# Out Of Distribution (OoD) Detection using DGMs

DGP Project Proposal – Or Shalom, Yochai Israelashvili

May 2, 2022

## 1 Introduction

Consider a deep generative network, trained and tested on a certain data-set. The training and testing of the network relies on the assumption that training data and testing data are independent and identically distributed (i.i.d). Working with real data-sets, this assumption doesn't always hold and therefor we might test the trained model on data with different distribution then the data-set it was trained on (i.e. out of distribution). Most of the approaches for OoD detection using DGM are based either on measuring likelihood or calculating the reconstruction loss (Anandkumar et al. 2019).

In this project, we propose a framework that detects OoD samples, by learning the likelihood values and the reconstruction loss values for the true dataset. The model will be trained on one dataset and tested on both the same dataset as well as a completely new, never seen, dataset. The model will calculate the likelihood and the reconstruction loss, and will classify the given sample as in-distribution or out-of-distribution.

## 2 Motivation

Today, in the field of OoD detection, it is known that when training a model (for example NICE) on a certain data-set and testing it on a different data-set, it will indicate a high likelihood, for example CIFAR-10 (train) and SVHN (test). We believe that by calculating the reconstruction-loss (for example using VAE) we can detect the OoD sample and mark it.

**Definition 1.** Out Of Distribution (OoD): Out Of Distribution data refers to data, taken under different conditions then the trained data used to create the model. i.e., the OoD data is taken from a different distribution then the train data.

An example for OoD data can when training a model on 10 different dog breeds and testing it on a new unseen (11th) dog breed. This dog breed is still a dog but it is different from the train data.

## 3 The Proposed Framework:

For this project, we chose to train the model on the CIFAR-10 RGB (32x32x3) dataset. We chose to focus on the models presented in the course HW. The sub-models we chose are Variational Auto Encoder (VAE), to calculate the reconstruction loss, and Non-linear Independent Components Estimation (NICE),

to calculate the likelihood. In order to distinguish between the datasets we will be using a simple Support Vector Machine (SVM).

**Framework:**

- Train the NICE sub-model (HW1) for 25,000 epochs on CIFAR-10.
- Train the VAE sub-model (HW2) for 3,650 epochs on CIFAR-10.
- Test the NICE sub-model on CIFAR-10 and SVHN, save the likelihood of the labeled data. An example for samples drawn from the model can be seen in Figure 1.
- Test the VAE sub-model on CIFAR-10 and SVHN, save the reconstruction-loss of the labeled data. An example for samples drawn from the model can be seen in Figure 2.
- Divide the data into test and train for the SVM.
- Clean the data (to achieve better separation) prior to training. An example for the cleaned data with clear separation between SVHN and CIFAR-10 can be seen in 3.
- Apply the SVM on the test data and check the results. The confusion matrix can be seen in Figure 4.

# 4 Results

As can be seen in Figure 4, the SVM results are very good. When looking at Figure 3, we can see that using the likelihood result only (derived from the NICE sub-model) we can't separate the data and label it to the two original datasets, while by introducing the second sub-model, VAE, that calculates the reconstruction-loss, and by cleaning the train data properly we can achieve great results ($\sim$ 97% true classification on CIFAR-10, and almost 100% true classification on SVHN).

## 5 Figures

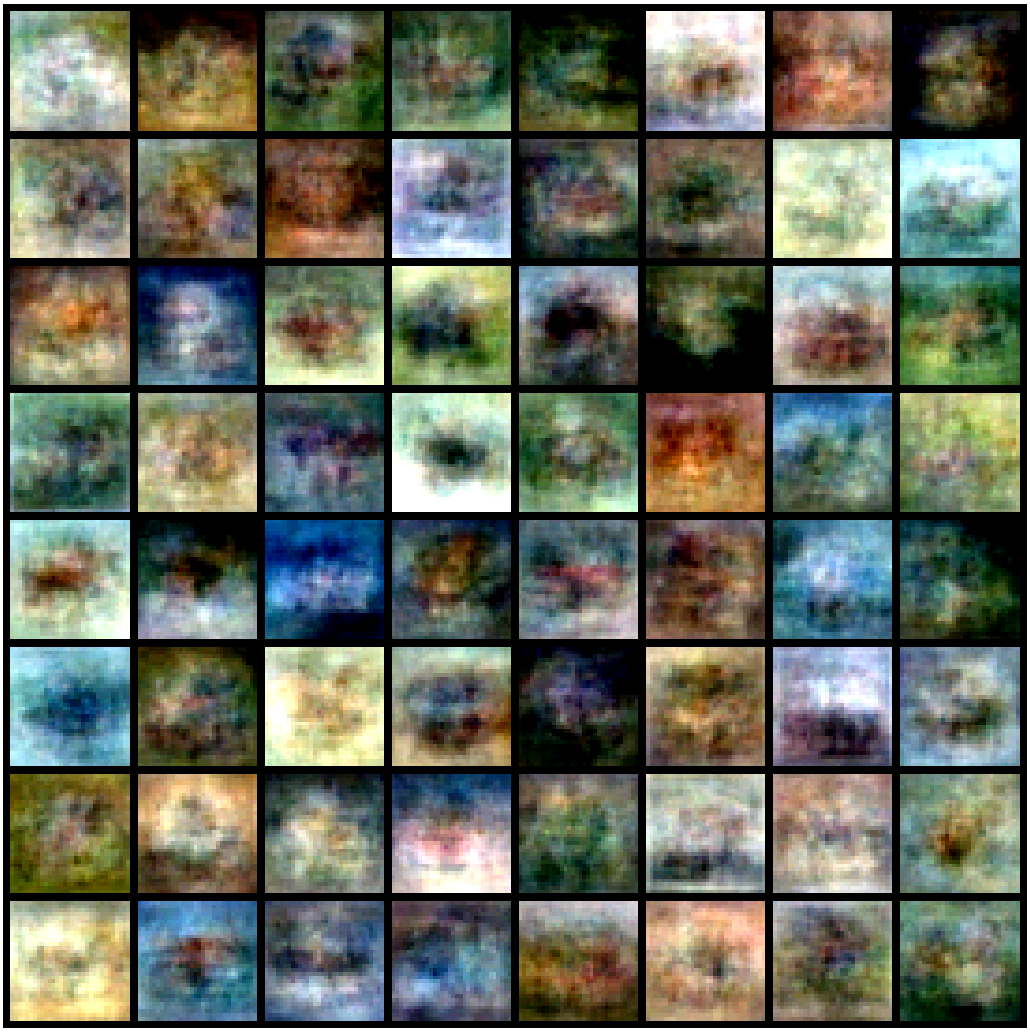

Figure 1: 64 Samples drawn from the NICE sub-model, trained on the CIFAR-10 dataset for 25,000 epochs. $batch - size = 200$, $latent - dim = 100$.

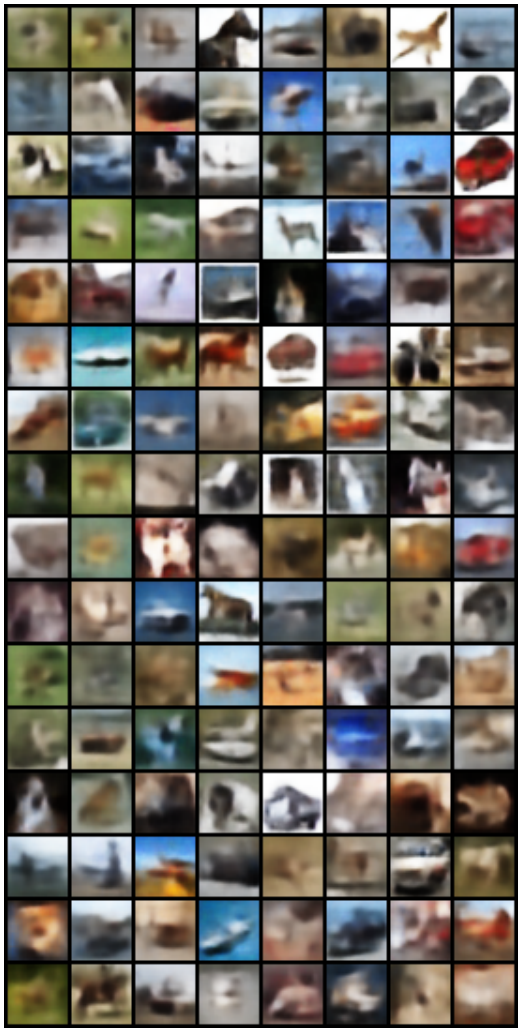

Figure 2: 128 Samples drawn from the VAE sub-model, trained on the CIFAR-10 dataset for 3,650 epochs. $batch - size = 128$, $latent - dim = 100$.

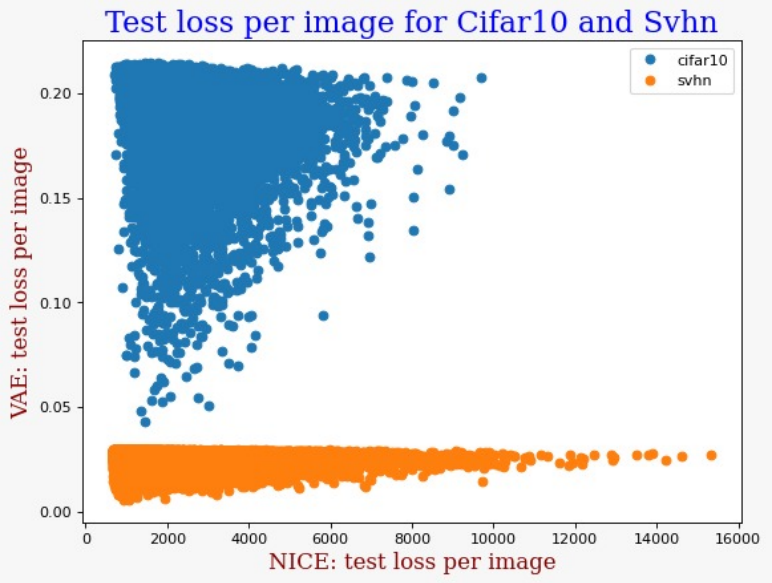


Figure 3: An example for the cleaned data (reconstruction loss and likelihood) for each input picture, divided by datasets.

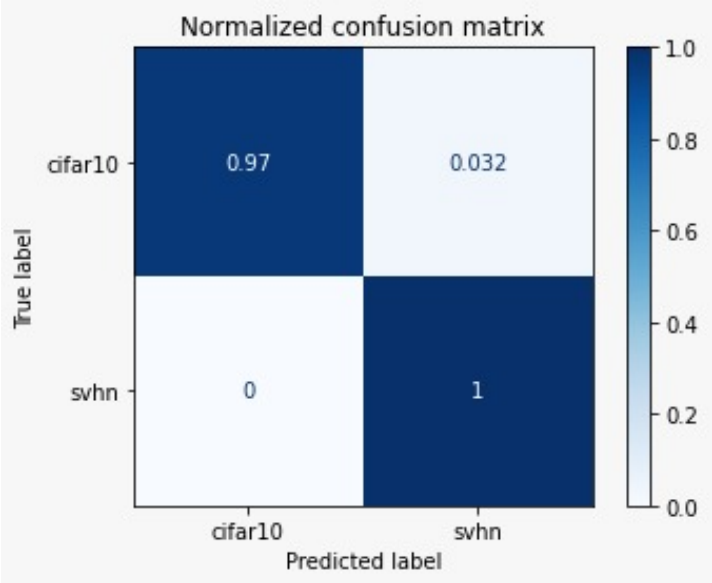


Figure 4: SVM confusion matrix on the test data, we can see great results.