Transductive Transfer Learning for Computer Vision

Teo de Campos
Nazli FarajiDavar and Josef Kittler

http://personal.ee.surrey.ac.uk/Personal/T.Decampos/
t.decampos@st-annes.oxon.org

Centre for Vision, Speech and Signal Processing (CVSSP)
UNIVERSITY OF SURREY

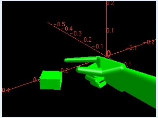
University of Bristol
21 April 2015

Outline

Outline

Previous projects

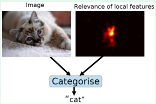
[1999-2001] Feature Selection, Face Detection and Recognition (before Viola&Jones' method)
University of São Paulo [4, 6, 8, 7, 18]



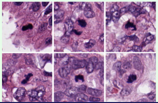
[2001-2006] 3D Hand Pose Estimation and Tracking
University of Oxford [9, 10]

[2005-2007] Computer Vision and Image Processing for Novel Display Systems
Sharp Labs of Europe

[2007] Character Recognition from Natural Scenes
Microsoft Research, India [3]



[2008-2009] Image Categorisation and Retrieval with Implicit Feedback
Xerox Research Centre Europe [5, 21, 26, 25]



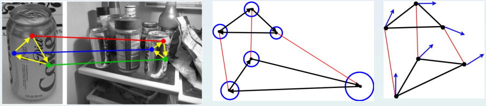
[2013-2014] Mitosis modelling using Gaussian Process Latent Variable Models
University of Sheffield [33]

Current project (1): Object matching using Keygraphs

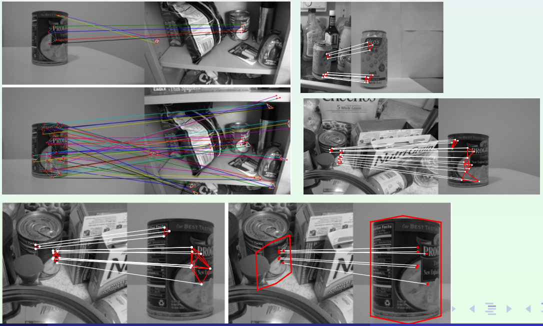
Collaboration with Estephan Wandekoken and Roberto Cesar-Jr [2]

Goal

Keygraph constraints

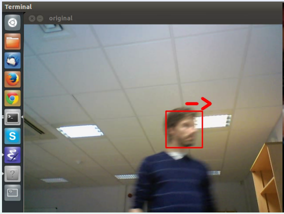


Results



Current project (2): S3A – Future Spatial Audio for an Immersive Listener Experience at Home

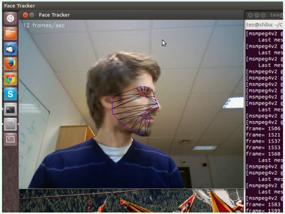
First vision task: 3D head tracking for adaptive sweet spot in 3D audio



Viola&Jones [34]

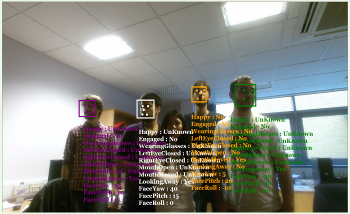
Zho&Ramanan [36]

Isard&Blake [20]



Saragih et al. [31]

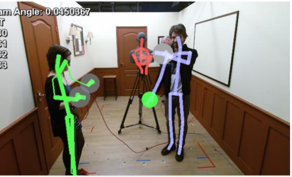
Fanelli et al. [12]

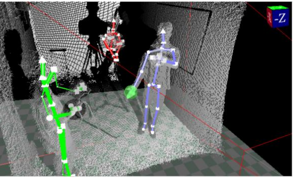


Kinect 2 SDK

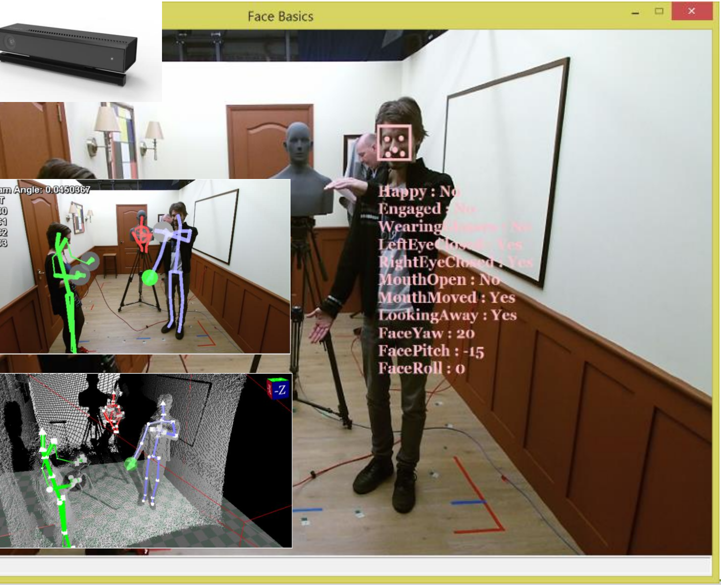
Head tracking pipeline

Head tracking pipeline

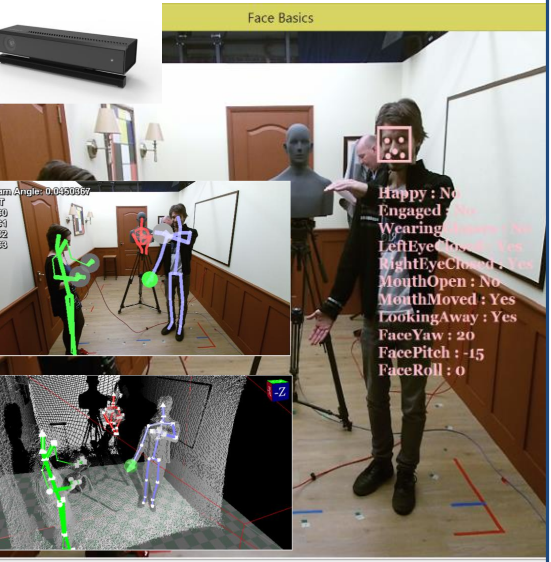




Head tracking pipeline

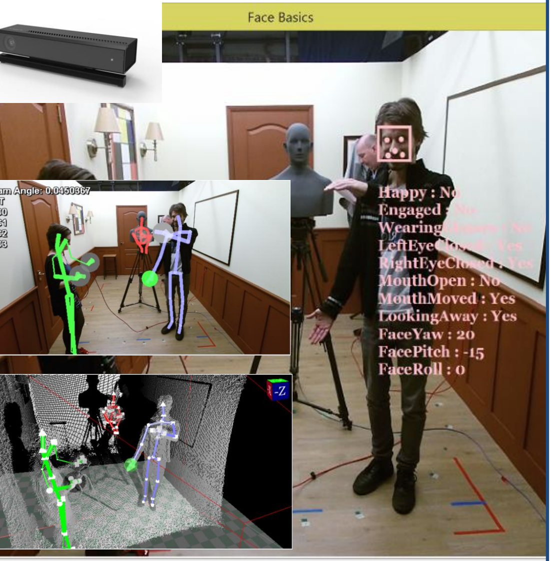


Head tracking pipeline

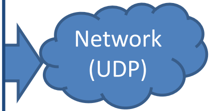



```
"nTime": "826637854179",
"People":
[
  {
    "iFace": "4",
    "bodyTime": "826637791679",
    "joints":
    {
      (...)
      "Head":
      {
        "X": "-1.132111",
        "Y": "0.04815324",
        "Z": "2.963933"
      },
      (...)
    },
    "jointOrientations":
    {
      (...)
      "Head":
      {
        "x": "0",
        "y": "0",
        "z": "0",
        "w": "0"
      },
      (...)
    },
    "validFace": "true",
    "faceRotation":
    {
      "x": "-0.06040232",
      "y": "0.1203637",
      "z": "0.03064966",
      "w": "0.9904165"
    },
    "faceYaw": "15",
    "facePitch": "-10",
    "faceRoll": "5",
    (...)
    "faceProperties":
    {
      "Happy": "No",
      "Engaged": "Maybe",
      "WearingGlasses": "No",
      "LeftEyeClosed": "Maybe",
      "RightEyeClosed": "Maybe",
      "MouthOpen": "No",
      "MouthMoved": "No",
      "LookingAway": "Maybe"
    }
  }
]
```

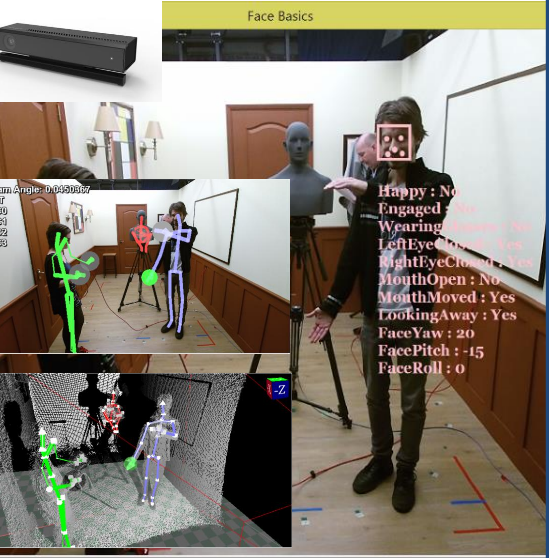
Head tracking pipeline



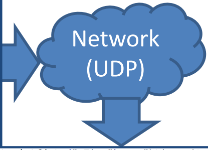

```
{
  "nTime": "826637854179",
  "People":
  [
    {
      "iFace": "4",
      "bodyTime": "826637791679",
      "joints":
      {
        (...)
        "Head":
        {
          "X": "-1.132111",
          "Y": "0.04815324",
          "Z": "2.963933"
        },
        (...)
      },
      "jointOrientations":
      {
        (...)
        "Head":
        {
          "x": "0",
          "y": "0",
          "z": "0",
          "w": "0"
        },
        (...)
      },
      "validFace": "true",
      "faceRotation":
      {
        "x": "-0.06040232",
        "y": "0.1203637",
        "z": "0.03064966",
        "w": "0.9904165"
      },
      "faceYaw": "15",
      "facePitch": "-10",
      "faceRoll": "5",
      (...)
      "faceProperties":
      {
        "Happy": "No",
        "Engaged": "Maybe",
        "WearingGlasses": "No",
        "LeftEyeClosed": "Maybe",
        "RightEyeClosed": "Maybe",
        "MouthOpen": "No",
        "MouthMoved": "No",
        "LookingAway": "Maybe"
      }
    }
  ]
```

Head tracking pipeline



"nTime": "826637854179",
"People":
[
  {
    "iFace": "4",
    "bodyTime": "826637791679",
    "joints":
    {
      (...)
      "Head":
      {
        "X": "-1.132111",
        "Y": "0.04815324",
        "Z": "2.963933"
      },
      (...)
    },
    "jointOrientations":




```
cout << "nTime: " << tree.get<uint64_t>("nTime") << "\tin units of
cout <<  "tree.count(People): " << tree.count("People") << endl;
for(auto&& kv : tree.get_child("People"))
{
  cout << "  ----\n  iFace: " << kv.second.get<int>("iFace") << end
  cout << "  Head position:\n    X=" << kv.second.get<float>("joint
  cout << "    Y=" << kv.second.get<float>("joints.Head.Y") << endl
  cout << "    Z=" << kv.second.get<float>("joints.Head.Z") << endl
  cout << "  Neck orientation:\n    x=" << kv.second.get<float>("jo
  cout << "    y=" << kv.second.get<float>("jointOrientations.Neck.
  cout << "    z=" << kv.second.get<float>("jointOrientations.Neck.
  cout << "    w=" << kv.second.get<float>("jointOrientations.Neck.
  if(kv.second.get<string>("validFace") == "true"){
    cout << "  Face detected in RGB!" << endl;
    cout << "    Face orientation quaternion:" << endl;
    cout << "      x=" << kv.second.get<float>("faceRotation.x") <<
    cout << "      y=" << kv.second.get<float>("faceRotation.y") <<
    cout << "      z=" << kv.second.get<float>("faceRotation.z") <<
    cout << "      w=" << kv.second.get<float>("faceRotation.w") <<
    cout << "    Face rotation in Euler angles (5 degrees of precis
    cout << "      Yaw=" << kv.second.get<float>("faceYaw") << endl
    cout << "      Pitch=" << kv.second.get<float>("facePitch") <<
    cout << "      Roll=" << kv.second.get<float>("faceRoll") << en
    cout << "    Face properties:" << endl;
    for(auto&& face_properties : kv.second.get_child("facePropertie
      cout << "      " << face_properties.first << ": " << face_pro
    }
  }
}
```

"RightEyeClosed": "Maybe",
      "MouthOpen": "No",
      "MouthMoved": "No",
      "LookingAway": "Maybe"
    }
  }
]

BBC
University of Salford MANCHESTER
EPSRC Pioneering research and skills
UNIVERSITY OF SURREY

Outline

Motivation: changes in the feature space

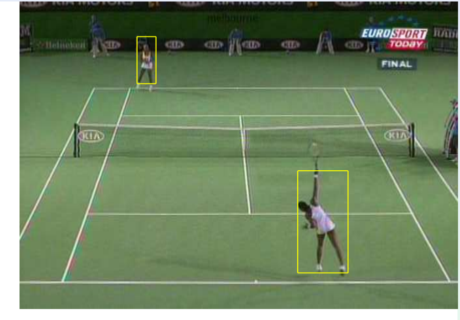
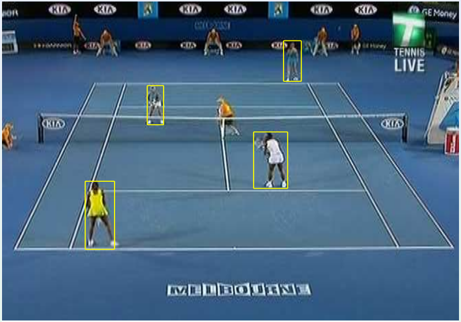
e.g. different image acquisition scenarios

Office+Caltech Dataset [30]:

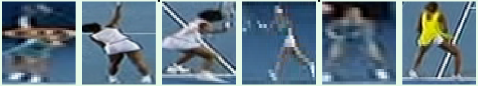
Webcam vs. Amazon

Action recognition in sports

different matches generate diffent visual features

TWSA03: Aus2003-Singles (PAL)	TWDA09: Aus2009-Doubles (NTSC)
	

serve (72)	hit (214)	non-hit (944)	serve (36)	hit (135)	non-hit (1064)



ACASVA: Player Action Dataset `http://cvssp.org/acasva/Downloads.html`

Domain change

the same actions appear different in other sports

BMSB08: Badminton, Beijing Olympics

serve (8)	hit (458)	non-hit (706)

Domain change

the same actions appear different in other sports

BMSB08: Badminton, Beijing Olympics

serve (8)	hit (458)	non-hit (706)

Other Transfer Learning Datasets

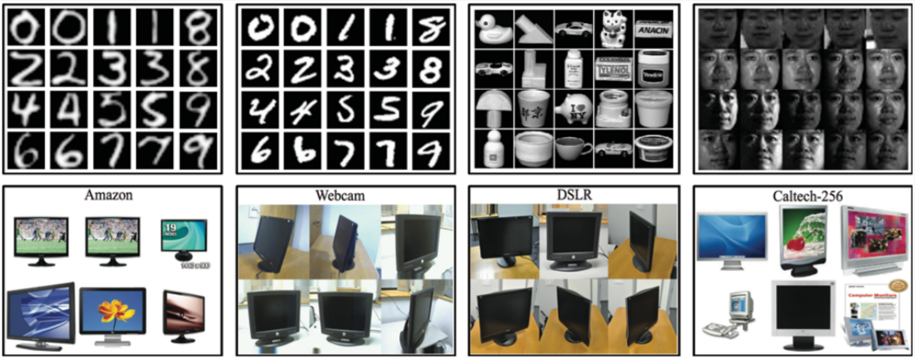
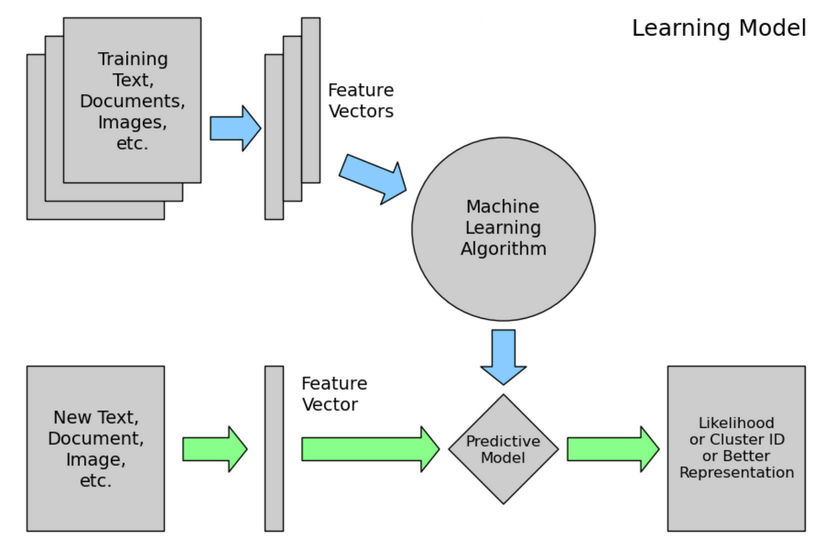


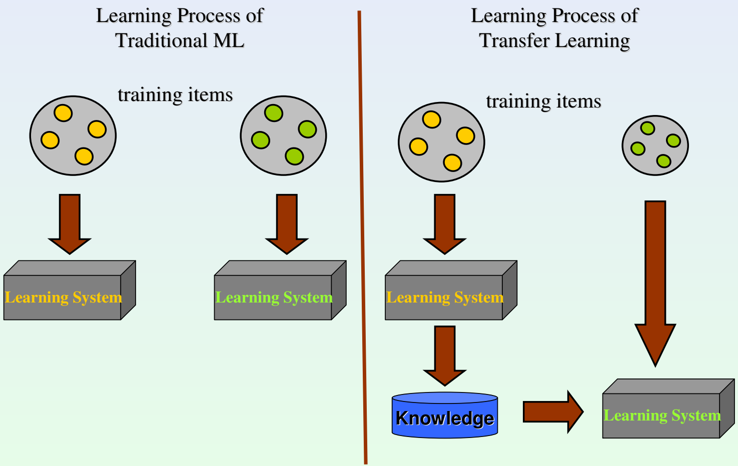
Figure 2. USPS, MNIST, COIL20, PIE, Office, and Caltech-256.

[24]

Statistical Pattern Recognition

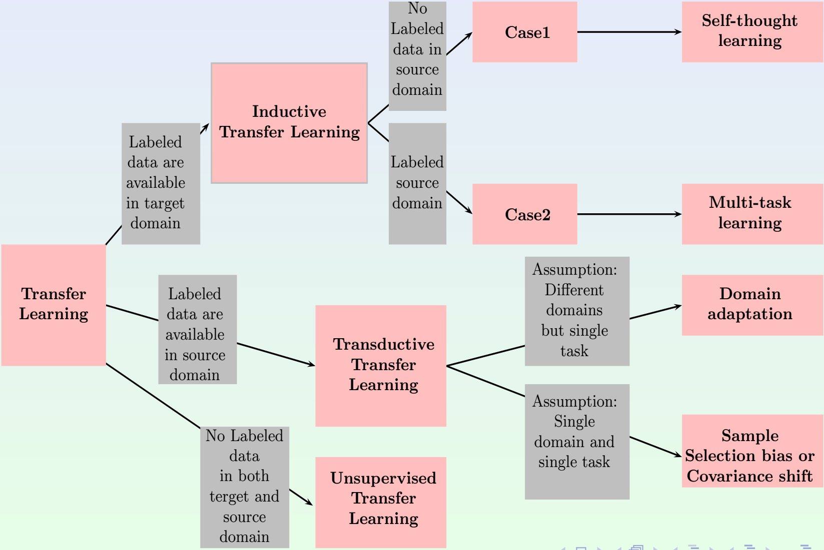


Traditional ML vs. TL



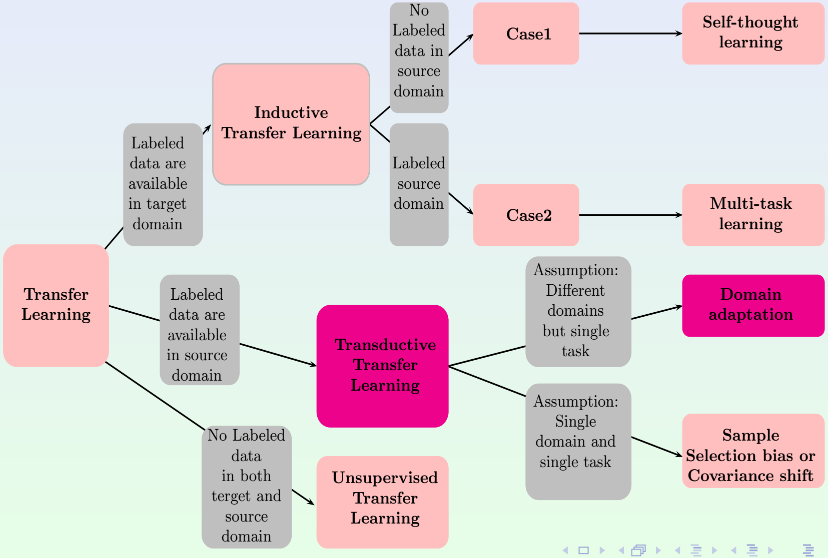
Transfer Learning Taxonomy

by Pan&Yang [28]



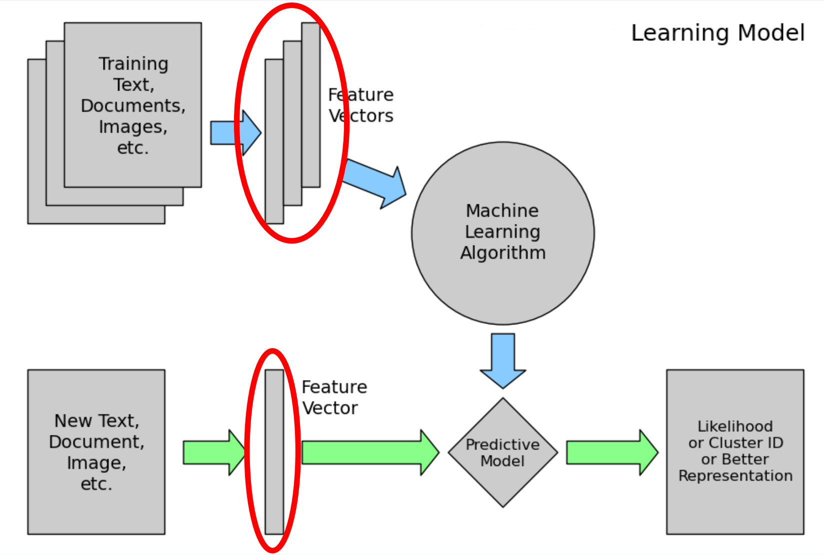
Transfer Learning Taxonomy – Transductive TL

or Unsupervised Domain Adaptation

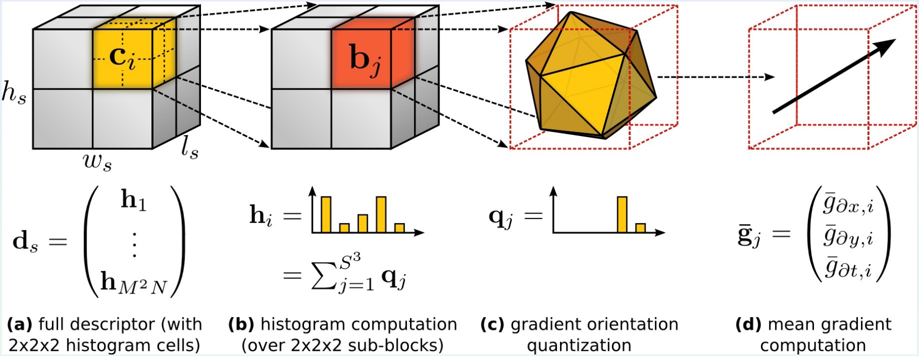


Outline

Feature Extraction



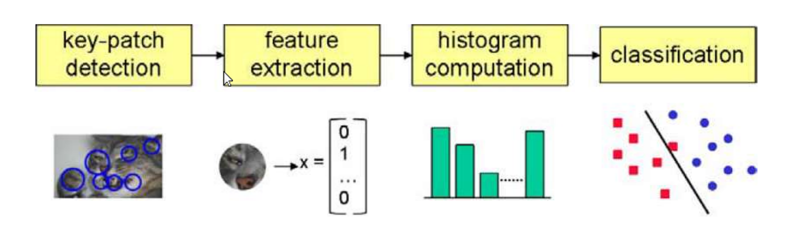
Player Action Data: HOG3D Features



Kläser *et al.*'s HOG3D Diagram, ©authors of [22]

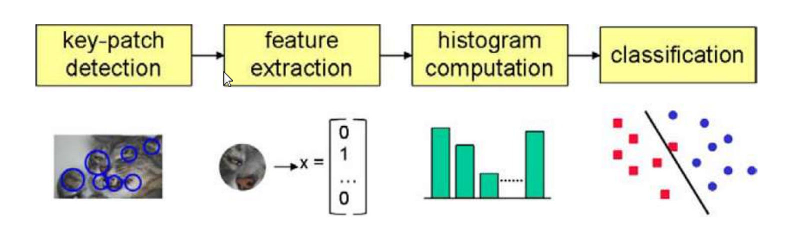
Caltech+Office

BoW-SURF Features© [29]

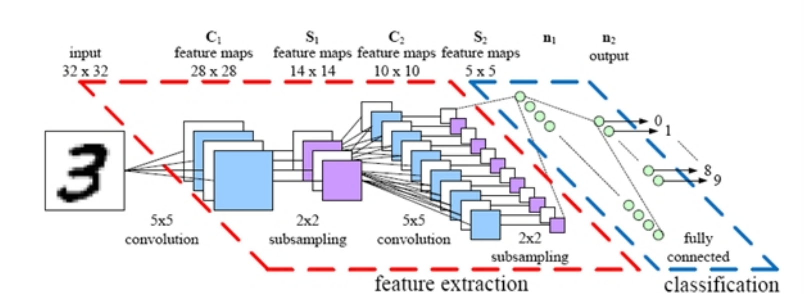


Caltech+Office

BoW-SURF Features© [29]



Convolutional Neural Network Features© [35]



Outline

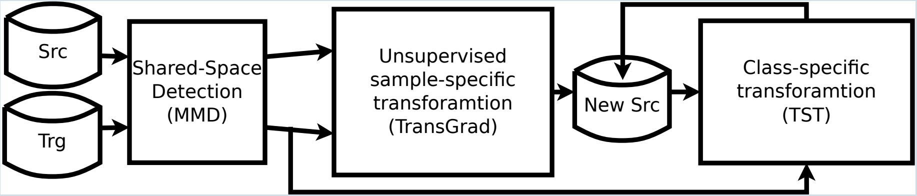
Transductive transfer learning by feature space transformation

Data

Input $D^{src} = \{(\mathbf{x}^{src}{}_i), y^{src}{}_i\}$, $D^{trg} = \{\mathbf{x}^{trg}{}_i\}$ and classification parameters

Output $P(y^{trg} \mid \mathbf{x}^{trg})$

Flowchart

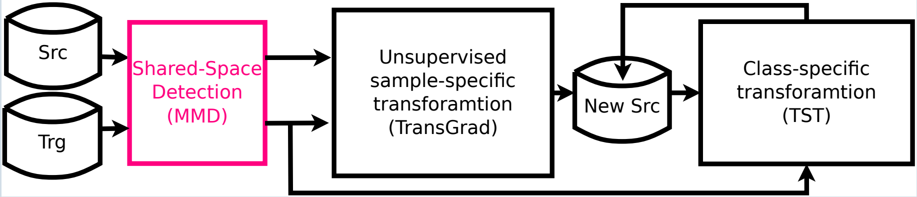


Transductive transfer learning by feature space transformation

Data

Input $D^{src} = \{(\mathbf{x}^{src}{}_i), y^{src}{}_i\}$, $D^{trg} = \{\mathbf{x}^{trg}{}_i\}$ and classification parameters
Output $P(y^{trg} \mid \mathbf{x}^{trg})$

Shared space detection using Maximum Mean Discrepancy



Shared space detection based on Maximum Mean Discrepancy (MMD)

Given the data matrix $\mathbf{X} = [\mathbf{X}^{src}; \mathbf{X}^{trg}]$, the goal is to get a linear projection $G(\mathbf{X}) = A^\top \mathbf{X}$ that

- preserves the maximum amount of information, i.e., maximises the data variance

$$\max_{A^T A = I} tr(A^T \mathbf{X} H \mathbf{X}^T A) \; , \tag{1}$$

  where H is the centring matrix $H = I - \frac{1}{n_{src}+n_{trg}} \mathbb{1}$
  and $\mathbb{1}$ is a $(n_{src} + n_{trg}) \times (n_{src} + n_{trg})$ matrix of ones;
- minimises the mean discrepancy between the two domains

$$\left\| \frac{1}{n_{src}} \sum_{i=1}^{n_{src}} A^T \mathbf{x}_i - \frac{1}{n_{trg}} \sum_{j=n_{src}+1}^{n_{src}+n_{trg}} A^T \mathbf{x}_j \right\|^2 = tr(A^T \mathbf{X} M \mathbf{X}^T A) \tag{2}$$

  where

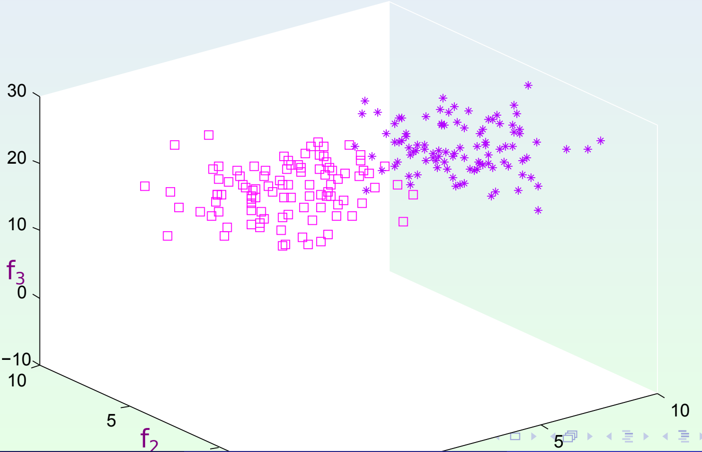
$$M^{ij} = \begin{cases} \frac{1}{n_{src} n_{src}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^{src} \\ \frac{1}{n_{trg} n_{trg}}, & \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}^{trg} \\ \frac{1}{n_{src} n_{trg}}, & \text{otherwise.} \end{cases}$$

MMD

That is solved via this eigendecomposition problem:

$$(\mathbf{X}M\mathbf{X}^\top + \epsilon I)A = \mathbf{X}H\mathbf{X}^\top D, \tag{3}$$

obtaining the eigenvectors in the columns of A and the eigenvalues in the diagonal matrix D.

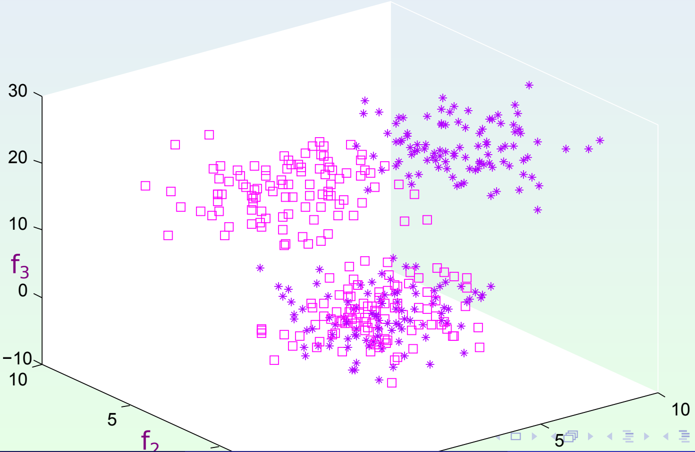


MMD

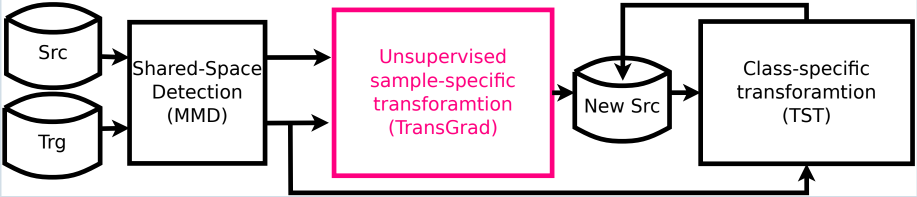
That is solved via this eigendecomposition problem:

$$(\mathbf{X}M\mathbf{X}^\top + \epsilon I)A = \mathbf{X}H\mathbf{X}^\top D, \qquad (3)$$

obtaining the eigenvectors in the columns of A and the eigenvalues in the diagonal matrix D.



TTM Diagram



Src
Trg
Shared-Space
Detection
(MMD)
Unsupervised
sample-specific
transforamtion
(TransGrad)
New Src
Class-specific
transforamtion
(TST)

Sample-based Adaptation with TransGrad

Problem Definition: Search for a sample-specific linear translation from source to target

$$G(\mathbf{x}) = \mathbf{x} + \gamma \mathbf{b}_{\mathbf{x}}$$

Assumption: The unlabelled target data is modelled by a GMM whose parameters are denoted by $\lambda = \{w_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}$

Solution: Maximize the likelihood of the translated source sample measured in the target domain

$$\max_{\mathbf{b}_{\mathbf{x}}} p(\mathbf{x} + \gamma \mathbf{b}_{\mathbf{x}} | \lambda)$$

Sample-based Adaptation with TransGrad

Using the Taylor expansion:

$$\max_{\mathbf{b_x}} p(\mathbf{x} + \gamma \mathbf{b_x} | \lambda) \approx p(\mathbf{x}|\lambda) + \gamma \nabla_{\mathbf{x}} p(\mathbf{x}|\lambda)^\top \mathbf{b_x}$$

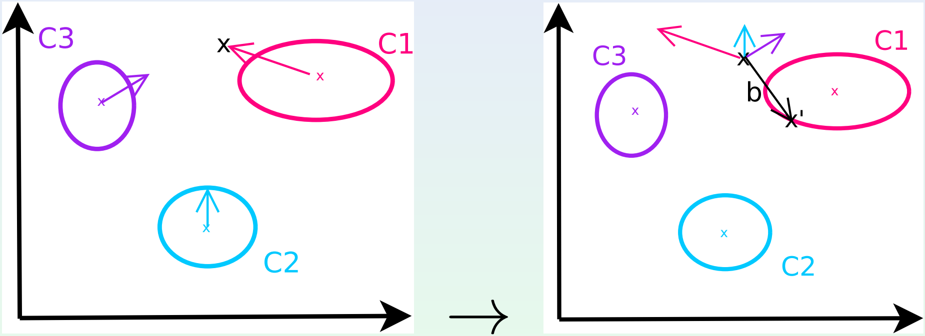
$$\text{s.t. } \mathbf{b_x}^\top \mathbf{b_x} = 1$$

Moving the source data-point in the direction of maximum gradient of the function $p(\mathbf{x}|\lambda)$:

$$\mathbf{b_x} = \nabla_{\mathbf{x}} p(\mathbf{x}|\lambda) = \sum_{k=1}^{K} w_k p(\mathbf{x}|\lambda_k) \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)$$

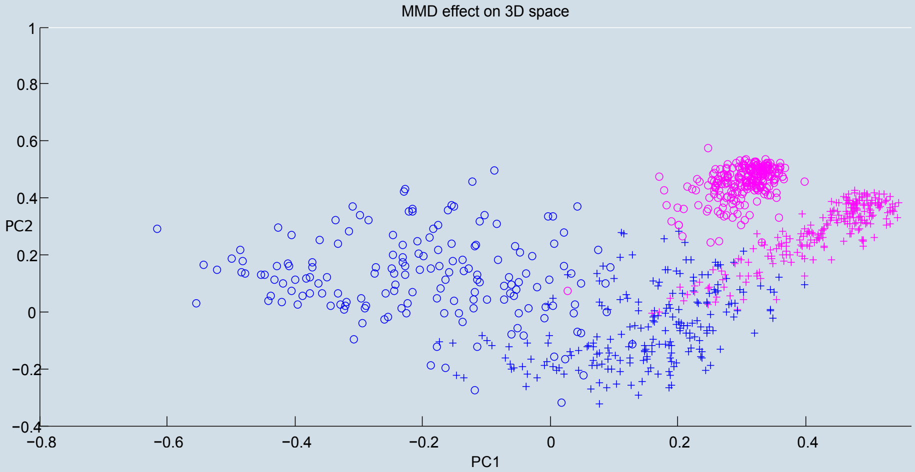
See [14] for details.

TransGrad illustration



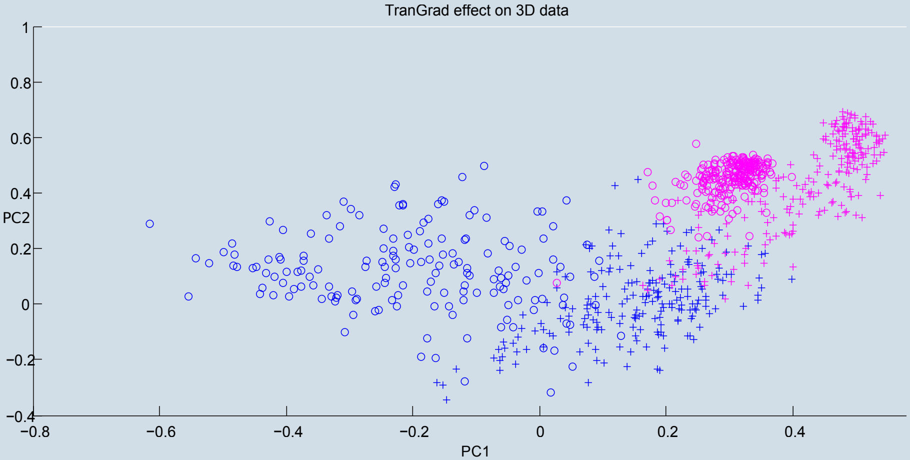
TransGrad on the digits datasets

Data after MMD



TransGrad on the digits datasets

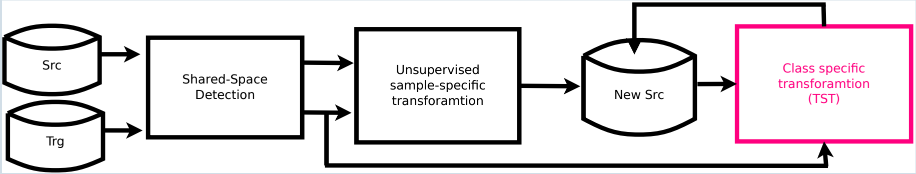
Data after TransGrad



Outline

Semi-supervised transformation

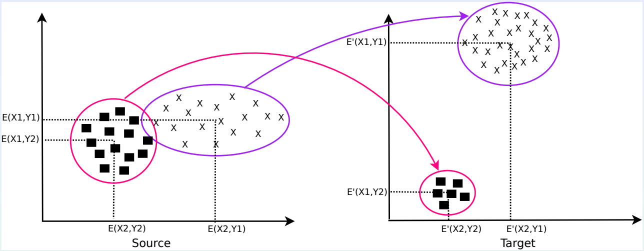
TTM Diagram



Algorithm

1. For each target sample $\mathbf{x}_i^{trg}$, estimate the posterior $P_{\Lambda_{src}}(y|\mathbf{x}_i^{trg})$ using the model trained with source data Λ_{src}
2. Learn the parameters of a transformation G from source to target spaces
3. Apply $G(x_j^i)$ on the source data set
4. Re-train the classifier and obtain $P_{\Lambda_{G(src)}}(y^{trg}|\mathbf{x}^{trg})$

Scaling / reweighting

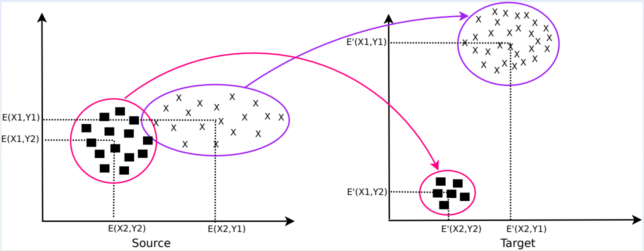


The transformation parameters are derived as: $\forall i = 1 : N^{src}_{train}$, $G(x^i_j) = x^i_j \frac{E^{trg}_{\Lambda_{src}}[x_j, y_i]}{E^{src}[x_j, y_i]}$

where the x^i_j is the j^{th} feature of sample x^i and Λ_{src} is the classification model learned using source samples and $E^{trg}_{\Lambda_{src}}$ is computed by:

$$E^{trg}[x_j, y] \approx E^{trg}_{\Lambda_{src}}[x_j, y] = \frac{\sum_{i=1}^{N^{trg}_{test}} x^i_j P_{\Lambda_{src}}(y|x_i)}{\sum_{i=1}^{N^{trg}_{test}} P_{\Lambda_{src}}(y|x_i)}$$

Scaling / reweighting

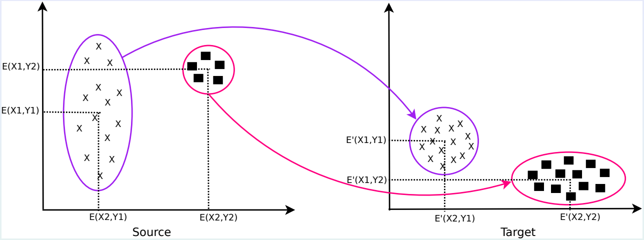


The transformation parameters are derived as: $\forall i = 1 : N^{src}_{train}, \; G(x^i_j) = x^i_j \frac{E^{trg}_{\Lambda_{src}}[x_j, y_i]}{E^{src}[x_j, y_i]}$

where the x^i_j is the j^{th} feature of sample x^i and Λ_{src} is the classification model learned using source samples and $E^{trg}_{\Lambda_{src}}$ is computed by:

$$E^{trg}[x_j, y] \approx E^{trg}_{\Lambda_{src}}[x_j, y] = \frac{\sum_{i=1}^{N^{trg}_{test}} x^i_j P_{\Lambda_{src}}(y|x_i)}{\sum_{i=1}^{N^{trg}_{test}} P_{\Lambda_{src}}(y|x_i)}$$

Translating and Scaling Transformation (TST)



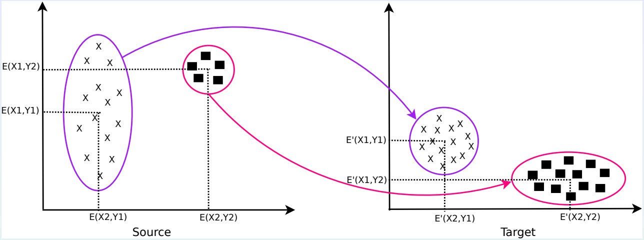
The transformation parameters will be estimated by:

$$\forall i = 1 : N^{src}_{train},\ G(x^i_j) = \frac{x^i_j - E^{src}[x_j, y_i]}{\sigma^{src}_{j,y_i}} \sigma^{trg}_{j,y_i} + E^{trg}_{\Lambda_{src}}[x_j, y_i]$$

where σ^{src}_{j,y_i} is the standard deviation of the feature x_j of the source samples $\mathbf{x}_k$ labelled as y_i and:

$$\sigma^{trg}_{j,y_i} = \sqrt{\frac{\sum_{k=1}^{N^{trg}_{test}} (x^k_j - E^{trg}_{\Lambda_{src}}[x_j,y_i])^2 P_{\Lambda_{src}}(y_i|\mathbf{x}_k)}{\sum_{k=1}^{N^{trg}_{test}} P_{\Lambda_{src}}(y_i|\mathbf{x}_k)}}$$

Translating and Scaling Transformation (TST)



The transformation parameters will be estimated by:

$$\forall i = 1 : N^{src}_{train},\ G(x^i_j) = \frac{x^i_j - E^{src}[x_j, y_i]}{\sigma^{src}_{j,y_i}} \sigma^{trg}_{j,y_i} + E^{trg}_{\Lambda src}[x_j, y_i]$$

where σ^{src}_{j,y_i} is the standard deviation of the feature x_j of the source samples $\mathbf{x}_k$ labelled as y_i and:

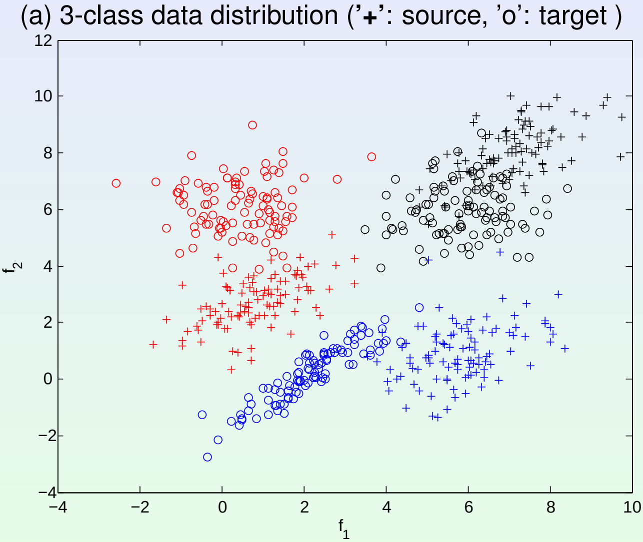
$$\sigma^{trg}_{j,y_i} = \sqrt{\frac{\sum_{k=1}^{N^{trg}_{test}} (x^k_j - E^{trg}_{\Lambda src}[x_j, y_i])^2 P_{\Lambda src}(y_i|\mathbf{x}_k)}{\sum_{k=1}^{N^{trg}_{test}} P_{\Lambda src}(y_i|\mathbf{x}_k)}}$$

Transformation Trade-off Factor

A trade-off factor θ has been introduced to control the degree to which we use the target conditional estimates to alter the source conditionals:

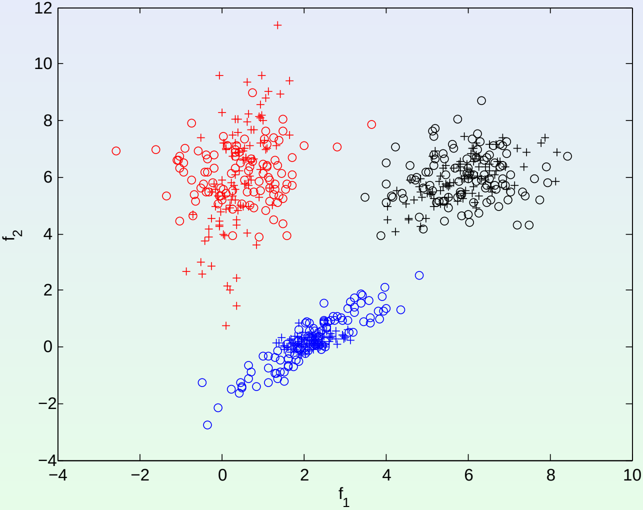
$$G'(x_j^i) = (1 - \theta)x_j^i + \theta G(x_j^i)$$

Exp.: Synthetic 3-class Dataset



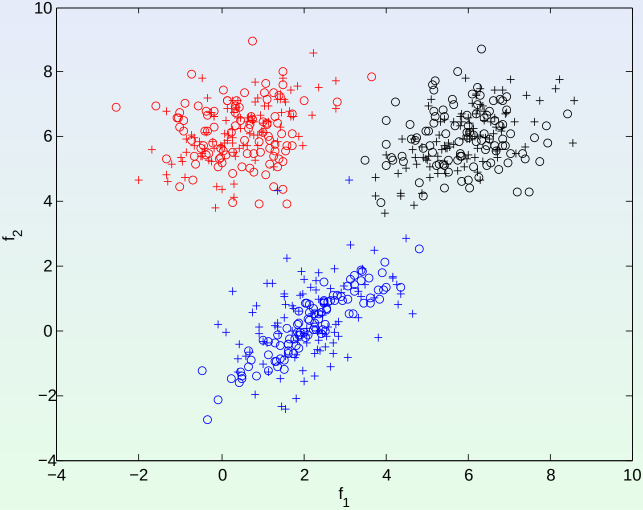
Exp.: Synthetic 3-class Dataset

(b) source data transformation using the re-weighting

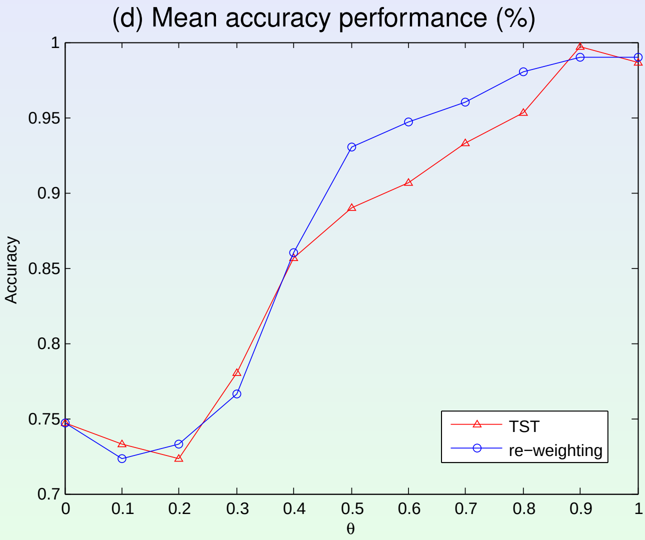


Exp.: Synthetic 3-class Dataset

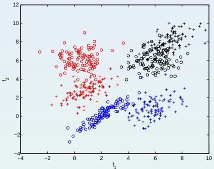
(c) source data transformation using the translation+scaling



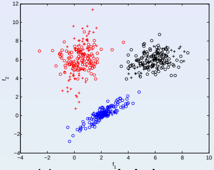
Exp.: Synthetic 3-class Dataset



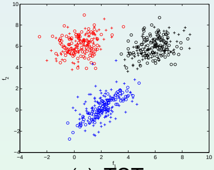
Exp.: Synthetic 3-class Dataset



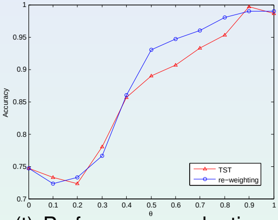
(q)



(r) re-weighting



(s) TST

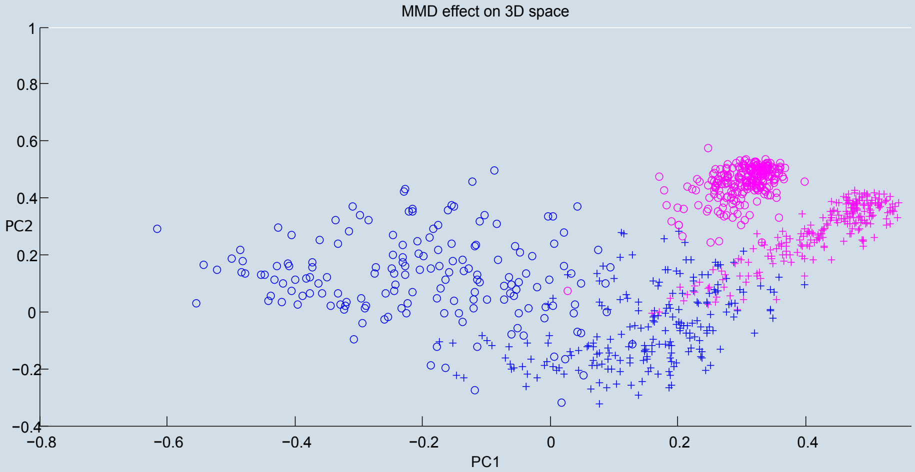


(t) Performance evaluation

dark ('+'):source light ('o'):target Classifier: KDA

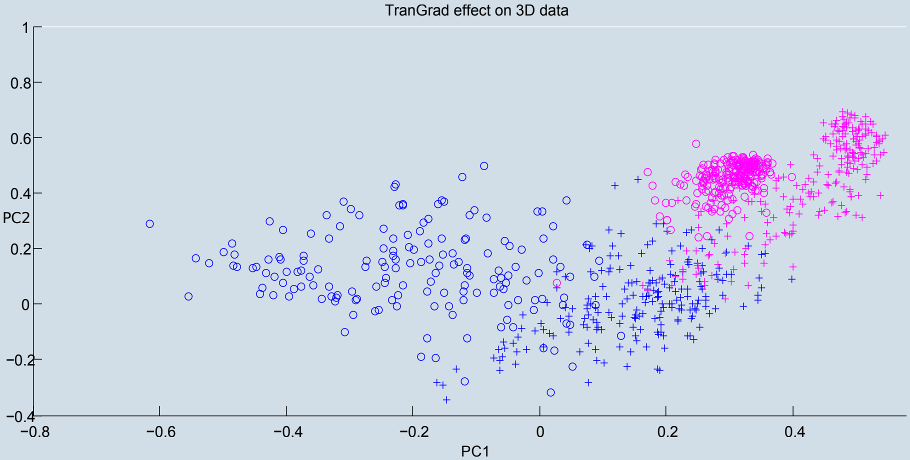
TTM step 1

Data after MMD



TTM step 2

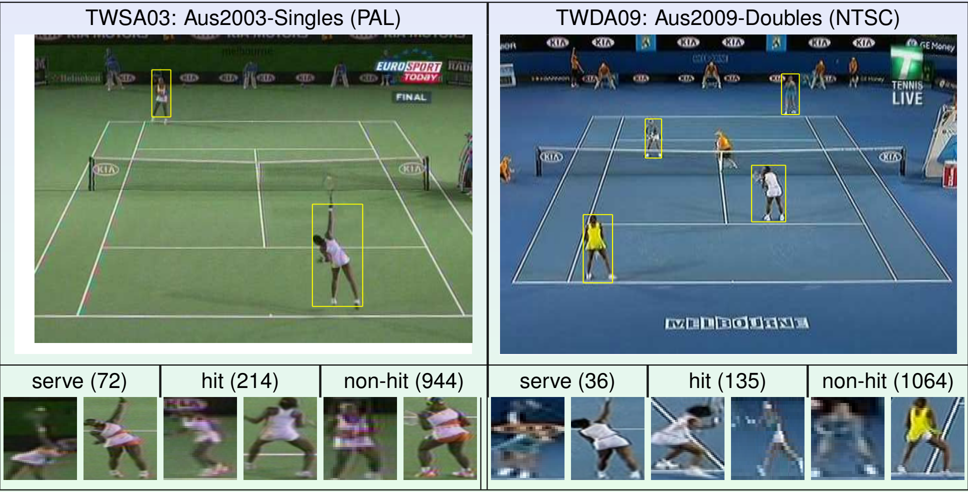
Data after TransGrad



TTM step 3

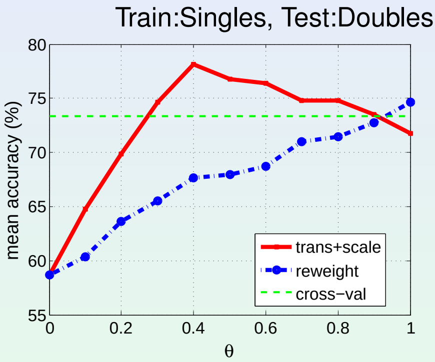
Data after TST

ACASVA: Player Actions Database

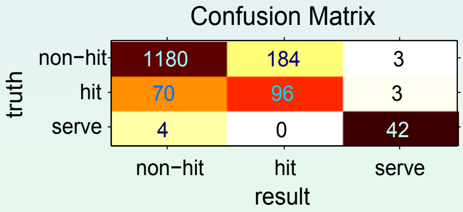


ACASVA: Player Action Dataset

Exp.: Tennis Singles → Tennis Doubles



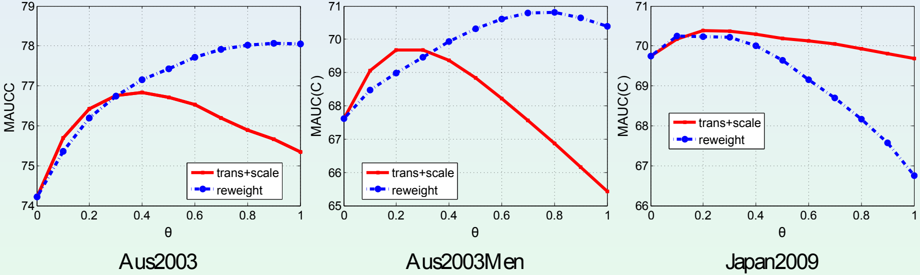
Confusion matrix at $\theta = 0.4$



Published in [15]

Exp.: Badminton → Tennis Singles

Published in [16, 17]



Remarks

- The initial target posterior estimation is based on the source model
- The conditional transformations (TST) do not include an initial marginal distribution adaptation scheme – that's why we combined TST with MMD and TransGrad
- Iterating on the proposed transformations can lead to further improvement
- A stopping criterion is required if iterations are used.

Outline

Classifiers evaluations on individual domains

Datasets	MNIST	USPS	COIL1	COIL2	Caltech	Amazon	Webcam	DSLR
Nature	Digit	Digit	Object	Object	Object	Object	Object	Object
Features	Raw	Raw	Raw	Raw	SURF	SURF	SURF	SURF
5-fold cross-val (PCA+NN)	91.97	93.64	99.02	98.91	38.80	60.59	79.58	**76.95**
" LR	86.15	89.22	92.36	92.22	56.27	72.46	80.01	67.49
" KDA	**94.05**	94.84	**100.00**	**99.71**	**58.16**	**78.73**	**89.54**	63.94
" SVM	91.80	**95.28**	99.72	99.44	57.17	74.86	86.44	75.80

ATTM Work-flow

ATTM Algorithm

Input: $\mathbf{X}^{src}, \mathbf{Y}^{src}, \mathbf{X}^{trg}$
Output: $\mathbf{Y}^{trg}$
1. Search for the shared subspace between the two domains
2. Adjust the marginal distribution mismatch between the two domains
3. Select the appropriate classifier, if it is kernel-based, tune σ using
while $T < 10$ and $|D^{\text{global}}(G^t(\mathbf{X}^{src}), \mathbf{X}^{trg})| > \textit{threshold}$ **do**
   4. Find the feature-wise TST transformation
   5. Transform the source domain clusters
   6. Retrain the classifier using the transformed source
end while

Global Domain Dissimilarity

Uses the mean classification (NN classifier) performance of samples labelled as source or target:

$$D^{global} = \frac{1}{2} \sum_{y_i \in \{src, trg\}} p(y_i = y|\mathbf{x})$$

Clusters Dissimilarity

Computed using the average dissimilarity between the source and target clusters $d(k^{src}, k^{trg})$

$$D^{clusters}(\mathbf{X}^{src}, \mathbf{X}^{trg}) = \frac{1}{K}\sum_{k=1}^{K} d(k^{src}, k^{trg})$$

where

- K is the number of classes
- $d(k^{src}, k^{trg})$ is the distance between the source class k^{src} centre and its nearest target cluster k^{trg}
- k^{trg} are obtained using K-means on the target data where the cluster centres are initialised using source class centres k^{src}.

Parameter Selection

D^{global}	$D^{clusters}$	Classifier	σ
$\Uparrow$	$\Uparrow$	NN	-
$\Downarrow$	$\Downarrow$	KDA	$\sigma = \sigma^{src}$
$\Downarrow$	$\Uparrow$	KDA	$\sigma' = \sigma^{src} \times \frac{const}{D^{\text{clusters}}(\mathbf{X}^{src}, \mathbf{X}^{trg})}$

NN : Nearest Neighbour classifier

KDA: Kernel discriminant Analysis with RBF kernel

σ: lengthscale parameter of the RBF kernel (KDA classifier parameter)

Outline

Recognition accuracies using Bag-of-SURF features

TTL test	base-line NN	base-line PCA	TCA [27]	TSL [32]	GFK (PLS, PCA) [19]	JDA (1NN) [24]	TTM0 (TST NN)	TTM1 (MMD + TTM0)	TTM2 (Trans-Grad + TTM1)	AJDA (Adapt. JDA)	ATTM (Adapt. TTM2)
M→ U	65.94	66.22	56.28	66.06	67.22	67.28	75.94	76.61	77.94	67.28	**77.94**
U→M	44.70	44.95	51.05	53.75	46.45	59.65	59.79	59.41	61.15	59.65	**61.15**
COIL1→2	83.61	84.72	88.47	88.06	72.50	89.31	88.89	88.75	93.19	**94.31**	92.64
COIL2→1	82.78	84.03	85.83	87.92	74.17	88.47	88.89	88.61	88.75	**92.36**	91.11
C→ A	23.70	36.95	38.20	44.47	41.4	44.78	39.87	44.25	46.76	58.56	**60.85**
C→ W	25.76	32.54	38.64	34.24	40.68	41.69	41.02	39.66	41.02	48.81	**62.03**
C→ D	25.48	38.22	41.40	43.31	41.1	45.22	50.31	44.58	47.13	45.86	**50.32**
A→ C	26.00	34.73	27.76	37.58	37.9	39.36	36.24	35.53	39.62	40.43	**42.92**
A→ W	29.83	35.59	37.63	33.90	35.7	37.97	37.63	42.37	39.32	49.83	**50.51**
A→ D	25.48	27.39	33.12	26.11	36.31	39.49	33.75	29.30	29.94	38.21	**39.49**
W→ C	19.86	26.36	29.30	29.83	29.3	31.17	26.99	29.83	30.36	**35.80**	34.02
W→ A	22.96	31.00	30.06	30.27	35.5	32.78	29.12	30.69	31.11	38.94	**39.67**
W→ D	59.24	77.07	87.26	87.26	80.89	89.17	85.98	89.17	89.81	89.17	**89.81**
D→ C	26.27	29.65	31.70	28.50	30.28	31.52	29.65	31.25	32.06	28.31	**32.41**
D→ A	28.50	32.05	32.15	27.56	36.1	33.09	31.21	29.75	30.27	37.47	**38.73**
D→ W	63.39	75.93	86.10	85.42	79.1	89.49	85.08	**90.84**	88.81	89.49	88.81
Avg	43.06	49.23	50.35	52.34	50.00	54.88	54.12	55.10	56.20	59.17	**60.72**

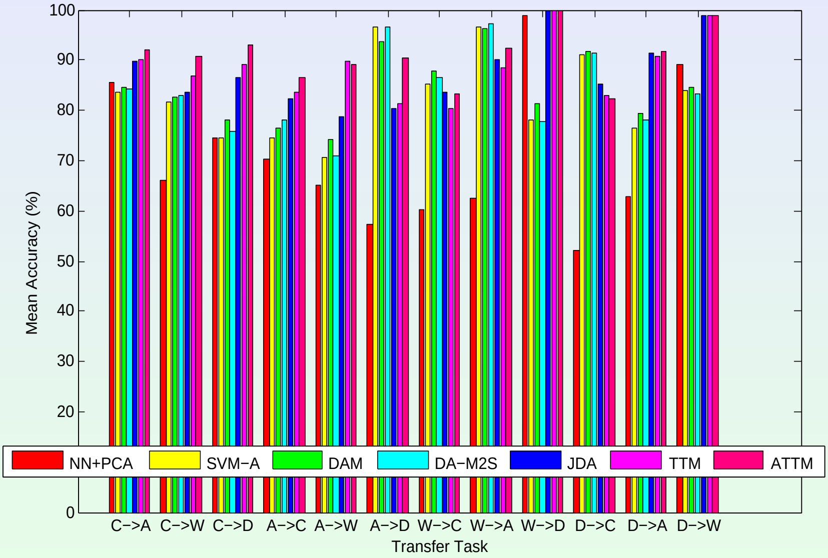
M: MNIST, U: USPS, C1: COIL-1, C2: COIL-2, C: Caltech, A: Amazon, W: Webcam, and D: DSLR

Recognition accuracies using DeCAF features

	Basline (NN)	SVM-A [1]	DAM [11]	DA-M2S (w/o depth) [23]	JDA (1NN) [24]	TTM (NN) [14]	ATTM [13]
C→ A	85.70	83.54	84.73	84.27	89.77	89.98	**92.17**
C→ W	66.10	81.72	82.48	82.87	83.73	86.78	**90.84**
C→ D	74.52	74.58	78.14	75.83	86.62	89.17	**92.99**
A→ C	70.35	74.36	76.60	78.11	82.28	83.70	**86.55**
A→ W	64.97	70.58	74.32	71.04	78.64	**89.81**	89.15
A→ D	57.29	96.56	93.82	**96.62**	80.25	81.36	90.45
W→ C	60.37	85.37	**87.88**	86.38	83.53	80.41	83.44
W→ A	62.53	96.71	96.31	**97.12**	90.19	88.52	92.27
W→ D	98.73	78.14	81.27	77.60	100	100	**100**
D→ C	52.09	91.00	**91.75**	91.37	85.13	82.90	82.28
D→ A	62.73	76.61	79.39	78.14	91.44	90.81	**91.65**
D→ W	89.15	83.89	84.59	83.31	98.98	98.98	**98.98**
Avg	70.33	83.95	84.06	84.97	87.55	87.87	**90.90**

C: Caltech, A: Amazon, W: Webcam, and D: DSLR

Recognition accuracies using DeCAF features



Outline

Conclusions

- We have proposed a transductive transfer learning system based on three processes:
  1. shared space detection;
  2. unsupervised sample-wise adaptation of marginal distributions and
  3. iterative semi-supervised transformation.
- We also proposed to use domain dissimilarity measures to automatically select classifiers and the kernel parameter.
- The combination above leads to state-of-the-art results.
- Its computational complexity is smaller than that of other recent methods, e.g. JDA takes, on average, 21.4s to transfer from MNIST to USPS, whilst our method takes 4.4s.
- Transductive transfer learning methods can be applied whenever a batch of unlabeled test samples is available, as long as a reasonable initial estimate of $P_{\Lambda_{src}}(y|\mathbf{x}_i)$ can be obtained.
- **Long term goal:** build a fully automated system, capable of detecting anomalies or domain changes to trigger a transfer learning method.

Conclusions

- We have proposed a transductive transfer learning system based on three processes:
  1. shared space detection;
  2. unsupervised sample-wise adaptation of marginal distributions and
  3. iterative semi-supervised transformation.
- We also proposed to use domain dissimilarity measures to automatically select classifiers and the kernel parameter.
- The combination above leads to state-of-the-art results.
- Its computational complexity is smaller than that of other recent methods, e.g. JDA takes, on average, 21.4s to transfer from MNIST to USPS, whilst our method takes 4.4s.
- Transductive transfer learning methods can be applied whenever a batch of unlabeled test samples is available, as long as a reasonable initial estimate of $P_{\Lambda_{src}}(y|\mathbf{x}_i)$ can be obtained.
- **Long term goal:** build a fully automated system, capable of detecting anomalies or domain changes to trigger a transfer learning method.

References I

[1] L. Bruzzone and M. Marconcini.
Domain adaptation problems: A DASVM classification technique and a circular validation strategy.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 32(5):770–787, 2010.

[2] E. Dazzi, T. deCampos, and R. Cesar-Jr.
Improved object matching using structural relations.
In *IAPR Joint International Workshops on Statistical Techniques in Pattern Recognition and Structural and Synthactic Pattern Recognition (S+SSPR)*, Lecture Notes in Computer Science (LNCS), pages 444–453, 2014.

[3] T. E. deCampos, B. R. Babu, and M. Varma.
Character recognition in natural images.
In *Proceedings of the International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, February 2009.

[4] T. E. deCampos, I. Bloch, and R. M. Cesar-Jr.
Feature selection based on fuzzy distances between clusters: First results on simulated data.
In *International Conference on Advances in Pattern Recognition (ICAPR)*, Rio de Janeiro, Brazil, 2001.

[5] T. E. deCampos, G. Csurka, and F. Perronnin.
Images as sets of locally weighted features.
Computer Vision and Image Understanding, 2012.

[6] T. E. deCampos, F. S. Feris, and R. M. Cesar-Jr.
Eigenfaces versus eigeneyes: First steps toward performance assessment of representarions for face recognition.
In *MICAI*, Acapulco, Mexico, 2000.

[7] T. E. deCampos, F. S. Feris, and R. M. Cesar-Jr.
A framework for face recognition from video sequences using gwn and eigenfeature selection.
In *WAICV*, Atibaia, Brazil, 2000.

[8] T. E. deCampos, F. S. Feris, and R. M. Cesar-Jr.
Improved face x non-face discrimination using fourier descriptors through feature selection.
In *SIBGRAPI*, Gramado, Brazil, 2000.

References II

[9] T. E. deCampos and D. W. Murray.
Regression-based hand pose estimation from multiple cameras.
In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, New York, June 2006. IEEE.

[10] T. E. deCampos, B. J. Tordoff, and D. W. Murray.
Recovering articulated pose: A comparison of two pre and postimposed constraint methods.
Transactions on Pattern Analysis and Machine Intelligence, 28(1), 2006.

[11] L. Duan, I. W. Tsang, D. Xu, and T. Chua.
Domain adaptation from multiple sources via auxiliary classifiers.
In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML, pages 289–296, New York, NY, USA, 2009. ACM.

[12] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool.
Random forests for real time 3d face analysis.
Int. J. Comput. Vision, 101(3):437–458, February 2013.

[13] N. FarajiDavar, T. deCampos, and J. Kittler.
Adaptive transductive transfer machine.
In *Preceedings of the British Machine Vision Conference (BMVC)*, Nottingham, September 2014.

[14] N. FarajiDavar, T. deCampos, and J. Kittler.
Transductive transfer machine.
In *Preceedings of the Asian Conference on Computer Vision (ACCV)*, Singapore, November 2014.

[15] N. FarajiDavar, T. E. deCampos, J. Kittler, and F. Yan.
Transductive transfer learning for action recognition in tennis games.
In *3rd International Workshop on Video Event Categorization, Tagging and Retrieval for Real-World Applications (VECTaR), in conjunction with ICCV*, 2011.

References III

[16] N. FarajiDavar, T. E. deCampos, D. Windridge, J. Kittler, and W. Christmas.
Domain adaptation in the context of sport video action recognition.
In *Domain Adaptation Workshop, in conjunction with NIPS*, 2011.

[17] N. FarajiDavar, T. deCampos D. Windridge, J. Kittler, and W. Christmas.
Domain adaptation in the context of sport video action recognition.
In *4th UK Computer Vision Student Workshop (BMVW), in conjunction with the British Machine Vision Conference (BMVC)*, Guildford, UK, 2012.

[18] F. S. Feris, T. E. deCampos, and R. M. Cesar-Jr.
Detection and tracking of facial features in video sequences.
In *MICAI*, Acapulco, Mexico, 2000.

[19] B. Gong, Y. Shi, F. Sha, and K. Grauman.
Geodesic flow kernel for unsupervised domain adaptation.
In *CVPR*, pages 2066–2073. IEEE, 2012.

[20] M. Isard and A. Blake.
Condensation – conditional density propagation for visual tracking.
International journal of computer vision, 29(1):5–28, 1998.

[21] A. Klami, C. Saunders, T. deCampos, and S. Kaski.
Can relevance of images be inferred from eye movements?
In *Proceeding of the International Conference on Multimedia Information Retrieval*, Vancouver, Canada, October 2008. ACM.

[22] A. Klaser, M. Marszalek, and C. Schmid.
A spatio-temporal descriptor based on 3D-gradients.
In *British Machine Vision Conference*, pages 995–1004, sep 2008.

[23] D. X. L. Chen, W. Li.
Recognizing rgb images by learning from rgb-d data.
In *IEEE International Conference on Computer Vision and Pattern Recognition*, CVPR, 2014.

References IV

[24] M. Long, J. Wang, G. Ding, and P. Yu.
Transfer learning with joint distribution adaptation.
In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[25] J. J. McAuley, T. deCampos, G. Csurka, and F. Perronnin.
Hierarchical image-region labeling via structured learning.
In *Proceedings of the British Machine Vision Conference*, London, September 2009.

[26] J. J. McAuley, T. E. deCampos, and T. S. Caetano.
Unified graph matching in euclidean spaces.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, 2010. IEEE.

[27] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang.
Domain adaptation via transfer component analysis.
IEEE Transactions on Neural Networks, 22(2):199–210, 2011.

[28] S. J. Pan and Q. Yang.
A survey on transfer learning.
IEEE Transactions on Knowledge and Data Engineering, 22:1345–1359, 2010.

[29] F. Perronnin, C. Dance, G. Csurka, and M. Bressan.
Adapted vocabularies for generic visual categorization.
In *In ECCV*, pages 464–475, 2006.

[30] K. Saenko, B. Kulis, M. Fritz, and T. Darrell.
Adapting visual category models to new domains.
In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 213–226, Berlin, Heidelberg, 2010. Springer-Verlag.

References V

[31] J. M. Saragih, S. Lucey, and J. F. Cohn.
Face alignment through subspace constrained mean-shifts.
In *12th International Conference on Computer Vision*, pages 1034–1041. IEEE, 2009.
DOI 10.1109/ICCV.2009.5459377.

[32] S. Si, D. Tao, and B. Geng.
Bregman divergence-based regularization for transfer subspace learning.
IEEE Trans. Knowl. Data Eng., 22(7):929–942, 2010.

[33] M. Veta, P. J. van Diest, S. M. Willems, H. Wang, A. Madabhushi, A. Cruz-Roa, F. Gonzalez, A. B. Larsen, J. S. Vestergaard, A. B. Dahl, D. C. Cirean, J. Schmidhuber, A. Giusti, L. M. Gambardella, F. B. Tek, T. Walter, C.-W. Wang, S. Kondo, B. J. Matuszewski, F. Precioso, V. Snell, J. Kittler, T. E. de Campos, A. M. Khan, N. M. Rajpoot, E. Arkoumani, M. M. Lacle, M. A. Viergever, and J. P. Pluim.
Assessment of algorithms for mitosis detection in breast cancer histopathology images.
Medical Image Analysis, 20(1):237 – 248, 2015.
DOI 10.1016/j.media.2014.11.010.

[34] P. Viola and M. J. Jones.
Robust real-time face detection.
Int. J. Comput. Vision, 57(2):137–154, May 2004.
DOI 10.1023/B:VISI.0000013087.49260.fb.

[35] J. Ward, S. Andreev, F. Heredia, B. Lazar, and Z. Manevska.
Efficient mapping of the training of convolutional neural networks to a cuda-based cluster.
Parallel Architecture Research Eindhoven, Feb 2015.

[36] X. Zhu and D. Ramanan.
Face detection, pose estimation, and landmark localization in the wild.
In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition, CVPR*, pages 2879–2886, Providence, RI, 19-21 June 2012.
DOI 10.1109/CVPR.2012.6248014.