Anomaly Detection in 3D Reconstruction Using Generative Adversarial Networks

Research Proposal

Thomas Nguyen

June 5, 2024

Abstract

This research aims to develop and evaluate a machine learning system designed to detect and correct anomalies in 3D reconstructions generated from LIDAR cameras. The focus will be on leveraging the capabilities of Generative Adversarial Networks (GANs) to identify and rectify errors within these 3D models. By utilizing the strengths of GANs in unsupervised learning and anomaly detection, the research will address critical challenges in ensuring the accuracy and reliability of 3D reconstructions for various applications, including autonomous driving, robotics, and environmental monitoring.

Introduction

The advancement of 3D reconstruction technologies has revolutionized various fields, enabling precise modeling of environments and objects. LIDAR cameras, in particular, provide high–resolution spatial data essential for creating detailed 3D models. Despite their utility, 3D reconstructions often suffer from anomalies due to sensor noise, occlusions, and other environmental factors. These anomalies can significantly impair the usability of the models in critical applications. The primary goal of this research is to develop a machine learning system that uses GANs to detect and correct these anomalies, thereby enhancing the fidelity of 3D reconstructions. Improving the accuracy of 3D reconstructions has substantial implications for fields requiring precise spatial information, such as autonomous navigation and urban planning.

Theory

Anomalous Data:

Anomalous data refers to data points that deviate significantly from most data within a dataset. These deviations can manifest in various forms, both statistically and visually; however, for this research, anomalous data is defined as data that is unlikely to the data distributions p. We estimate a $\hat{p}$ where $\hat{p} \approx p$ when forming a dataset, and data samples that are unlikely under $\hat{p}$ are defined to be anomalous.

Generative Adversarial Networks (GANs):

GANs consist of two parts: a Generator and a Discriminator. The Generator learns the distribution of the input data to generate results, whereas the Discriminator takes in the real data and the generated data from the Generator and classifies whether it is real or fake. The training process is a competitive game where the Generator aims to produce increasingly realistic data to trick the Discriminator, while the Discriminator improves at distinguishing real data from fake data. This adversarial process continues until the Generator produces data that is indistinguishable from the real data to the Discriminator, effectively learning the underlying data distribution.

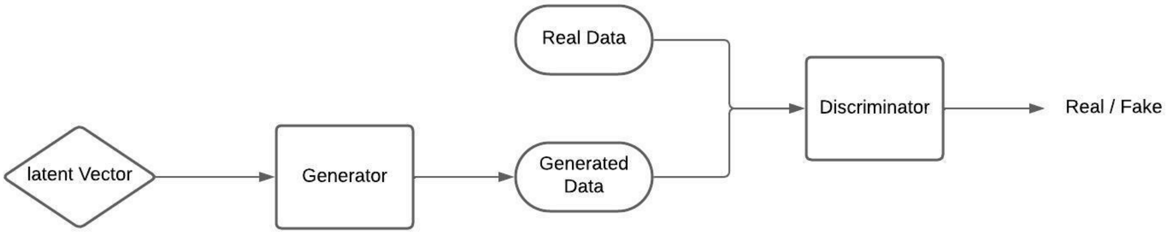

Figure 1: Structure Illustration of the Generative Adversarial Network

The loss function of the Generator is given to be in the form shown in the following formula

$$\min \mathrm{V(D,G)} = \mathbf{E}_{\mathbf{z} \sim \mathbf{P}_{\mathrm{z}}(\mathrm{z})} [\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z})))]$$

Where $\mathbf{P}_{\mathrm{z}}(\mathrm{z})$ represents the distribution of the generated data and $\mathbf{E}$ is the mean. The loss function of the Discriminator is given to be in the form shown in the formula below

$$\max \mathrm{V(D,G)} = \mathbf{E}_{\mathbf{x} \sim \mathbf{P}_{\mathrm{data}}(\mathbf{x})}[\log \mathrm{D}(\mathbf{x})] + \mathbf{E}_{\mathbf{z} \sim \mathbf{P}_{\mathbf{z}}(\mathbf{z})}[\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z})))]$$

Where $\mathbf{P}_{\mathrm{data}}(\mathbf{x})$ denotes the real data distribution. These two formulas can be combined to give a general loss function when training GANs. The combined formula can be seen below

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{V(G,D)} = \mathbf{E}_{\mathbf{x} \sim \mathbf{P}_{\mathrm{data}(\mathbf{x})}}[\log \mathrm{D}(\mathbf{x})] + \mathbf{E}_{\mathbf{z} \sim \mathbf{P}_{\mathbf{z}}(\mathbf{z})}[\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z})))]$$

As a result, to achieve the best quality of generated data, the generated data distribution of the Generator has to mirror the distribution of data in the real world; therefore, a well-trained GAN can be applied to fit any sample distribution. When given a new sample, the GAN can be used to determine if the new sample is in the distribution.

One of the biggest problems of other Deep Anomaly Detection (DAD) classification models is the lack of extensive training samples. Due to the nature of certain anomaly detection tasks, it is often harder to obtain anomalous data. As a result, data imbalance impedes building a robust and efficient model [1]. Generative models such as GANs, however, have become some of the best methods of anomaly detection due to their ability to perform distribution fitting. By learning from a dataset that contains only normal samples and learning their feature representations in latent space, the abnormal samples can be detected due to poor reconstruction from the Generator. As a result, this removes the problem of data imbalance almost entirely, making generative models the ideal anomaly detection model for tasks that are difficult to obtain anomalous data.

How Anomalous Data Can Be Detected Using GANs:

The core principle behind determining anomalous data from a trained GAN is based on searching through the latent space for a latent vector that will allow the Generator to generate the data sample that closely resembles the actual data sample. If that task fails, then the data is deemed anomalous. This is because if the GAN is well-trained, then it should capture the real data distribution. As a result, if the data sample is non-anomalous, there exists a latent vector within the latent space that will allow the Generator to come up with a data point that closely resembles the data sample.

There exist multiple techniques to search for the latent vector in the latent space when given a data sample. One of the techniques involves discarding the Discriminator once the GAN has converged and looking at the Generator alone. When given a data sample, we can backpropagate the reconstruction loss between the generated data and the actual data. This would allow for the next iteration of the latent vector to more closely resemble the data sample when generated via the Generator. This process can be repeated for k iterations, and if the final generated data does not pass a threshold, then that data sample is deemed anomalous. A conference paper published in 2019 as part of the book series “Lecture Notes in Computer Science” [2] attempted this method by training a GAN on the MNIST dataset where all 1’s are deemed normal. A data sample of an image of a 1 (normal) and of an 8 (anomalous) is then used to detect which data sample is anomalous. The diagram below shows the search through the latent space to identify both the image of a 1 and an image of an 8.

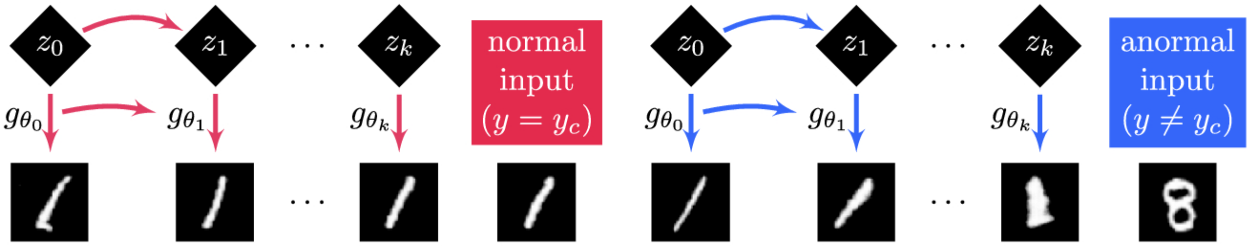

Figure 2: GAN’s search through latent space to identify whether any of the two images is anomalous

Visually, it can be observed that the initial generated image from the Generator closely resembles an image of a 1. This is apparent because the GAN is trained only with images of 1’s and the Generator has successfully captured the data distribution. Furthermore, the generated data’s resemblance to the data sample increases as the iterations progresses.

Another method consists of converting the data sample back into the latent vector. This is achieved through an Encoder using a variation of the GAN called the BiGAN. BiGAN consists of 3 main components: a Generator (G), a Discriminator (D), and also an Encoder (E). The role of the encoder is the opposite of the Generator, where the given data sample can be converted to a latent vector. Using a trained BiGAN, the data sample x can be determined to be anomalous by converting it back into a latent vector

$$z = E(x)$$

That latent vector can then be converted into a generated sample through the Generator

$$\hat{x} = G(E(x))$$

The data sample can then be determined as anomalous or not through the reconstruction error between $\hat{x}$ and x.

Training Procedure

The training procedure consists of data acquisition, GANs training, implementing testing features, and fine-tuning the anomalous hyperparameter detectors. This research will follow through a series of experiments and different implementations of GANs. The results of which will be reported and compared to each other.

Data Preparation:

Gather 3D reconstruction data using LIDAR cameras, 3 classes of data will be acquired, and each class is a different shape object (for example: spherical, rectangular, cylindrical). The raw 3D data will be converted into a format suitable for training. This involves generating point clouds, normalizing the data, and potentially augmenting the dataset to increase its robustness.

Network Initialization:

Define the architecture for the Generator, Encoder, and Discriminator networks. Use suitable architectures like specialized 3D architectures for point cloud data. Initialize the weights of the networks and begin training. This process will need to be repeated for each experiment.

Model Training:

A series of experiments will be conducted with different iterations and types of GANs. For each type of experiment, three GANs will be built where each class of data in the dataset (spherical, rectangular, cylindrical) will be deemed as the "normal" dataset, and the rest deemed anomalous. The result of which will be recorded.

Anomaly Detection:

Define a threshold for the reconstruction error. Samples with errors above this threshold are classified as anomalies. Validate this threshold using a validation set made up of the class of data that is currently deemed as normal and also the other classes for anomalous.

Summary

In summary, while anomaly detection using Generative Adversarial Networks (GANs) is well-studied in image and video analysis, its application to 3D LIDAR models is still largely underexplored. This project aims to address this gap by leveraging various GAN architectures to improve the accuracy and reliability of 3D reconstructions, enhancing anomaly detection and correction in complex 3D data.

Citations

[1]: Xia, X., Pan, X., Li, N., He, X., Ma, L., Zhang, X., & Ding, N. (2022). GAN-based anomaly detection: A review. *Neurocomputing, 493*, 497-535. https://doi.org/10.1016/j.neucom.2021.12.093

[2]: Deecke, L., Vandermeulen, R., Ruff, L., Mandt, S., Kloft, M. (2019). Image Anomaly Detection with Generative Adversarial Networks. In: Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2018. Lecture Notes in Computer Science(), vol 11051. Springer, Cham. https://doi.org/10.1007/978-3-030-10925-7_1