Multi-Agent Reinforcement Learning Benchmark on Highway Environment for Autonomous Driving

Université Paris-Saclay Machine Vision Project

Charbel Abi Hana
Department of Electrical Engineering
Université Paris-Saclay
Paris, France
charbel-a-h@outlook.com

***Abstract*—This is the abstract of the paper**
***Index Terms*—**

I. Introduction

hello this is a test in the paper [2], in this text i am going to cite another paper exactly here [1].

II. Related Work

III. Methods

A. Conditional Generative adversarial network(GANs)

$$\min_G \max_D \mathcal{V}_{\text{GAN}}(D, G) = \tag{1}$$

$$\mathbb{E}_{x \sim p_{\text{data}(x)}} \left[\log \{D(x)\}\right] + \mathbb{E}_{z \sim p_z(z)} \left[\log \{1 - D(G(z))\}\right].$$

where $G : R^{100} \longrightarrow R^{16,384}$

$$L_D = -\sum_{x \in \chi, z \in \zeta} \log(D(x)) + \log(1 - D(G(z))) \tag{6}$$

$$L_G = -\sum_{z \in \zeta} \log(D(G(z)) \tag{7}$$

B. Model Architecture

1) Generator:

$$h^{[i]} = LeakyRELU(W^{[i-1]} h^{[i-1]} + b[i-1]) \tag{2}$$

α, with h[i] $\epsilon R^{16\alpha 2^i}$ and we output the vector o $\epsilon R^{16,384}$ via

$$o = \tanh(W^{[L]} h^{[L]} + b[L]) \tag{3}$$

Where L is the final layer.

2) Discriminator: $h[0]$ denote the input image, $W[j]$ and $b[j]$ denoting the weight matrix and the bias vector in the L output layer, we have:

$$o = sigmoid(W^{[L]} h^{[L]} + b[L]) \tag{4}$$

3) Classifier:

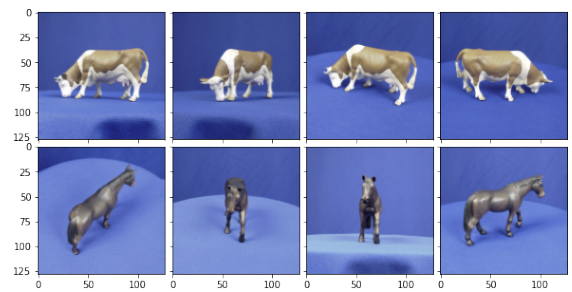

Fig. 1. Random Cows and Horses Images from Original Dataset

IV. Experiments

A. Dataset

B. Evaluation Metrics

C. Experimentation Details

D. GAN Experimentation Results

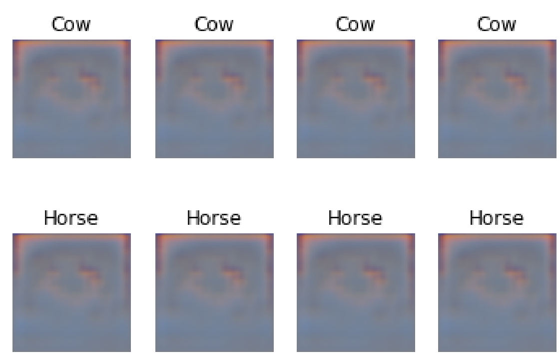

Fig. 2. Generated Output after 100 Steps

E. CycleGAN Experimentation Results

- Pass real images through the generators and get the generated images

Identify applicable funding agency here. If none, delete this.

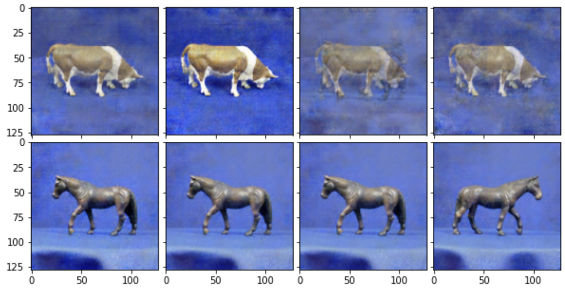

Fig. 3. Generated Output after 35000 Steps

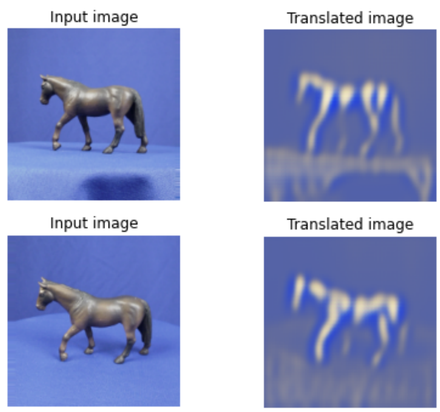

Fig. 4. Generated Translated Output after 1 Epoch

- Pass the generated images back to the generators to check if we can predict the original image from the generated image.
- Do an identity mapping of the real images using the generators.
- Pass the generated images in 1) to the corresponding discriminators.
- Calculate the generators total loss (adverserial + cycle + identity)
- Calculate the discriminators loss
- Update the weights of the generators
- Update the weights of the discriminators
- Return the losses in a dictionary.

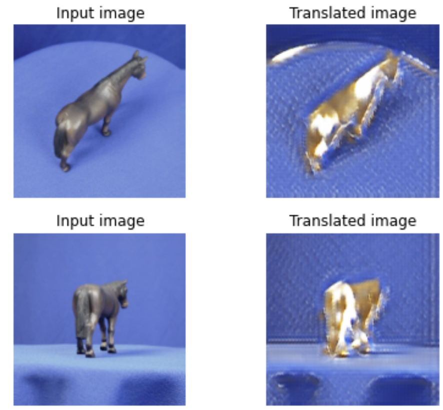

Fig. 5. Generated Output after 50 Epochs

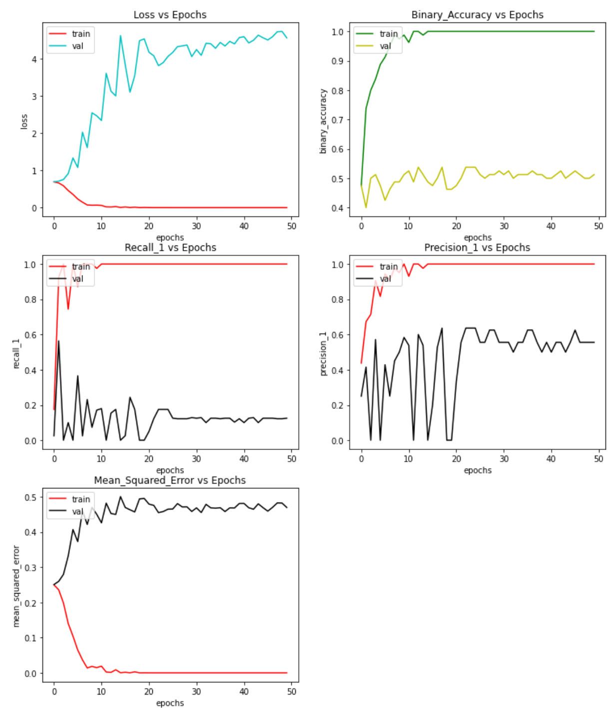

Fig. 6. Classification metrics for classifier trained on **original dataset**

F. Classifier Experimentation Results

Metric	Orig		Orig-250-GAN		Orig-500-GAN		Orig-1000-GAN	
-	**train**	**val**	**train**	**val**	**train**	**val**	**train**	**val**
BCE-Loss	0.01	5	0.01	0.7	0.01	0.7	8	0.7
Accuracy	1.0	0.5	1.0	0.6	1.0	0.5	0.5	0.5
Precision	1.0	0.1	1.0	0.5	1.0	0.5	0.5	0.5
Recall	1.0	0.5	1.0	0.4	1.0	1.0	1.0	1.0

TABLE I
COMPILED TABLE OF CLASSIFICATION METRICS

V. CONCLUSION

REFERENCES

[1] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

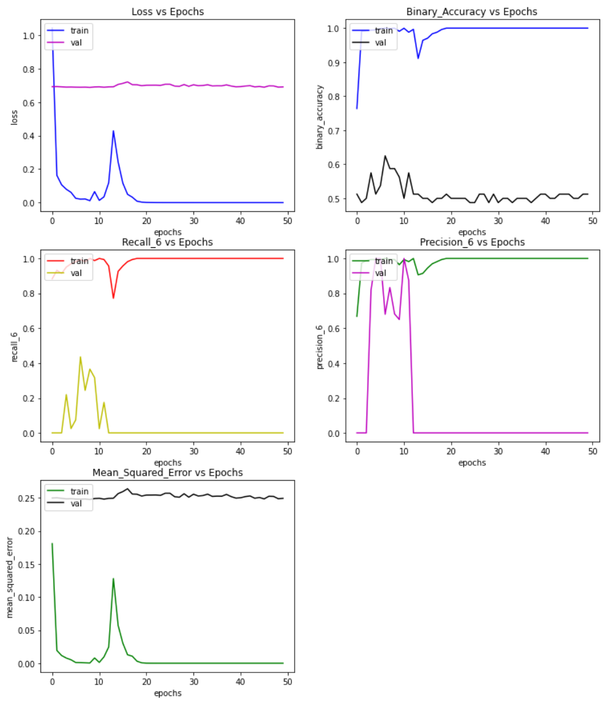

Fig. 7. Classification metrics for classifier trained on **original+250 GAN-based**

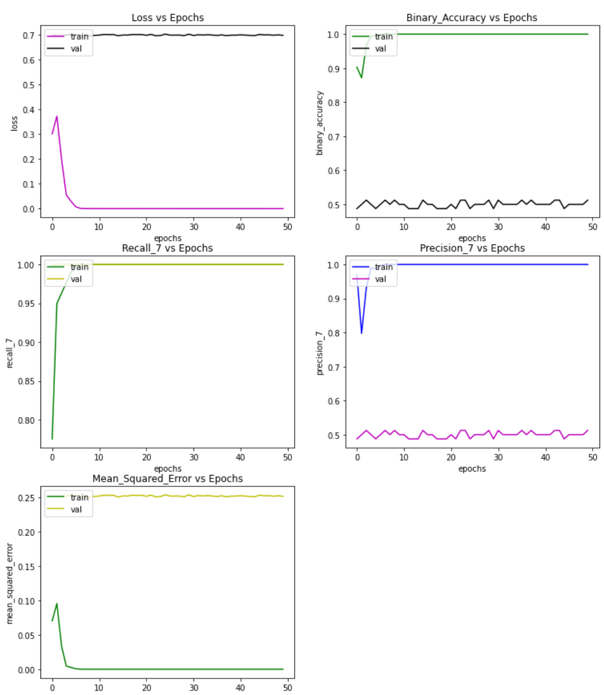

Fig. 8. Classification metrics for classifier trained on **original+250 GAN-based**

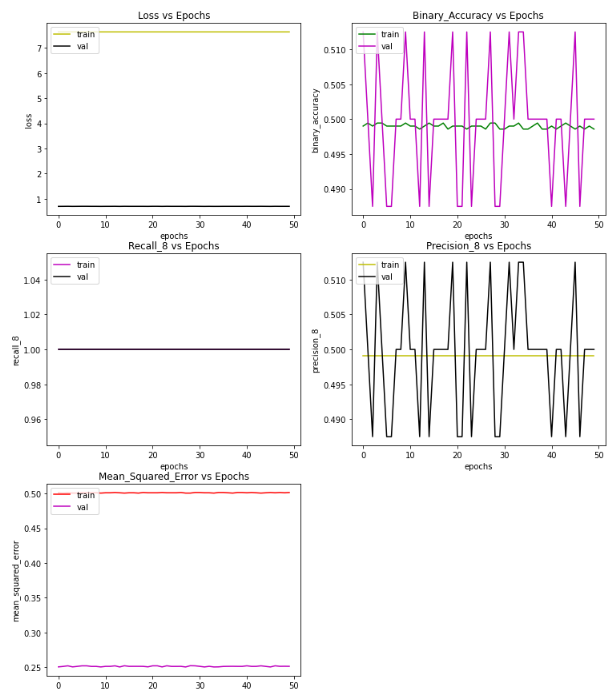

Fig. 9. Classification metrics for classifier trained on **original+250 GAN-based**