Traditional machine learning methods assume that the training data (labeled) and testing data (unlabeled) are from same feature space and following similar distributions. This assumption may not be true in real applications.

The goal of cross-domain recognition is to train a classifier from one domain and use it for another domain.

Our proposed method includes five steps:

1. **Constructing subspaces in source domain.** Constructing the subspaces in source domain $M_S^x = \{X_i^x \}_{i \in \mathcal{X}^x} : \mathcal{X}^x \subset \mathcal{Y}$, where $\mathcal{X}^x \subset \mathcal{Y}$.

2. **Constructing anchor subspaces in target domain.** Constructing $K$ anchor subspaces $\{M_T^y\}_{y \in \mathcal{Y}}$ by grouping target data samples with high similarities, to ensure that each anchor subspace only contains data from a single class.

3. **Labeling the anchor subspaces.** To assign a label for each anchor subspace, we define a cost function that considers:
   1) Cross-domain distance;
   2) Within-domain topological relation.

4. **Constructing compact joint subspaces.** Constructing compact joint subspace $M_{JS} = \{M_S^x, M_T^y\}_{(x, y) \in \mathcal{X} \times \mathcal{Y}}$.

5. **Training classifiers on the compact joint subspaces.** Training one-vs.-rest SVM classifiers for each class using in the compact joint subspace.

Our main assumption is that the data from the same class should lay on a low-dimensional subspace, even if they come from different domains. Problem: $\{X_i^x\}_{i \in \mathcal{X}^x} \in \mathbb{R}^{d \times n_x}$ and $\{X_i^y\}_{i \in \mathcal{X}^y} \in \mathbb{R}^{d \times n_y}$ denote the data from source and target domains. $\mathcal{Y} = \{Y_i\}_{i \in \mathcal{Y}} \in \mathbb{R}^{1 \times n_y}$ are the labels of all source data. Our aim is to estimate $\mathcal{Y} \in \mathbb{R}^{1 \times n_y}$, the labels of all the data in the target domain.

The overview of our approach is shown in Figure 1.