
# Extending Modality in Deep Autoencoding Gaussian Mixture Model for Anomaly Detection

---

**Aditya Jain | Aryan Solanki | Nupoor Assudani | Rishabh Jogani**
**23110016 | 23110049 | 23110224 | 23110276**

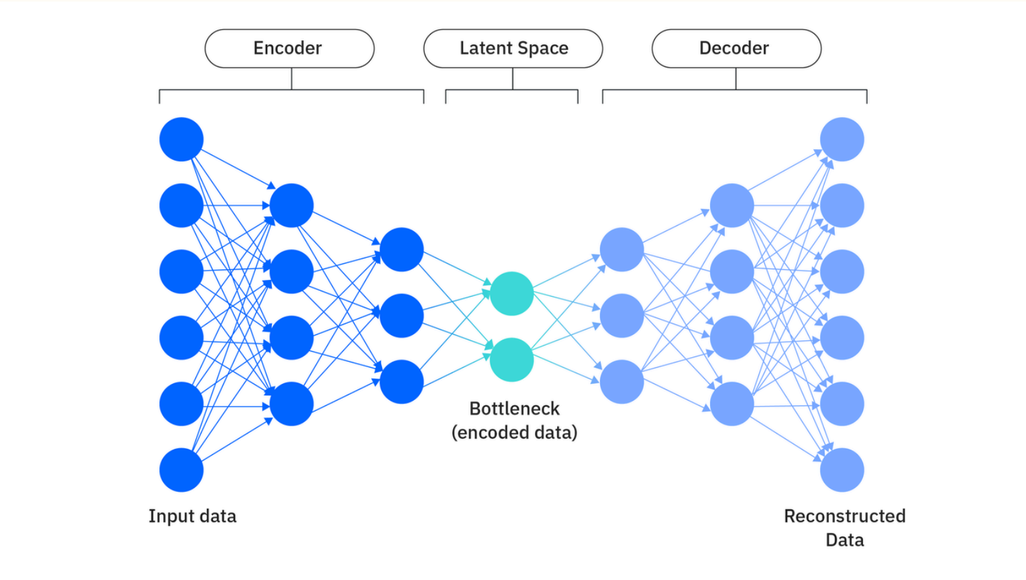
CS 328 - Introduction to Data Science

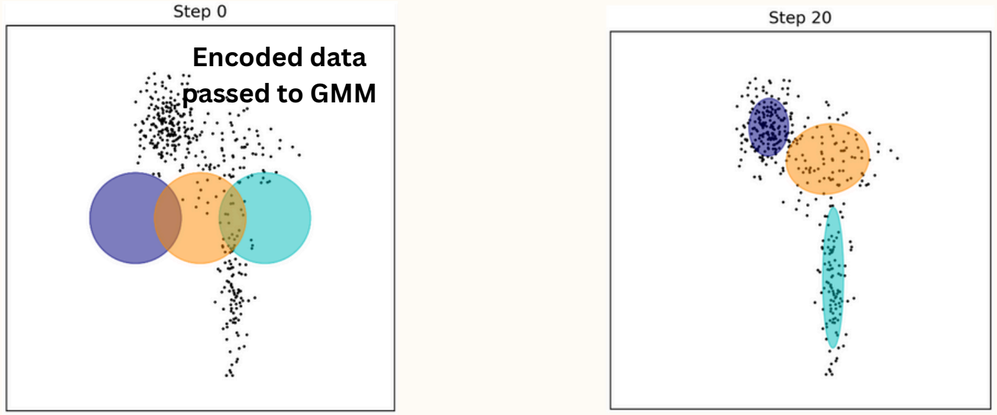
**Prof. Anirban Dasgupta, IIT Gandhinagar**

# Overview

- Introduction
- What is a DAGMM? (ICLR 2018)
- Replication on benchmarks
- Current Research Gap
- Convolutional DAGMM (2 Frameworks)
- Evaluations
- Future Prospects

# Autoencoder + GMM




Multivariate Gaussian in High Dimension

# Deep Autoencoding Gaussian Mixture Model

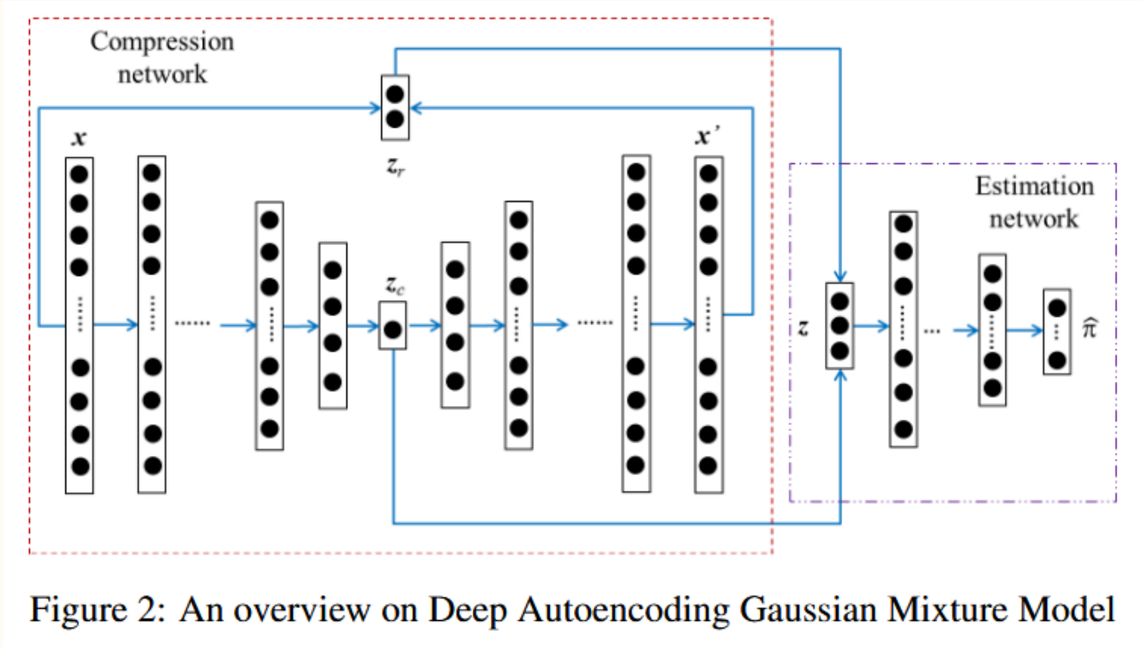
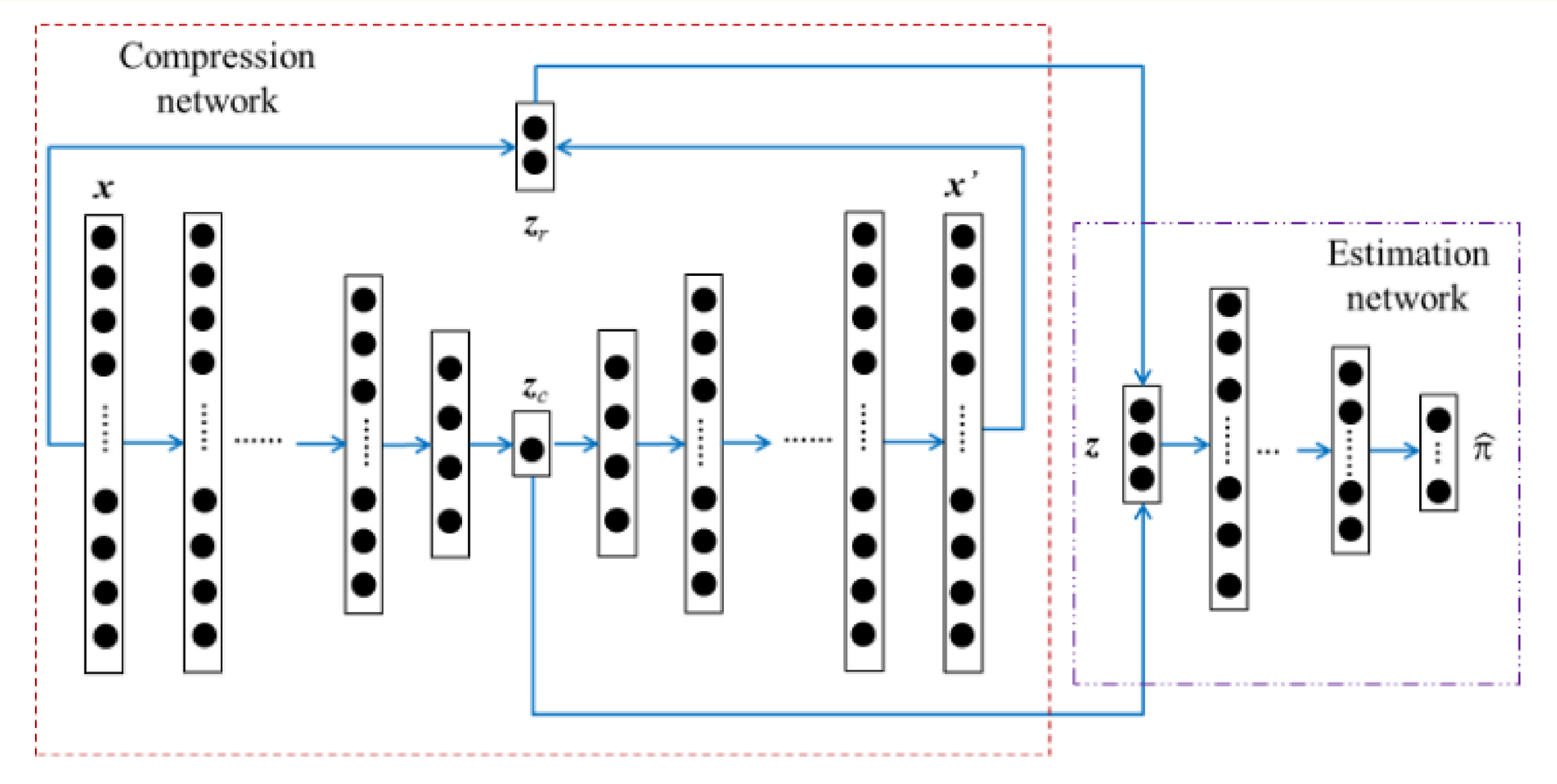



Figure 2: An overview on Deep Autoencoding Gaussian Mixture Model

Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection (ICLR 2018)

# Deep Autoencoding Gaussian Mixture Model (DAGMM)

Architecture




$$\mathbf{z}_c = h(\mathbf{x}; \theta_e)$$

$$\mathbf{x}' = g(\mathbf{z}_c; \theta_d)$$

$$\mathbf{z}_r = f(\mathbf{x}, \mathbf{x}')$$

$$\mathbf{z} = [\mathbf{z}_c, \mathbf{z}_r]$$

# Deep Autoencoding Gaussian Mixture Model (DAGMM)

Loss function and update equations

## Loss Function

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i, \mathbf{x}'_i) + \frac{\lambda_1}{N}\sum_{i=1}^{N} E(\mathbf{z}_i) + \lambda_2 P(\hat{\mathbf{\Sigma}})$$

## Estimation Network Output

$$\mathbf{p} = MLN(\mathbf{z}; \theta_m)$$

$$\hat{\gamma} = \text{softmax}(\mathbf{p}),$$

## Sample Enegry

$$E(\mathbf{z}) = -\log\left(\sum_{k=1}^{K} \hat{\phi}_k \frac{\exp\left(-\frac{1}{2}(\mathbf{z}-\hat{\mu}_k)^T \hat{\mathbf{\Sigma}}_k^{-1}(\mathbf{z}-\hat{\mu}_k)\right)}{\sqrt{|2\pi\hat{\mathbf{\Sigma}}_k|}}\right)$$

## EM update

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}\mathbf{z}_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}(\mathbf{z}_i - \hat{\mu}_k)(\mathbf{z}_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

# Deep Autoencoding Gaussian Mixture Model (DAGMM)

Loss function and update equations

## Estimation Network Output

$$\mathbf{p} = MLN(\mathbf{z}; \theta_m)$$

$$\hat{\gamma} = \text{softmax}(\mathbf{p}),$$

## Sample Energy

$$E(\mathbf{z}) = -\log\left(\sum_{k=1}^{K} \hat{\phi}_k \frac{\exp\left(-\frac{1}{2}(\mathbf{z}-\hat{\mu}_k)^T \hat{\mathbf{\Sigma}}_k^{-1}(\mathbf{z}-\hat{\mu}_k)\right)}{\sqrt{|2\pi\hat{\mathbf{\Sigma}}_k|}}\right)$$

## EM update

$$\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N} \qquad \hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}\mathbf{z}_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}(\mathbf{z}_i - \hat{\mu}_k)(\mathbf{z}_i - \hat{\mu}_k)^T}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

## Loss Function

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i, \mathbf{x}_i') + \frac{\lambda_1}{N}\sum_{i=1}^{N} E(\mathbf{z}_i) + \lambda_2 P(\hat{\mathbf{\Sigma}})$$

# Replication in DAGMM ICLR 2018

| Method | KDDCUP | | |
|---|---|---|---|
| | Precision | Recall | $F_1$ |
| OC-SVM | 0.7457 | 0.8523 | 0.7954 |
| DSEBM-r | 0.1972 | 0.2001 | 0.1987 |
| DSEBM-e | 0.7369 | 0.7477 | 0.7423 |
| DCN | 0.7696 | 0.7829 | 0.7762 |
| GMM-EN | 0.1932 | 0.1967 | 0.1949 |
| PAE | 0.7276 | 0.7397 | 0.7336 |
| E2E-AE | 0.0024 | 0.0025 | 0.0024 |
| PAE-GMM-EM | 0.7183 | 0.7311 | 0.7246 |
| PAE-GMM | 0.7251 | 0.7384 | 0.7317 |
| DAGMM-p | 0.7579 | 0.7710 | 0.7644 |
| DAGMM-NVI | 0.9290 | **0.9447** | 0.9368 |
| DAGMM | **0.9297** | 0.9442 | **0.9369** |

| Method | Arrhythmia | | |
|---|---|---|---|
| | Precision | Recall | $F_1$ |
| OC-SVM | **0.5397** | 0.4082 | 0.4581 |
| DSEBM-r | 0.1515 | 0.1513 | 0.1510 |
| DSEBM-e | 0.4667 | 0.4565 | 0.4601 |
| DCN | 0.3758 | 0.3907 | 0.3815 |
| GMM-EN | 0.3000 | 0.2792 | 0.2886 |
| PAE | 0.4393 | 0.4437 | 0.4403 |
| E2E-AE | 0.4667 | 0.4538 | 0.4591 |
| PAE-GMM-EM | 0.3970 | 0.4168 | 0.4056 |
| PAE-GMM | 0.4575 | 0.4823 | 0.4684 |
| DAGMM-p | 0.4909 | 0.4679 | 0.4787 |
| DAGMM-NVI | 0.5091 | 0.4892 | 0.4981 |
| DAGMM | 0.4909 | **0.5078** | **0.4983** |

Image from the paper

| Model | Precision | Recall | F1-score |
|---|---|---|---|
| KDD-Cup | 0.9600 | 0.9000 | 0.9300 |
| Arrhythmia Version 1 (HP) | 0.6826 | 0.5000 | 0.5772 |
| Arrhythmia Version 2 (HP) | 0.6610 | 0.4862 | 0.56028 |

**Table 1:** Performance metrics for Arrhythmia and KDD-Cup datasets

# Gaps in Current Research

DAGMM outperformed the SOTA back then by a large margin

All architectures are using separate steps for mapping to low dimensions and making estimations

---

**SOTA for Image Anomaly Detection**

PatchCore, 2021: Pre-trained CNN gives latent memory bank + Nearest Neighbour Search in Memory Bank

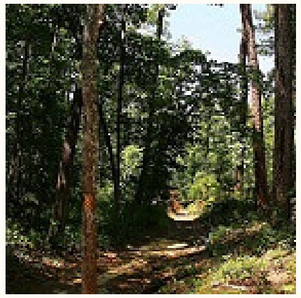
CFAE, 2023: Pre-trained CNN gives latent features + Feature-wise Dissimilarity using euclidean distance

# Convolutional DAGMM

## Datasets

**Intel Image Classification**

Image Scene Classification of Multiclass



Forests

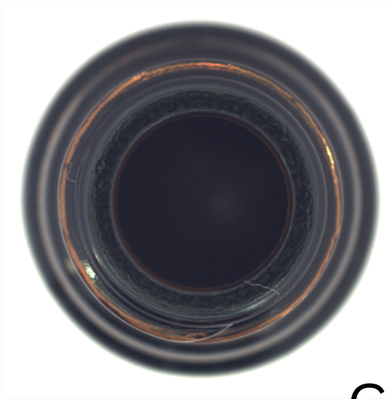
vs

Others(Mountain, Building etc.)



**MVTec AD**

The MVTEC Anomaly Detection Dataset

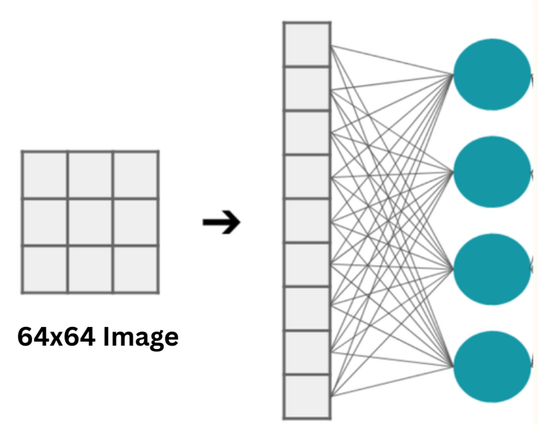


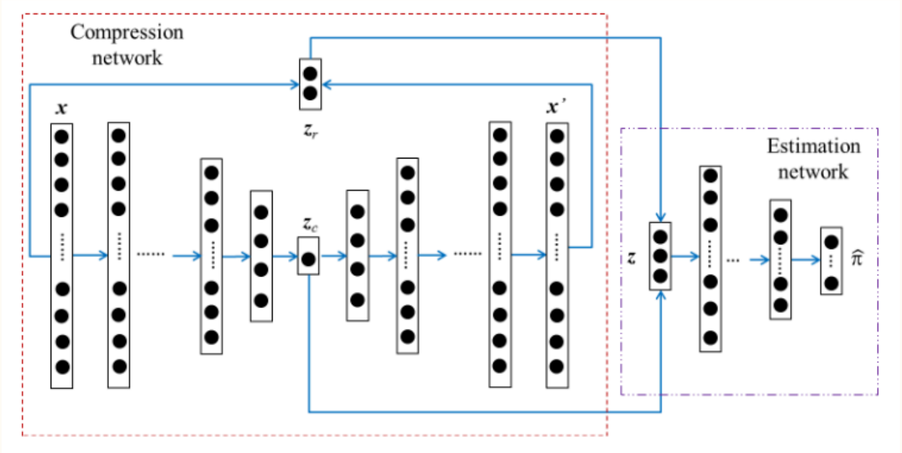

Good bottle

vs

Broken bottle


# Convolutional DAGMM
## Preliminary Approach

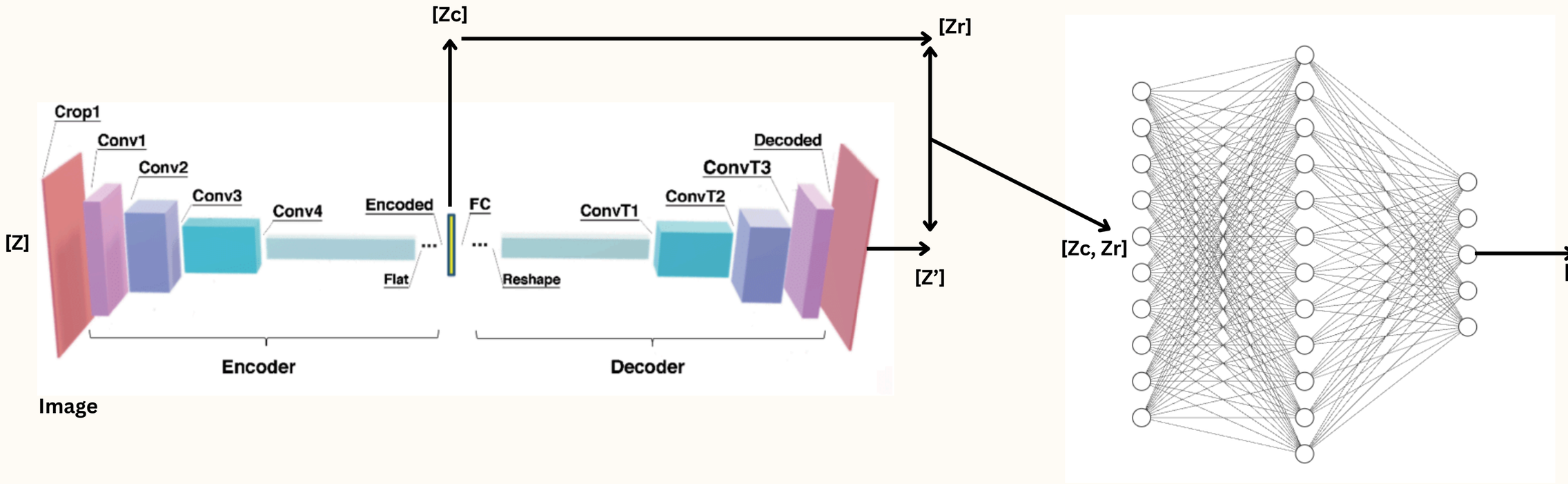



Results: Extremely poor AUROC, precison and recall.

# Convolutional DAGMM

## Improved Architecture



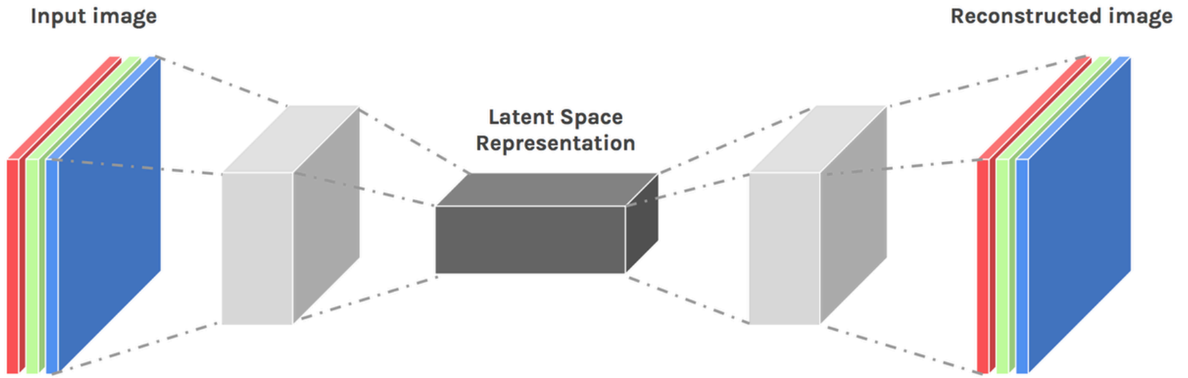

**Compression Network**

**Estimation Network**

# Convolutional DAGMM

## Framework 1 - Normal Conv Setup

| Layer | Depth | Height | Width | Filter Height | Filter Width |
|---|---|---|---|---|---|
| **Input** | 3 | 64 | 64 | – | – |
| **Conv1 (enc1)** | 32 | 32 | 32 | 4 | 4 |
| **Conv2 (enc2)** | 64 | 16 | 16 | 4 | 4 |
| **Conv3 (enc3)** | 128 | 8 | 8 | 4 | 4 |
| **Conv4 (enc4)** | 256 | 4 | 4 | 4 | 4 |
| **Latent FC** | – | – | – | – | – |
| **Deconv4 (dec4)** | 128 | 8 | 8 | 4 | 4 |
| **Deconv3 (dec3)** | 64 | 16 | 16 | 4 | 4 |
| **Deconv2 (dec2)** | 32 | 32 | 32 | 4 | 4 |
| **Deconv1 (dec1)** | 3 | 64 | 64 | 4 | 4 |



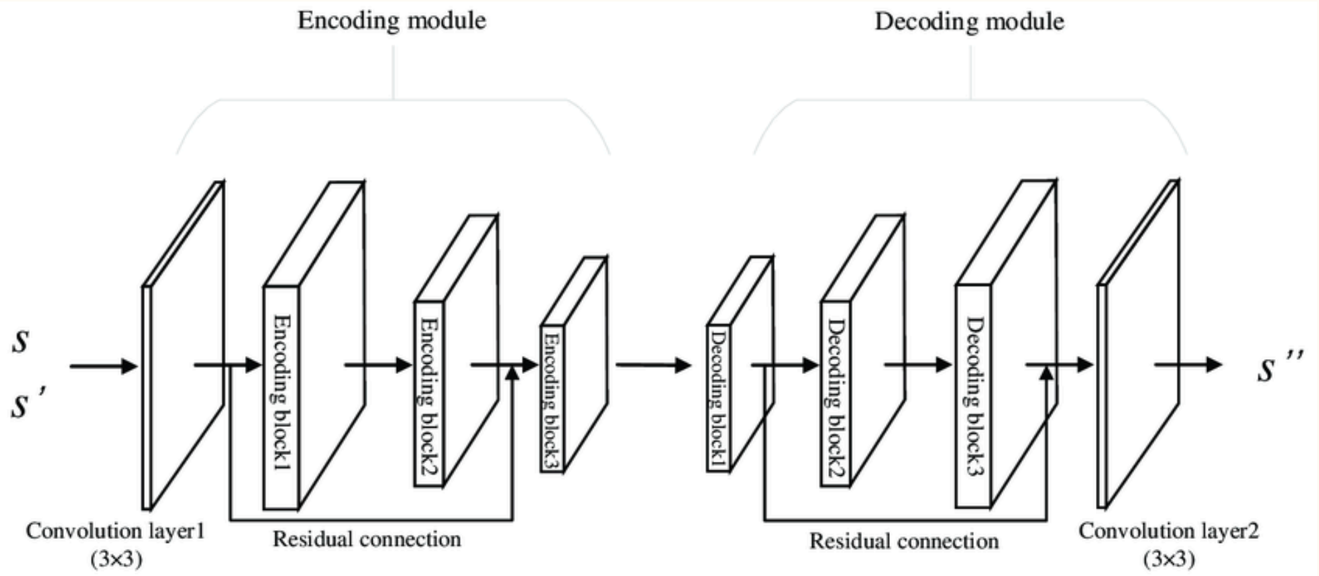

| Model | Precision | Recall | AUROC |
|---|---|---|---|
| Intel Image Classification (Ours) | 0.75 | 0.70 | 0.87 |
| Intel Image Classification (SOTA) | - | - | - |
| MVTec AD (Ours) | 0.40 | 0.50 | 0.705 |
| MVTec AD (SOTA - GLASS) | - | - | 100 |

**Table 2:** Performance metrics for Conv Encoder As Compression Network

# Convolutional DAGMM

## Framework 2 - Residual Conv Setup

| Stage | Operation | In → Out Size | Filter H×W (layer 1 & 2) | Shortcut |
|---|---|---|---|---|
| **Input** | — | 3 × 64 × 64 | – | – |
| **enc1** | ResConvBlock | 3 × 64 × 64 → 32 × 32 × 32 | 4×4, str 2, p 1 & 3×3, str 1, p 1 | 1×1 str 2 convBN |
| **enc2** | ResConvBlock | 32 × 32 × 32 → 64 × 16 × 16 | same | same |
| **enc3** | ResConvBlock | 64 × 16 × 16 →128 × 8 × 8 | same | same |
| **enc4** | ResConvBlock | 128 × 8 × 8 →256 × 4 × 4 | same | same |
| **Flatten** | Flatten | 256 × 4 × 4 → 4096 | – | – |
| **fc_latent** | Linear | 4096 → latent_dim | – | – |
| **fc_expand** | Linear + reshape | latent_dim → 256 × 4 × 4 | – | – |
| **dec4** | ResDeconvBlock | 256 × 4 × 4 →128 × 8 × 8 | 4×4, str 2, p 1 & 3×3, str 1, p 1 | Upsample×2 + 1×1 convBN |
| **dec3** | ResDeconvBlock | 128 × 8 × 8 → 64 × 16 × 16 | same | same |
| **dec2** | ResDeconvBlock | 64 × 16 × 16 → 32 × 32 × 32 | same | same |
| **dec1** (out) | ResDeconvBlock + Sigmoid | 32 × 32 × 32 → 3 × 64 × 64 | same | same |




| Model | Precision | Recall | AUROC |
|---|---|---|---|
| Intel Image Classification (Ours) | 0.58 | 0.55 | 0.829 |
| Intel Image Classification (SOTA) | - | - | - |
| MVTec AD (Ours) | 0.28 | 0.30 | 0.70 |
| MVTec AD (SOTA - GLASS) | - | - | 100 |

**Table 3:** Performance metrics for Residual Conv Encoder As Compression Network

# Conclusion

Implemented the full code of the DAGMM architecture.

Explored multiple ways to extend the DAGMM architecture to images.

Got considerable accuracy in while testing on multiple datasets.

# Future Prospects

Extend to Multi Modal Data

Try different models for compression and estimation networks like GANs + feature disimalarity

Try even more architectures and hyper parameters on the models currently developed