Deep Optimal Transport for Domain Adaptation Application to Vehicle Re-Identification

Yujin CHO, Israfel SALAZAR

M1 E3A @ Université Paris-Saclay. Paris, France

Introduction

Vehicle re-Identification is the task of finding the queried vehicle from the non-overlapping cameras, and the goal of attribute recognition is to predict the presence of a set of attributes from an image. It is part of smart city concept that aims to improve the quality of life of city dwellers by making the city more adaptive and efficient, using new technologies that are based on an ecosystem of objects and services.

Define the Problem

For each vehicle, there are big intra-class difference and learning all class distributions would need a huge amount of data. We proposed to train a model using only the front and rear view of vehicles, our hypotheses was that using domain adaptation we would be able to obtain good results in the vehicle re-identification problem for different distributed datasets.

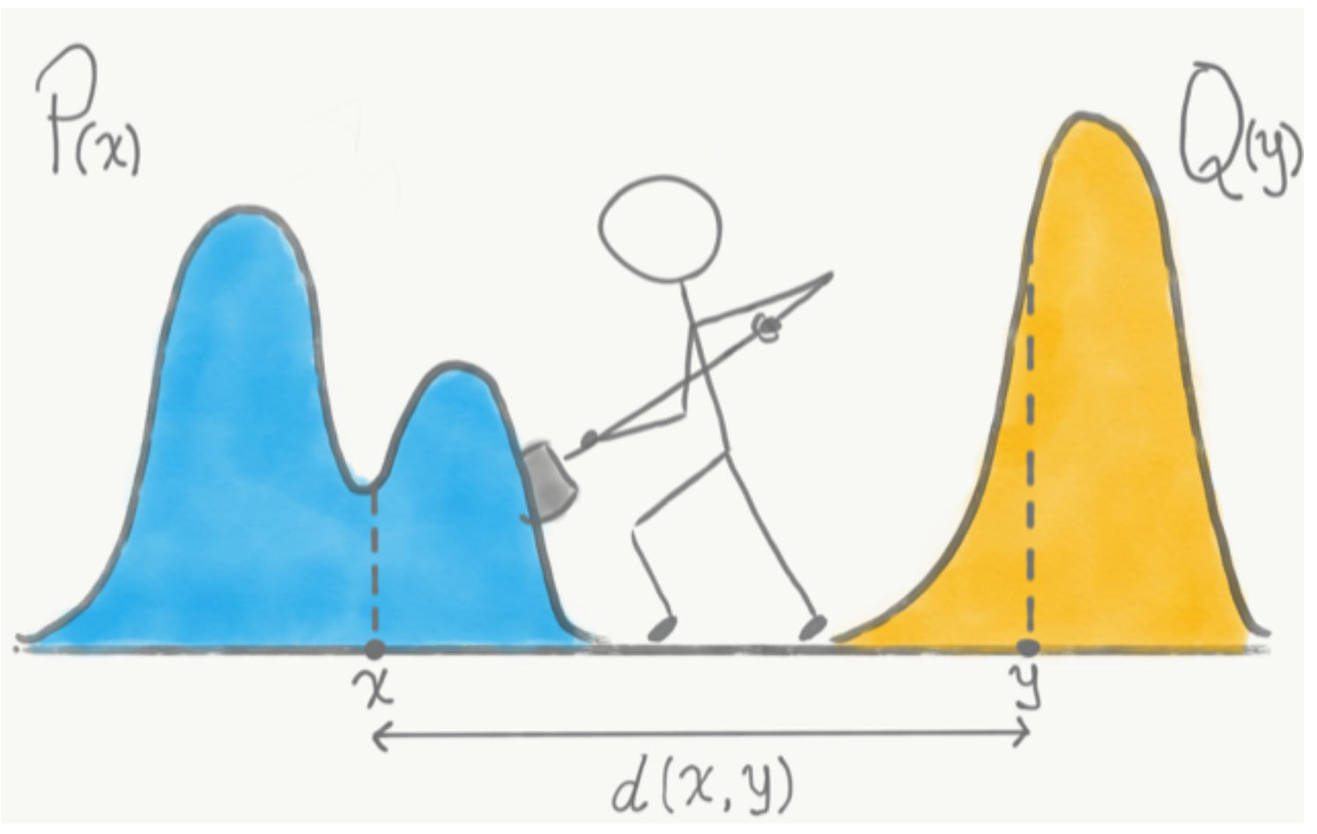
Objectives

With a limited amount of data using a dataset created using few cameras (view of vehicles), implement the domain adaptation algorithm using optimal transport to improve accuracy of model in unseen and with different distribution cameras).

What is optimal transport?

Gaspard Monge

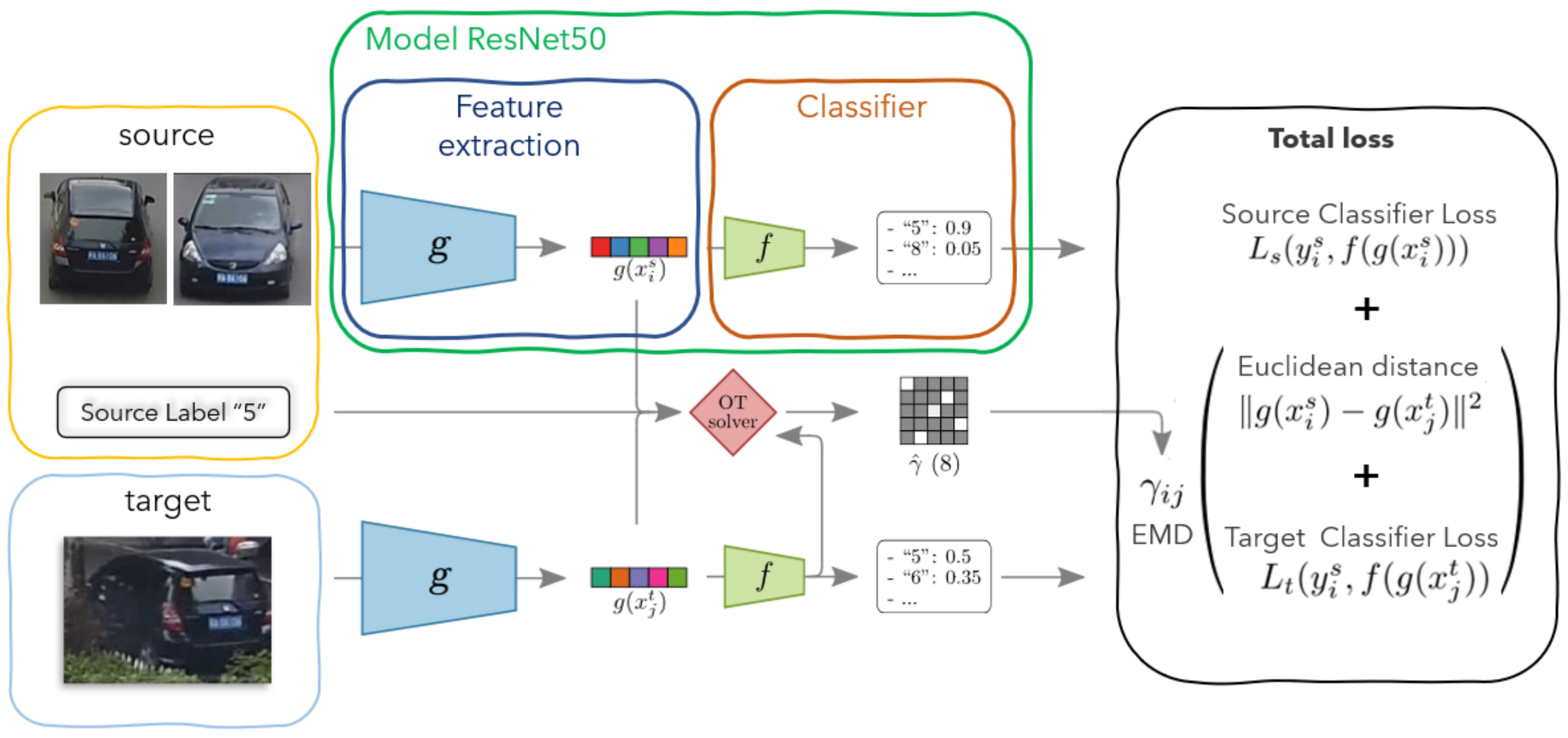
Optimal Transport problem allows us to compare probability distributions in a geometrically manner. We search for the most economical way, for example in distance, to transport the distribution of a set of data to another using the geometry of the feature space.



Earth mover distance

The earth mover's distance (EMD) is a measure of the distance between two probability distributions over a region D (Also known as the Wasserstein metric). The EMD is the minimum cost to move from distribution P(x) to Q(y)

DeepJDOT Implementation



DeepJDOT architecture

DeepJDOT uses a CNN model to solve the problem of different distribution between source and target. We used a ResNet50 model (pre-trained on ImageNet) divided in 2 parts (g = feature extractor and f = classifier).

Algorithm 1 DeepJDOT stochastic optimization

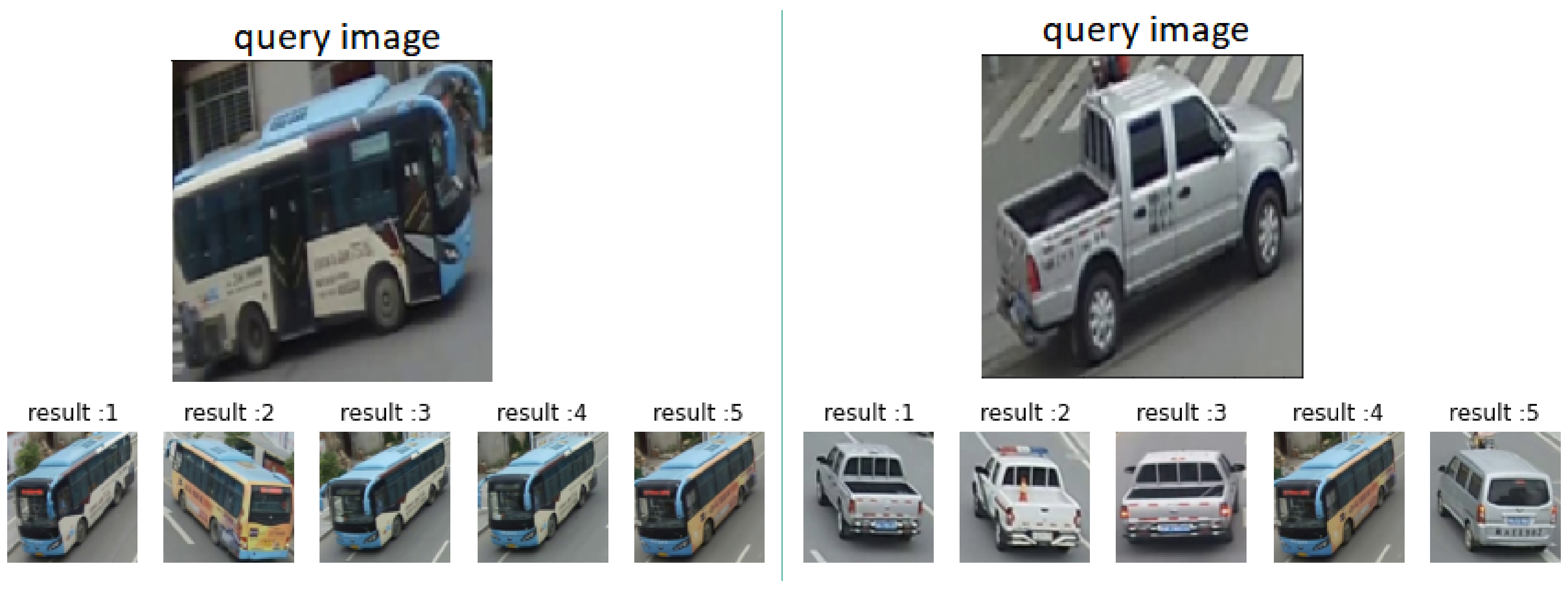
```
Require: $\mathbf{x}^s$: source domain samples, $\mathbf{x}^t$: target domain samples, $\mathbf{y}^s$: source domain labels
for each batch of source $(\mathbf{x_b}^s, \mathbf{y_b}^s)$ and target samples $(\mathbf{x_b}^t)$ do
    fix $\hat{g}$ and $\hat{f}$, solve for $\gamma$ as in equation (8)
    fix $\hat{\gamma}$, and update for $g$ and $f$ according to equation (9)
end for
```

DeepJDOT algorithm

Cross Entropy loss is used to back-propagate classifier estimations. The cost function γ is computed between the features of source and target, then it's minimized by using EMD. The total loss is then computed and we update the models by backpropagation.

Expected results

Model trained with DeepJDOT method should perform better on target data than model trained only on source data



Top5 result closest result in source dataset

Difficulties of implementation

- Dataset management to split source and training datas
- DeepJDOT training stability : best accuracy 25%
- Implementation of EMD (Optimal transport solver)
- Hyperparamters Searching and Fine tuning