Computer Vision Report

Team Members -

1. Samyak Jain - 20161083
2. Anvesh Chaturvedi - 20161094
3. Yudhik Agrawal - 20161093

TOC

Abstract

The paper proposes an algorithm that learns representations which explicitly compensate for domain mismatch and which can be efficiently realized as linear classifiers. The goal is to form a linear transformation that maps features from the target (test) domain to the source (training) domain as part of training the classifier. Transformation and classifier parameters are jointly optimized and an efficient cost function is introduced based on misclassification loss. The method combines several features previously unavailable in a single algorithm: multi-class adaptation through representation learning, ability to map across heterogeneous feature spaces, and scalability to large datasets.

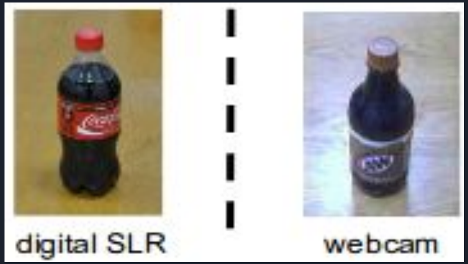
Understanding the problems

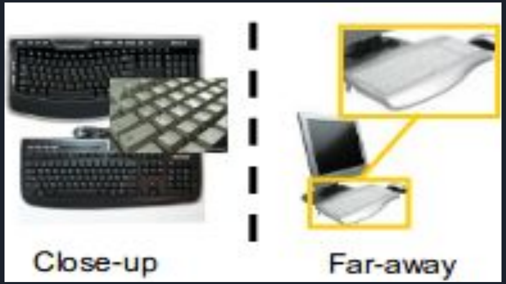
01 Input feature distributions change due to different image sensors and noise conditions , pose changes , a shift from commercial to consumer video.

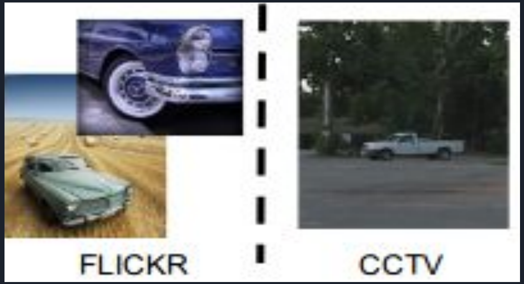
02 Training datasets are biased by the way in which they were collected.

03 This paper proposes a solution to the problem of learning domain-invariant image representations for multi-class classifiers, so that this can be overcome.

Variations in Image in different domains.







Datasets Used

- Office
  - Object recognition
  - sample - 4110
- Caltech256
  - Object recognition
  - 2553
- Bing
  - Object recognition

Optimization Problem

The optimization problem proposed is convex is nature and is similar to soft margin linear SVM with some additional penalties. The objective function is as follows where C_S penalizes the source classification error and C_T penalizes the target adaptation error. Our goal is to minimize the objective function.

$$
\begin{aligned}
J(W, \theta_k, b_k) \quad = \quad & \frac{1}{2}||W||_F^2 + \sum_{k=1}^{K} \left[\frac{1}{2}||\theta_k||_2^2 \right. \\
& \left. + C_S \sum_{i=1}^{n_S} \mathcal{L}\left(y_i^s, \begin{bmatrix} x_i^s \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}\right) + C_T \sum_{i=1}^{n_T} \mathcal{L}\left(y_i^t, W \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix}\right) \right]
\end{aligned}
$$

where

$$\mathcal{L}(y, x, \theta) = \max\{0, 1 - \delta(y, k) \cdot x^T\theta\}$$

Algorithm

- Set iteration $j = 0$, $W^{(j)} = 0$.
- Solve the sub-problem $(\theta_k^{(j+1)}, b_k^{(j+1)}) = \arg\min_{\theta k, bk} J(W^{(j+1)}, \theta_k, b_k)$ by solving:

$$\min_{\theta, b} \sum_{k=1}^{K} \left[\frac{1}{2} ||\theta_k||_2^2 + C_S \sum_{i=1}^{n_S} \mathcal{L}\left(y_i^s, \begin{bmatrix} x_i^s \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k; \\ b_k \end{bmatrix} \right) + C_T \sum_{i=1}^{n_T} \mathcal{L}\left(y_i^t, W^{(j)} \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k \\ b_k \end{bmatrix} \right) \right]$$

- Solve the subproblem $W^{(j+1)} = \arg\min_W J(W, \theta^{(j+1)}, b^{(j+1)})$ by solving:

$$\min_{W} \quad \frac{1}{2} ||W||_F^2 + C_T \sum_{k=1}^{K} \sum_{i=1}^{n_T} \mathcal{L}\left(y_i^t, W \cdot \begin{bmatrix} x_i^t \\ 1 \end{bmatrix}, \begin{bmatrix} \theta_k^{(j+1)} \\ b_k^{(j+1)} \end{bmatrix} \right)$$

The steps of the proposed iterative algorithm can be solved using standard QP solvers, the algorithm can be easily implemented. Additionally, since the constraints in the algorithm grow linearly with the number of training points, it can be solved in linear feature space. The optimization can be solved efficiently even as the number of training points grows.

Project timeline

11 Feb — First Milestone

Complete understanding of the problem.

20 Feb — Starting the project.

Start working on the implementation of the MMDT algorithm being proposed.

1 Mar — Second Milestone

Exploring the dataset and start implementing the research paper.

10 Mar — Training the model

Hyperparameter training on the validation set. Modifying the work based on review in the second milestone.

20 Mar — Extensive Testing

Exploring multiple images and comparing the results with state of the art algorithms.

31 Mar — Final Presentation

Buffer period for completion of work and preparing the final presentation.