Politecnico di Torino

AML & DAAI

# EXERCISE 3
# DOMAIN ADAPTATION

Teaching Assistant:

**Leonardo Iurada**
leonardo.iurada@polito.it

# OVERVIEW

1. Description of the problem: **Domain Shift**
2. Description of the solution: **Domain Adaptation** through **DANN/GRL**
3. Dataset: **PACS**
4. Tools: **Google Colab**
5. Required steps:
    a. Implement DANN/GRL architecture
    b. Train on Cartoon, test on Sketch **without** adaptation
    c. Train on Cartoon, test on Sketch **with** adaptation (using DANN/GRL)
    d. Change the values of some hyperparameters

# DOMAIN SHIFT

## Training Set (Source Domain)

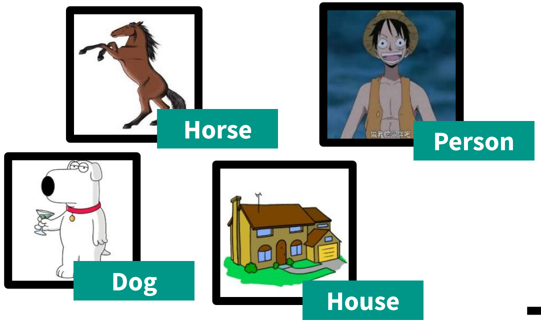
GTA 5 (Synthetic)

## Test Set (Target Domain)

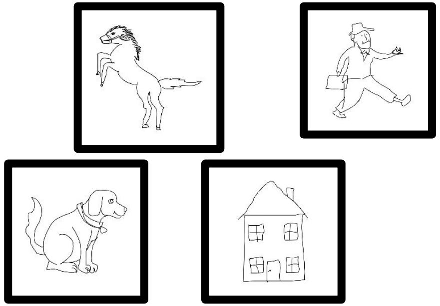
Cityscapes (Real)

# UNSUPERVISED DOMAIN ADAPTATION (UDA)

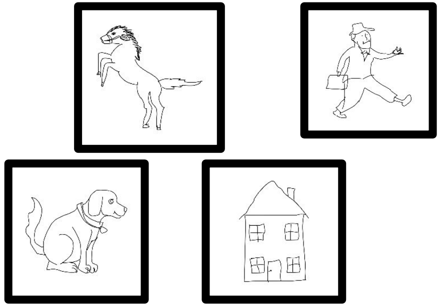
## Training Set (Cartoon) – With Labels



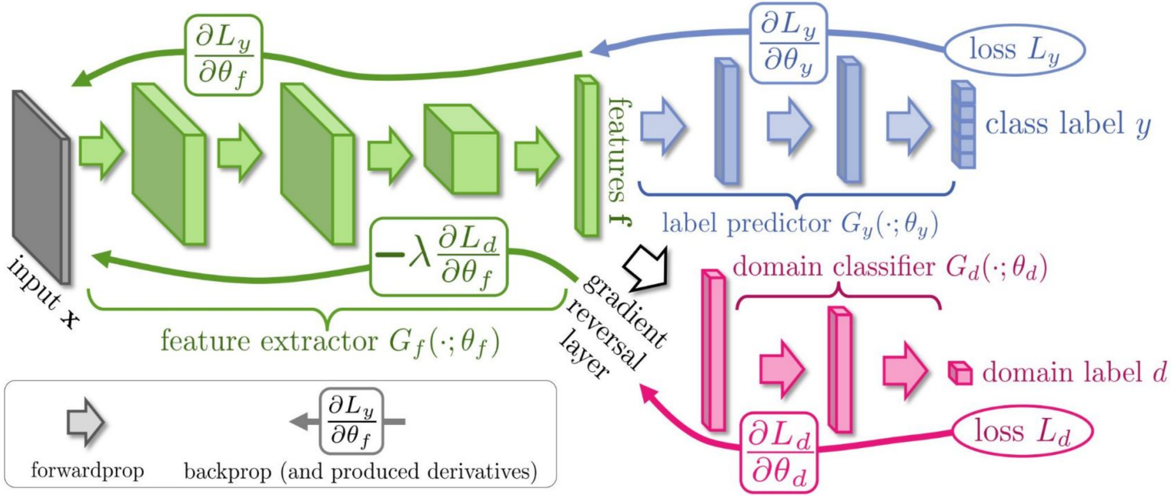

## Training Set (Sketch) – No Labels



## Test Set (Sketch)



# DOMAIN-ADVERSARIAL ADAPTATION (DANN/GRL)




We will note the prediction loss and the domain loss respectively by

$$
\begin{aligned}
\mathcal{L}_y^i(\theta_f, \theta_y) &= \mathcal{L}_y\big(G_y(G_f(\mathbf{x}_i;\theta_f);\theta_y), y_i\big)\,,\\
\mathcal{L}_d^i(\theta_f, \theta_d) &= \mathcal{L}_d\big(G_d(G_f(\mathbf{x}_i;\theta_f);\theta_d), d_i\big)\,.
\end{aligned}
$$

Training DANN then parallels the single layer case and consists in optimizing

$$
E(\theta_f, \theta_y, \theta_d) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y^i(\theta_f, \theta_y) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_d^i(\theta_f, \theta_d) + \frac{1}{n'}\sum_{i=n+1}^{N}\mathcal{L}_d^i(\theta_f, \theta_d)\right)
$$

Ganin et al. “Domain-Adversarial Training of Neural Networks”. The Journal of Machine Learning Research, 17(1), 2096-2030, 2016.

# PACS

PACS is an image classification dataset:

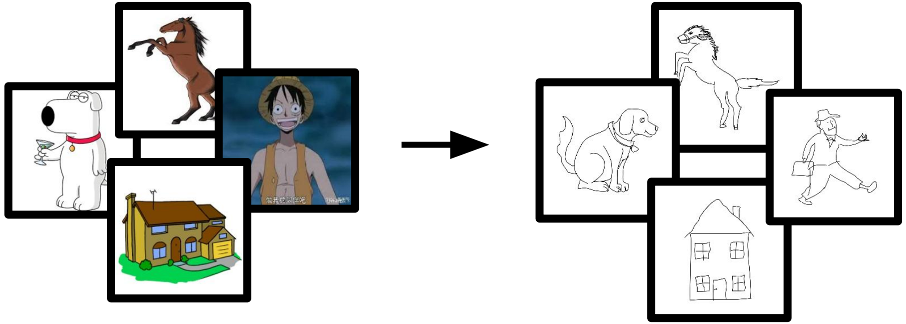
- **7 object categories**

  (Dog, Elephant, Giraffe, Guitar, Horse, House, Person)

- **4 visual domains**

  (**P**hoto, **A**rt Painting, **C**artoon, **S**ketch)


**Source Domain (Cartoon)** **Target Domain (Sketch)**



Li et al. “Deeper, Broader and Artier Domain Generalization.” 2017 IEEE International Conference on Computer Vision. ICCV, 2017.

# SETUP THE ENVIRONMENT

- You can find the starter Google Colab Notebook here
  (or at https://t.ly/oEgEU)

- Click "Copy on Drive" to clone the notebook on your Google Drive

- Now you should be able to connect to a GPU Runtime and edit the code
  ("Runtime" dropdown menu → "Change runtime type" → Select "T4 GPU" and click "Save")

# a. IMPLEMENT DANN/GRL ARCHITECTURE

Starting from the initial code, modify the architecture:

- Modify the **`__init__` function** of the **`AlexNet`** class to add a new classifier that performs domain discrimination

- Initialize the **weights of AlexNet pre-trained on ImageNet**

- Extend the **forward pass & the loss** calculation:
    - **Source** domain data (**labeled**) passes through the **class discriminator:**

    $$\mathcal{L}_y^i(\theta_f, \theta_y) \quad = \quad \mathcal{L}_y\big(G_y(G_f(\mathbf{x}_i; \theta_f); \theta_y), y_i\big)$$

    - **Source** domain data passes through the **domain discriminator:**

    $$\mathcal{L}_d^i(\theta_f, \theta_d) \quad = \quad \mathcal{L}_d\big(G_d(G_f(\mathbf{x}_i; \theta_f); \theta_d), d_i\big)$$

    - **Target** domain data passes through the **domain discriminator:**

    $$\mathcal{L}_d^i(\theta_f, \theta_d) \quad = \quad \mathcal{L}_d\big(G_d(G_f(\mathbf{x}_i; \theta_f); \theta_d), d_i\big)$$

# b. TRAIN WITHOUT ADAPTATION

For this exercise you have to consider as:

- **Source Domain**: Cartoon
- **Target Domain**: Sketch

- Implement the `Dataset` class, build the **Dataset** and **DataLoader objects**

- Setup the **optimizer (SGD)** and the **learning rate scheduler (StepLR)**

- Write the **training loop** and the **test loop**

**On the Target Domain, you should obtain an accuracy of ~50% at test time**

# C. TRAIN WITH ADAPTATION

For this exercise, again, you have to consider as:

- **Source Domain**: <u>Cartoon</u>
- **Target Domain**: <u>Sketch</u>

- Write a new **training loop** that uses the Domain Classifier with the Gradient Reversal Layer (DANN/GRL architecture)
  [**HINT 1:** you can zip train & test dataloaders together, when iterating over the data]
  [**HINT 2:** source examples have domain label “0”, while target examples have domain label “1”]

**On the Target Domain, you should obtain an increase in the accuracy at test time (+2/3%)**

# d. CHANGE THE VALUE OF SOME HPs

Analyze the behavior of the training, by changing the value of some hyperparameters (e.g. **Lambda** or **LR**) – You can perform a **grid search** of their values

Optionally, you may perform at least **3 runs** with each set of hyperparameters and report in a spreadsheet (e.g. it is suggested to use Google Sheets) the **mean** and **standard deviation** of the runs

| LR | $\lambda$ | CARTOON_ACC | SKETCH_ACC | AVG_ACC |
|---|---|---|---|---|
| 0.1 | None | 0.163 | 0.191 | 0.177 |
| 0.01 | None | 0.313 | 0.310 | 0.312 |
| 0.005 | None | 0.262 | 0.261 | 0.261 |
| 0.001 | None | 0.250 | 0.199 | 0.224 |
| 0.1 | 0.5 | 0.163 | 0.191 | 0.177 |
| 0.01 | 0.5 | 0.162 | 0.191 | 0.176 |
| 0.005 | 0.5 | 0.165 | 0.192 | 0.178 |
| 0.001 | 0.5 | 0.162 | 0.192 | 0.177 |
| 0.1 | 0.1 | 0.163 | 0.192 | 0.177 |
| 0.01 | 0.1 | 0.520 | 0.191 | 0.356 |
| 0.005 | 0.1 | 0.468 | 0.191 | 0.330 |
| 0.001 | 0.1 | 0.277 | 0.198 | 0.237 |
| 0.1 | 0.05 | 0.164 | 0.191 | 0.177 |
| 0.01 | 0.05 | 0.507 | 0.171 | 0.339 |
| 0.005 | 0.05 | 0.339 | 0.583 | 0.461 |
| 0.001 | 0.05 | 0.253 | 0.362 | 0.308 |