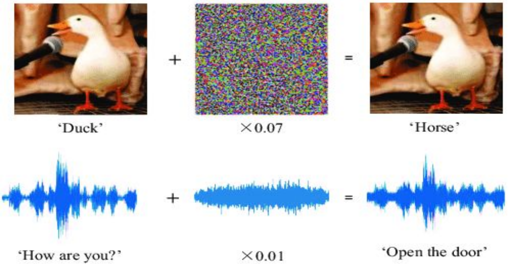
# Self-Supervised Learning with Adversarial Examples

**Student Name**: Şadi Uysal
**Advisor** : İnci Meliha Baytaş

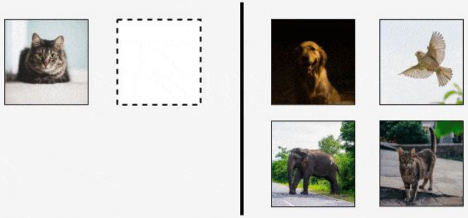
Depth.of Computer Engineering,
Bogazici University,TURKEY

## Introduction (Adversarial Examples,Contrastive Learning)



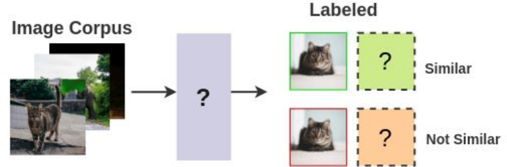
- Adversarial ex.(Duck is classified as Horse)

### Match the correct animal



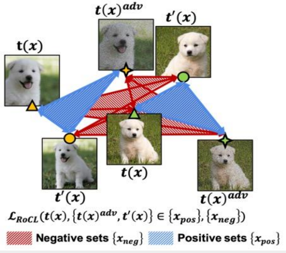
**Contrastive learning** *aims to learn the* common characteristics *of the dataset* ***without labels*** *via training and teaching the model which data points are similar or dissimilar.*

### How can we automatically generate pairs?



## Related Work

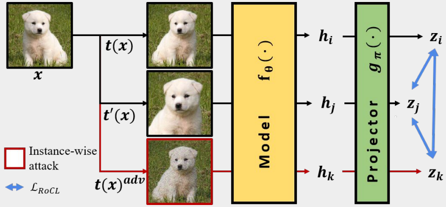
*Contrastive loss set creation method from* ***RoCL***


$t(x)^{adv}$
$t'(x)$
$t(x)$
$t(x)$
$t'(x)$
$t(x)^{adv}$
$\mathcal{L}_{RoCL}(t(x), \{t(x)^{adv}, t'(x)\} \in \{x_{pos}\}, \{x_{neg}\})$
Negative sets $\{x_{neg}\}$
Positive sets $\{x_{pos}\}$

Defined contrastive loss function where z, {zpos}, and {zneg} are corresponding latent vectors obtained by images t(x),t'(x) , x'. Cosine similarity between two vectors denoted by sim(u, v) operator, τ is a temperature parameter.

$$\mathcal{L}_{\text{con},\theta,x}(x, \{x_{\text{pos}}\}, \{x_{\text{neg}}\})$$

$$:= -\log \frac{\sum_{\{z_{\text{pos}}\}} \exp(\text{sim}(z, \{z_{\text{pos}}\})/\tau)}{\sum_{\{z_{pos}\}} \exp(\text{sim}(z, \{z_{\text{pos}}\})/\tau) + \sum_{\{z_{\text{neg}}\}} \exp(\text{sim}(z, \{z_{\text{neg}}\})/\tau)}$$

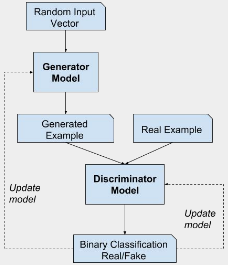


They integrated adversaries into the contrastive learning objective such that the objective seeks to minimize the distance between t(x),t'(x) , and generated adversary t(x)_adv in the latent space by maximizing the similarity between them.

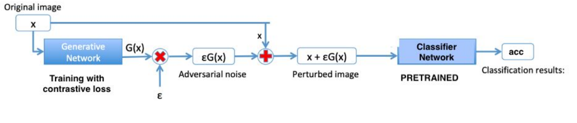
$$\mathcal{L}_{\text{RoCL},\theta,\pi} := \mathcal{L}_{\text{con},\theta,\pi}(t(x), \{t'(x), t(x)^{adv}\}, \{t(x)_{\text{neg}}\})$$

$$\mathcal{L}_{\text{total}} := \mathcal{L}_{\text{RoCL},\theta,\pi} + \lambda \mathcal{L}_{\text{con},\theta,\pi}(t(x)^{\text{adv}}, \{t'(x)\}, \{t(x)_{\text{neg}}\})$$

## Methodology

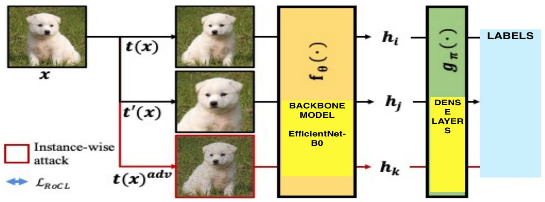


On the left, you can see the general architecture for the GANs. Generally, generator and discriminator networks are trained together. As you can see, both network updates its weights accordingly. But in our case, we used a pre-trained classifier network with fixed weights and used its layers as a backbone architecture for our representations.



Architecture diagram

In the GAN approach above, we generated adversarial noises using contrastive loss with the embeddings from backbone architecture, then created adversaries with adding noise to clean images, then we checked whether pre-trained classifier is fooled or not.



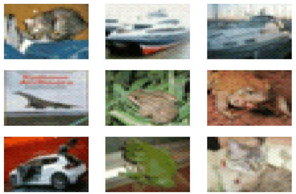
In our study, schema more or less similar to upper figure for our classifier.

## Methodology

We used Efficient-Net B0 architecture as a backbone model for our representations namely h vectors in the figure. We also added dense layers at the top of the backbone model for classifying into labels. After training the classifier model for CIFAR10 dataset, we freeze the classifier model.

## Results & Future Work

We trained our classifier before the generator training up to 88% accuracy on both training and test set. We were able to decrease the classifier's accuracy by 5 % with 10 epochs generator training.Some attacks below:



Because of technical difficulties, we were able to experiment small set of configurations but our study achieved promising results. With a better hyper-parameter optimization procedure better results can be achieved.Despite powerful attacks, using them for a robust network is another topic to study in future.