Self-improving classification performance through GAN distillation

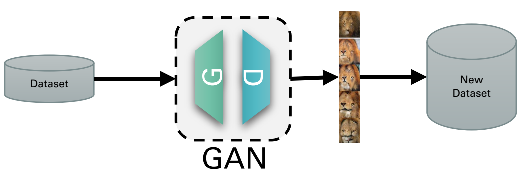
M. Pennisi, S. Palazzo, C. Spampinato

PeRCeiVe Lab, DIEEI, University of Catania, Italy

ILDAV
ICCV 2021 Workshop

Motivation

- The availability of a **large dataset** can be a key factor in achieving good generalization capabilities when training deep learning models.
- Unfortunately, **dataset collection is an expensive and time-consuming task**, especially in specific application domains (e.g., medicine).



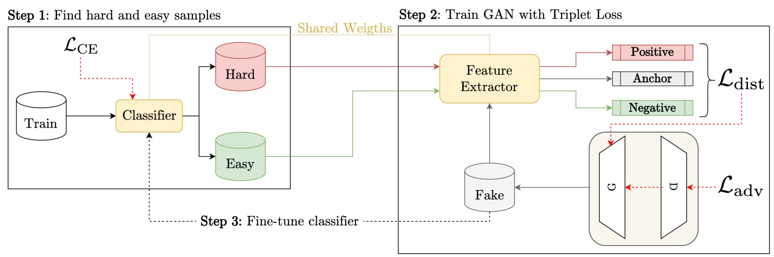
Existing augmentation approaches act just on the dataset before training.

Our idea

- As the final goal is to train a classifier it makes sense to include it in the augmentation process.
- We propose to leverage the training status of the classifier in order to **distill data that is more informative for the model.**

Method

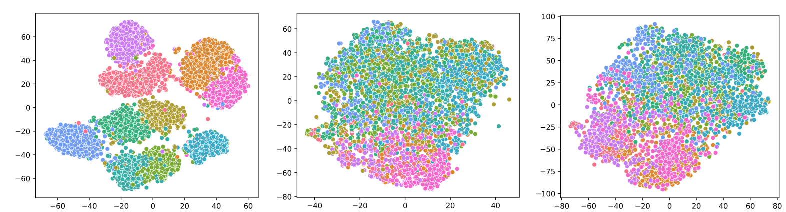
Framework Architecture



$$\mathcal{L}_{\text{dist}} = \mathbb{E}_{z,x_h,x_e}\left[\max(\|\mathbf{F}(\mathbf{G}(z)) - \mathbf{F}(x_p)\|_2 - \|\mathbf{F}(\mathbf{G}(z)) - \mathbf{F}(x_e)\|_2 + m, 0)\right]$$

1. Pre-train the classifier on the dataset, and label training data between **easy and hard samples.**
2. Pre-train the GAN using a **triplet loss** that encourages the model to generate realistic samples that match the feature distribution of hard samples.
3. Train both models simultaneously, fine-tuning the GAN to approximate the changing hard sample feature distribution while **training the classifier with a mixture of real and synthetic data.**

Feature Space of Classifier (t-SNE)



Easy Samples | Hard Samples | Synthetic Samples

Results

Results with different Classifiers:

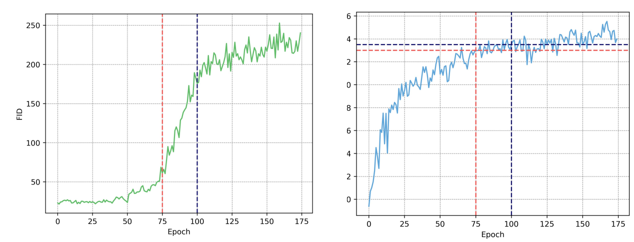
	AlexNet	ResNet-50	DenseNet-121
Baseline	69.63	71.85	79.41
GAN distil.	**74.56**	**77.48**	**81.50**
Gain	+ 4.93	+5.63	+2.09

GAN Distillation vs GAN augmentation:

	Accuracy	Accuracy Gain
Baseline	69.63	-
GAN augmentation	72.17	+2.54
GAN distillation	**74.56**	**+4.93**

Mode Collapse

- After the collapse of the Generator the Classifier continues to improve



- Collapsed images features lie on the borders of class clusters, effectively representing hard features.