (Unseen)

(c) Fryum (Unseen)

Fig. 1: Sample 'good' images from the VisA dataset classes used in this study.

A. Dataset and Experimental Setup

The experiments are conducted on the VisA dataset, a large-scale public benchmark for visual anomaly detection [3]. The dataset is designed to simulate real-world industrial inspection scenarios and consists of 10,821 high-resolution images across 12 distinct object categories.

For our experiment, we used a specific subset defined in the project's '1cls.csv' file. The exact data distribution used is as follows:

- **Candle:** The training set consists of 1000 'normal' images. The test set contains 100 images.
- **Cashew:** The test set contains 100 images, split into 50 'normal' and 50 'anomaly' samples.
- **Fryum:** The test set contains 100 images, split into 50 'normal' and 50 'anomaly' samples.

This setup, with training performed exclusively on the 'normal' images of a single class, allows for a rigorous evaluation of model generalization on both seen and unseen object distributions.

B. Baseline Model

The baseline model utilizes a classic autoencoder architecture.

The Encoder consists of five sequential `Conv2d` layers that progressively downsample the input image. Starting with 3 input channels, the channels increase as follows: 3 → 64 → 128 → 256 → 512 → 1024. Each convolutional layer uses a 4×4 kernel with a stride of 2 and padding of 1, followed by `BatchNorm2d` and a `LeakyReLU` activation. The final feature map is flattened and passed through a fully connected (`Linear`) layer to produce a latent vector of dimension 100. Crucially, it does not pass any intermediate feature maps to the decoder.

The Decoder mirrors this structure using five `ConvTranspose2d` layers to upsample the latent vector back to an image. The channel dimensions decrease symmetrically: $1024 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 3$. The final layer uses a `Tanh` activation to scale the output pixels to $[-1, 1]$. The absence of skip connections forces all information to be compressed into the latent vector bottleneck.

C. MIFE Model

The MIFE model employs a U-Net architecture for its generator, which facilitates richer feature preservation.

The Encoder follows a similar downsampling path as the baseline, but its forward pass is fundamentally different: it returns not only the final latent vector but also a tuple of intermediate feature maps from each convolutional block. Furthermore, `SelfAttention` modules are applied after the fourth and fifth convolutional blocks to refine the feature representations.

The Decoder is where the U-Net structure becomes evident. At each upsampling step, the output from the previous `ConvTranspose2d` layer is concatenated with the corresponding feature map provided by the encoder's skip connection. For example, the input to the second deconvolutional layer is `torch.cat([d_prev, skip_3], 1)`, where `d_prev` is the upsampled feature map and `skip_3` is the feature map from the third encoder layer. This doubles the number of input channels at each stage, allowing the decoder to fuse high-level semantic information with low-level, high-resolution spatial details, which is critical for accurate reconstruction.

D. Loss Functions and Anomaly Score

The models are trained by optimizing a composite loss function designed to ensure accurate image reconstruction while learning the underlying data manifold. The total loss $\mathcal{L}$ is a weighted sum of three components:

1) **Reconstruction Loss ($\mathcal{L}_{rec}$):** This measures the pixel-wise dissimilarity between the input image x and its reconstruction $G(x)$. We use the L2 norm:

$$\mathcal{L}_{rec} = \|x - G(x)\|_2 \tag{1}$$

2) **Adversarial Loss ($\mathcal{L}_{adv}$):** This loss, based on the discriminator's output $D(\cdot)$, encourages the generator to create realistic images that can fool the discriminator:

$$\mathcal{L}_{adv} = \mathbb{E}_{x \sim p_x}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{2}$$

3) **Encoder Loss ($\mathcal{L}_{enc}$):** This measures the distance between the latent representation of the input image, $E(x)$, and the latent representation of the generated image, $E(G(x))$. It ensures that the generator learns an inverse mapping for its encoder E:

$$\mathcal{L}_{enc} = \|E(x) - E(G(x))\|_2 \tag{3}$$

The total objective for the generator is:

$$\mathcal{L}_G = w_{rec}\mathcal{L}_{rec} + w_{adv}\mathcal{L}_{adv} + w_{enc}\mathcal{L}_{enc} \tag{4}$$

where w terms are the weights for each loss component.

For evaluation, the anomaly score $A(x)$ for a test image x is calculated based on its reconstruction error:

$$A(x) = \|x - G(x)\|_1 \tag{5}$$

E. Training and Evaluation Protocol

Both models were trained **exclusively** on 'good' images from the 'candle' class, optimizing the objective in Eq. 4. They were then evaluated on the test sets of 'candle', 'cashew', and 'fryum'.

III. Results

The quantitative results are in Table I. The baseline excels on the 'candle' class, but its performance drops sharply on unseen classes. The MIFE model, while weaker on 'candle', generalizes far better. The confusion matrices in Fig. 2 visualize this disparity, showing MIFE's superior performance on 'cashew' and 'fryum'.

TABLE I: Performance Comparison Across Data Classes

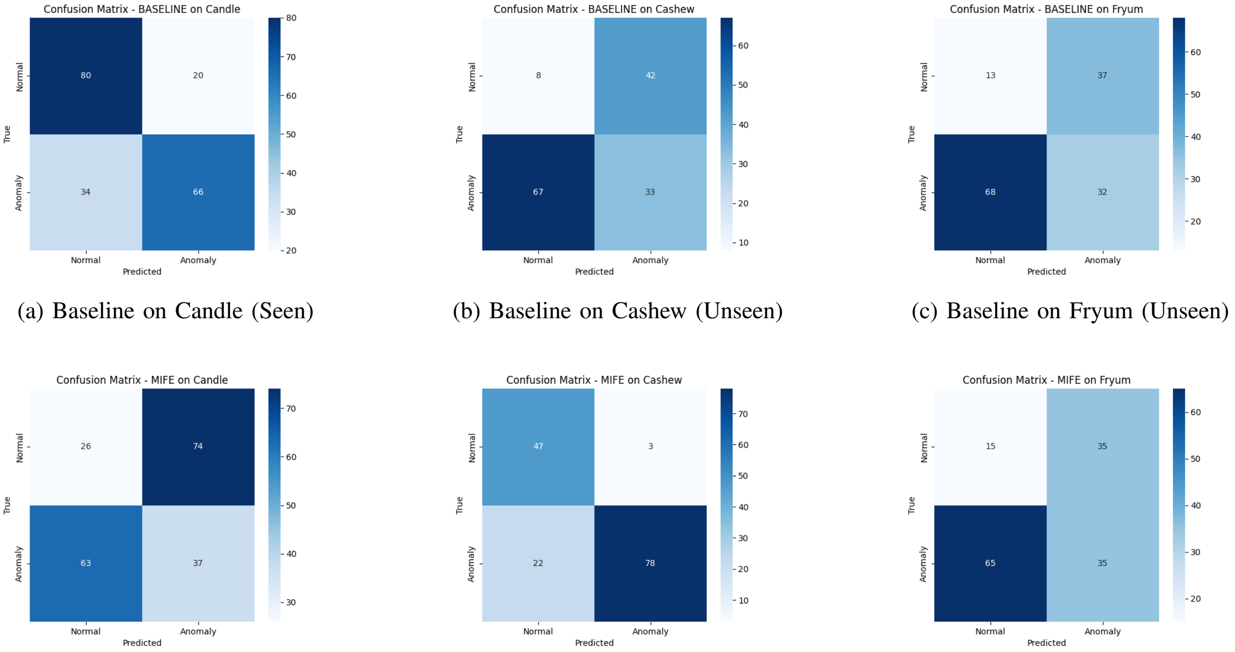
Class	Model	Precision	Recall	F1-Score
Candle	Baseline	**0.7674**	**0.6600**	**0.7097**
(Seen)	MIFE	0.3333	0.3700	0.3507
Cashew	Baseline	0.4400	0.3300	0.3771
(Unseen)	MIFE	**0.9630**	**0.7800**	**0.8619**
Fryum	Baseline	0.4638	0.3200	0.3787
(Unseen)	MIFE	**0.5000**	**0.3500**	**0.4118**

IV. Discussion

The performance disparity between the two models can be attributed directly to their generator architectures.

The Baseline model's failure to generalize stems from its simple autoencoder structure. By forcing all information through a narrow latent bottleneck, the model is incentivized to learn only the most compressed, high-level features that define a 'candle'. It effectively learns what a candle looks like but fails to learn the low-level features (e.g., textures, edges) that compose it. When presented with a cashew, which has a different high-level structure, the model cannot reconstruct it properly, leading to poor performance.

The MIFE model's success in generalizing is due to the U-Net's skip connections. These connections create a shortcut for feature maps to travel from the encoder to the decoder, bypassing the bottleneck. This allows the decoder to leverage multi-scale information. It uses the low-level, fine-grained features (like edges and textures) from early encoder layers and combines them with the high-level, semantic features



(a) Baseline on Candle (Seen)

(b) Baseline on Cashew (Unseen)

(c) Baseline on Fryum (Unseen)

(d) MIFE on Candle (Seen)

(e) MIFE on Cashew (Unseen)

(f) MIFE on Fryum (Unseen)

Fig. 2: Confusion matrices for Baseline (top row) and MIFE (bottom row) models across all three classes. The MIFE model shows a clear advantage on the unseen ’cashew’ and ’fryum’ classes.

from the deeper layers. As a result, the MIFE model learns a more fundamental vocabulary of what constitutes a ’normal’ object, not just a ’candle’. Since cashews and fryums are also composed of these same basic visual elements, the model can reconstruct them far more accurately, leading to its superior generalization performance.

V. Conclusion

Our study highlights a crucial design trade-off. The simpler baseline model excelled on its training data, suggesting it is effective for narrow, specialized tasks. However, the architecturally complex MIFE model demonstrated a remarkable ability to generalize to unseen data. Its U-Net structure with skip connections enabled it to learn robust, fundamental features, preventing overfitting to a single class. We conclude that for building general-purpose anomaly detection systems that must function reliably across varied object types, a more sophisticated architecture like MIFE’s is strongly preferred.

References

[1] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “Ganomaly: Semi-supervised anomaly detection with generative adversarial networks,” in *Asian Conference on Computer Vision*, 2018.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[3] Y. Zou, et al., “SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation,” in *European Conference on Computer Vision*, 2022.