Multi-Domain Learning

Vasu Sharma
LTI, CMU

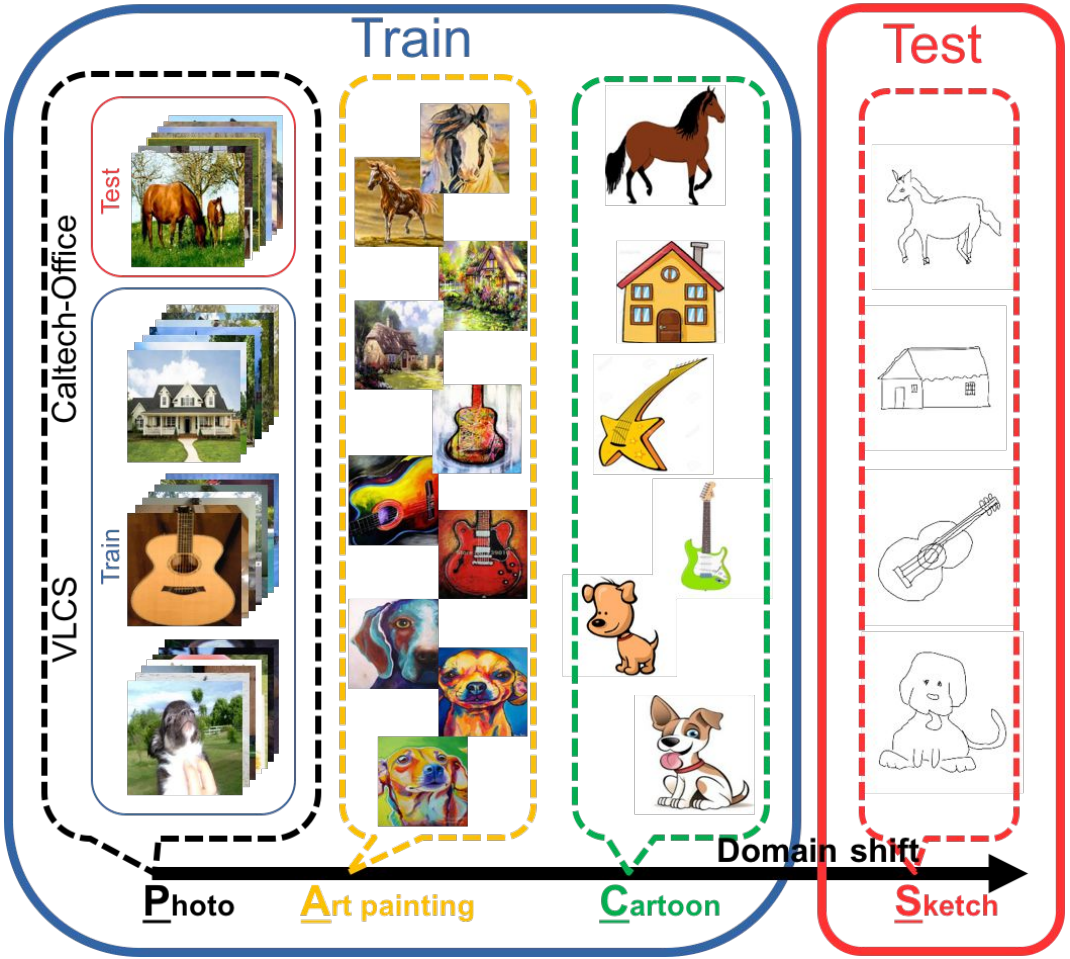
What is Multi-Domain Learning?

Problems with present ML models

- Most ML models work on a specific data domain
- Don’t generalize very well to new domains
- Needs to be re-trained for a different dataset

How does Multi-Domain Learning help?

- Multi Domain Learning allows for learning with data from multiple different domains
- Retains the advantages of targetted training on a single domain


Train
Test
Caltech-Office
VLCS
Test
Train
Domain shift
Photo
Art painting
Cartoon
Sketch

Visual Tracking

- Problem of tracking an object over time in a video
- Finds use in video surveillance, activity recognition, medical imaging etc
- Often handled as an object detection problem
- Most prior approaches solve it using object detection at a per frame level
- Typical challenges
  - Occlusion
  - Small object sizes
  - Fast moving objects

Branch-Activated Multi-Domain Convolutional Neural Network for Visual Tracking (BAMDCNN)

Key ideas and concepts:

- Propose a Multi Branch architecture with dedicated generic and domain specific branches
- Use similarity based grouping and online clustering to identify multiple domains without labelled domain information
- Train the domain specific branches per domain to retain the ‘targetted’ learning ability of model

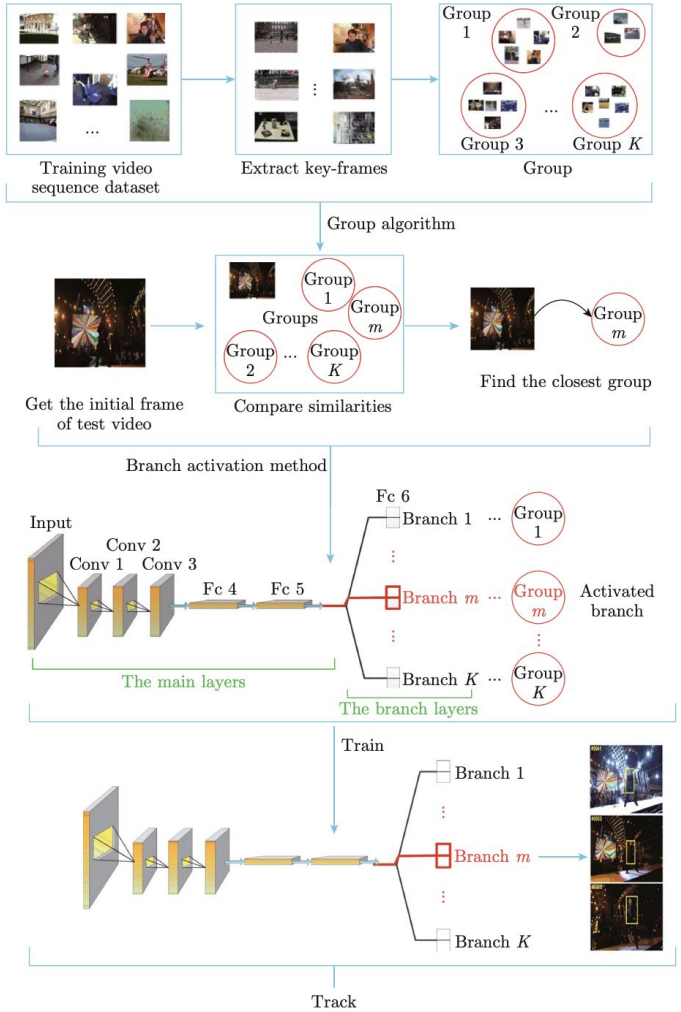


Fig. 1 Flow chart of BAMDCNN

Model Architecture: Key Components

The BAMDCNN model has 4 major components

- Generic Image Feature Extractor
- Key-Frame Extraction
- Group Algorithm Based on Similarity
- Branch Activation Method

Generic Image feature extractor

- Allows model to extract domain independent image features
- Uses the conv layers of the relatively shallower VGG-M network
- Smaller input size, hence shallower network enough
- Smaller receptive field allows focussing on smaller objects and ignore clutter
- Focus on finer features rather than global one

Key-Frame Extraction

- Extracts key-frames representative of the overall domain of the video
- Computations with key frames cheaper than entire video
- Based on mutual comparisons and unsupervised clustering

Steps for Key-Frame Extraction

1. Mean and Variance based image Normalization, performed as:

$$P_{i,j} = \frac{p_{i,j} - \bar{p}}{\sigma},$$

2. Each frame assigned to nearest cluster center measured as:

$$\rho_{\mathrm{c}} = \min \rho(f_i, c_j),$$
$$j = 1, 2, \cdots, n_{\mathrm{c}},$$

3. If the minimum distance to the cluster center is greater than threshold then create a new cluster with this frame as centroid
4. Clusters with > 10% of total frames chosen as ‘effective clusters’
5. Frames closest to cluster centroid of ‘effective clusters’ chosen as key-frames
6. Duplicated key-frame removal if similarity between keyframes from same video greater than a threshold

Group Algorithm Based on Similarity

- This step is used to determine the cluster membership of a video
- Consider 2 videos $q_i = (f_1, f_2 \ldots f_d)$ and $q_j = (f'_1, f'_2, \ldots, f'_m)$ where f_i and f'_j represent the keyframes in the video then distance between the 2 videos is:

$$S_{i,j} = \frac{1}{dm} \sum_{t=1}^{d} \sum_{r=1}^{m} \rho(f_t, f'_r),$$

- Now the distance of a new video q_m from the cluster c can be computed as:

$$D_{g_c} = \frac{1}{|g_c|} \sum_{i=1}^{|g_c|} S_{m,i},$$

$$\hat{c} = \arg\min_{c} D_{g_c},$$

$$c = 1, 2, \cdots, n_g,$$

where $|g_c|$ denotes number of videos in cluster c and $\hat{c}$ denotes the cluster closest to the video q_m

Group Algorithm Based on Similarity (contd.)

- Assignment cases:
  - $D_{g\hat{c}} < \theta$ *and* $|g_{\hat{c}}| < \boldsymbol{\tau}$: q_m is assigned to ĉ
  - $D_{g\hat{c}} \geq \theta$ *or* $|g_{\hat{c}}| \geq \boldsymbol{\tau}$ *:* q_m is assigned to a new cluster and made it's centroid

- The threshold θ controls the maximum distance between videos in a cluster and centroid. This ensures clusters are well knit

- $\boldsymbol{\tau}$ controls the maximum membership of a cluster. This ensures that the clusters don't become too large and different groups dont combine into a larger group

Branch Activation Method

- Used at inference time to decide which network branch to use
- Uses initial video frame f_{in} to compute distance from all clusters as:

$$S_{q_i}^{(1)} = \frac{1}{|q_i|} \sum_{j=1}^{|q_i|} \rho(f_{\text{in}}, f_j)$$

  Here S_{qi} denotes distance of the video from cluster q_i
- The video is assigned to the cluster ĉ as:

$$D_{g_c}^{(1)} = \frac{1}{|g_c|} \sum_{i=1}^{|g_c|} S_{q_i}^{(1)},$$

$$\hat{c} = \arg\min_{c} D_{g_c}^{(1)},$$

$$c = 1, 2, \cdots, n_{\text{g}}.$$

- The branch corresponding to cluster ĉ gets activated for this video and predicted the tracked bounding box co-ordinates

Experiments

- Perform One-Pass evaluation (OPE) and Spatial Robustness Evaluation (SRE) where frames are translated and scaled
- Distortions added as:
  - fast motion (FM)
  - motion blur (MB)
  - deformation (Def)
  - low resolution (LR)
  - occlusion (Occ)
  - out-of-plane rotation (OPR)
  - out of view (OV)
  - in-plane-rotation (IPR)
  - illumination variation (IV)
  - background clutter (BC)
  - scale variation (SV)

Experiments (contd.)

- Comparison against 8 state of the art approaches:
  - Multi-Domain CNN (MDCNN)
  - Constructing adaptive complex cells tracker (CACCT)
  - Structured output tracker (SOT)
  - Locality sensitive histogram tracker (LSHT)
  - Adaptive structural local sparse appearance model (ASLSAM)
  - Sparsity-based collaborative model (SCM)
  - Tracking-learning-detection method (TLDM)

Results

Table 1 Average precision scores on individual attributes

Attribute	Average precision							
	CACCT	SOT	LSHT	ASLSAM	SCM	TLDM	MDCNN	BAMDCNN
Occ	0.651	0.562	0.581	0.466	0.635	0.506	0.718*	0.710#
SV	0.687	0.629	0.505	0.539	0.661	0.592	0.780#	0.812*
IV	0.671	0.549	0.554	0.489	0.579	0.500	0.691#	0.737*
MB	0.557	0.556	0.604	0.283	0.358	0.523	0.709#	0.712*
IPR	0.659	0.623	0.557	0.505	0.586	0.563	0.821#	0.855*
BC	0.672	0.564	0.634	0.471	0.557	0.401	0.771*	0.769#
LR	0.303	0.545	0.361	0.156	0.305	0.349	0.707#	0.716*

Note: the first and second best results are highlighted with $*$ and #, respectively.

** fast motion (FM), motion blur (MB), deformation (Def), low resolution (LR), occlusion (Occ), out-of-plane rotation (OPR), out of view (OV), in-plane-rotation (IPR), illumination variation (IV), background clutter (BC) , scale variation (SV)*

Results

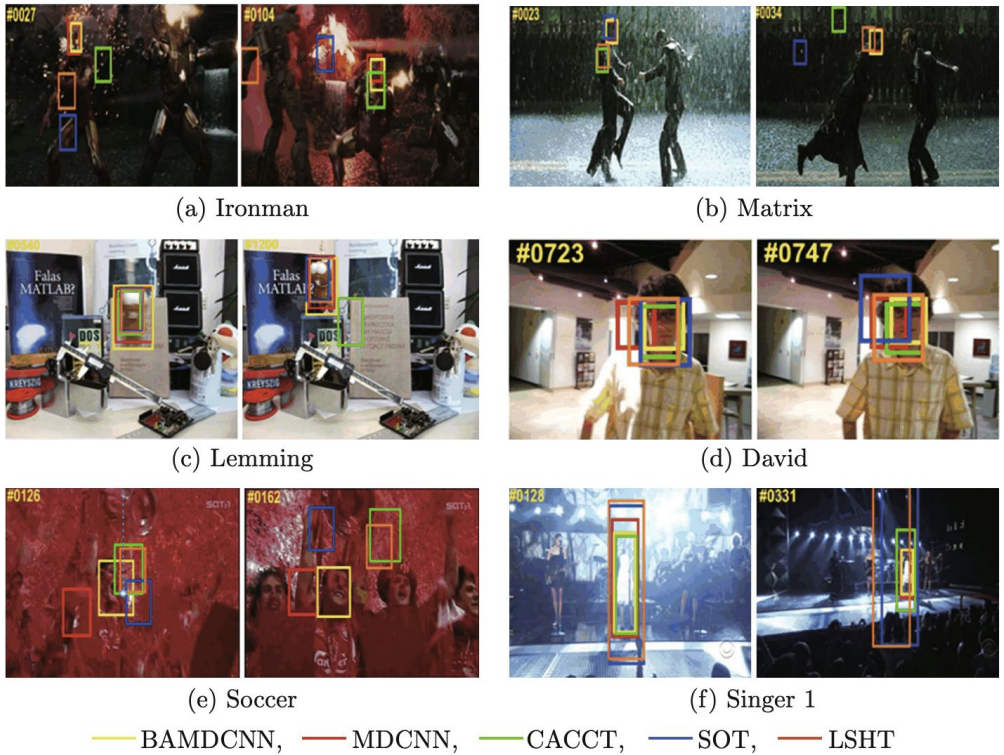


Fig. 4 Qualitative evaluation of BAMDCNN, MDCNN, CACCT, SOT and LSHT methods on six challenging sequences

Results

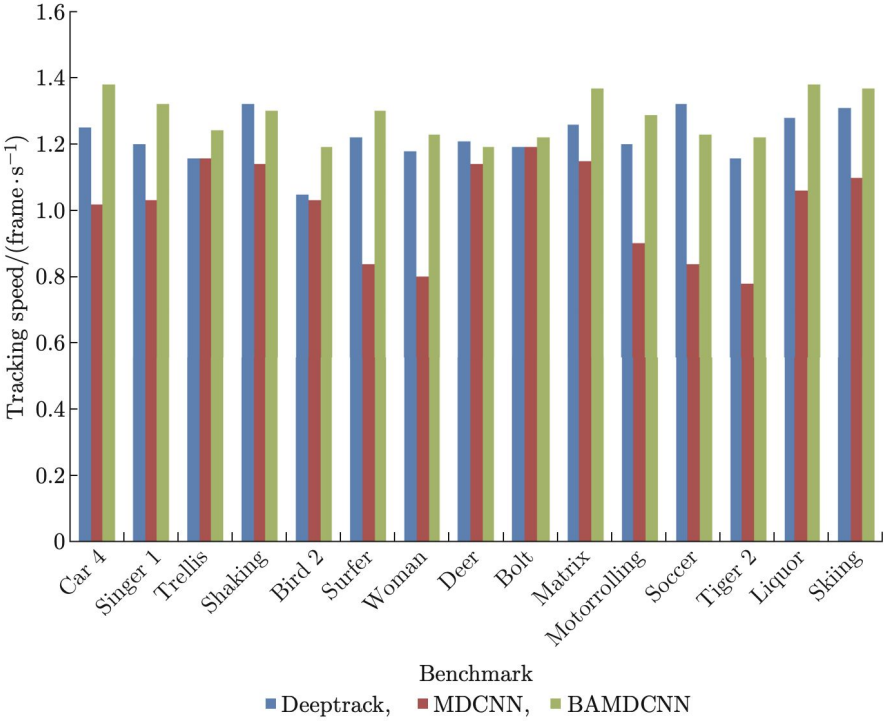


Fig. 5 Tracking speed of BAMDCNN, MDCNN and Deeptrack[10] on 15 challenging sequences

Thank You!
Questions?