L1-regularized Logistic Regression Stacking and Transductive CRF Smoothing for Action Recognition in Video

Svebor Karaman, Lorenzo Seidenari, Andrew D. Bagdanov, Alberto Del Bimbo

Media Integration and Communication Center (MICC)
University of Florence, Florence, Italy

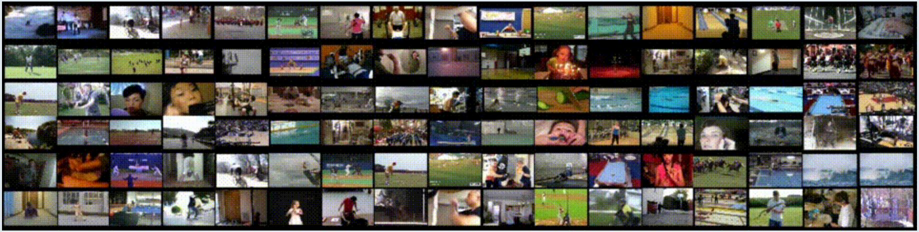
{svebor.karaman, lorenzo.seidenari}@unifi.it,
{bagdanov, delbimbo}@dsi.unifi.it

http://www.micc.unifi.it/vim/people

THUMOS Workshop

Text

First International Workshop on Action Recognition with a Large Number of Classes



- 101 Classes, 5 types: Human-Object Interaction, Human-Human Interaction, Body-Motion Only, Playing Musical Instruments, Sports.
- 13320 videos (25 groups)
- Pre-computed and pre-encoded (hard-assigned 4000 BoW) low-level features: STIP, Dense Trajectory Features (MBH, HOG, HOF, TR)
- 3 splits : 2/3 train, 1/3 test (disjoint groups in train/test)

Introduction

Our game plan and our goals

- Priority: establish a working BOW pipeline on given hard assigned coded features (MBH, HOG, HOF, STIP, TR) to establish our baseline
- Limitations:
  - Loss due of hard assignment
  - No contextual features
  - Lots and lots of classes and features, unclear how to fuse
- Goal 1: improve the features in our baseline
  - Use better encoding of provided features (after re-extraction)
  - Add static contextual features extracted from keyframes
- Goal 2: experiment with fusion schemes
  - Regularized stacking of experts
  - Transductive smoothing of expert outputs
- Note we did not use any external data or the provided attributes

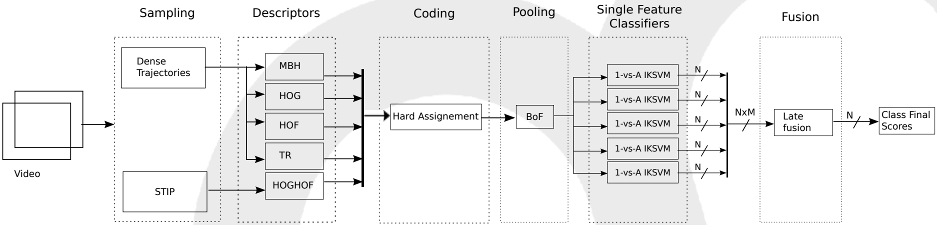
Baseline with provided features (Run-1)

Run 1: a respectable baseline

- Late fusion (sum) of 1-vs-All SVM classifiers (Histogram Intersection Kernel) learned on $M = 5$ features

$$\text{class}(\mathbf{x}) = \arg\max_{c} \sum_{f \in \mathcal{F}_{\text{org}}} E_c^f(\mathbf{x}) \qquad (1)$$

- Performance: 74.6% (Split1: 72.85%, Split2: 74.96% , Split3: 75.97%)



Better encoding of dense trajectories features

- Extraction of dense trajectories [Wang:2013]
  - On a modest cluster of 20CPUs:
    - 5 nodes
    - Quad Core 2.7Ghz CPUs
    - 48GB Total RAM
  - Total time to extract: 25h
  - Disk usage: 660GB
- Extracted features:
  - Separate x- and y-components (MBHx and MBHy)
  - Standard concatenation of the two local descriptors (MBH).
  - Histogram of Gradients (HoG)
- Fisher encoding of all features independently:
  - 256 Gaussians with diagonal covariance
  - Gradients with respect to means and covariances

Is context relevant for action recognition?

- We extract the central frame of each video as keyframe
- Visualizing the mean keyframe each class is illuminating:

Is context relevant for action recognition?

- We extract the central frame of each video as keyframe
- Visualizing the mean keyframe each class is illuminating:

Basketball

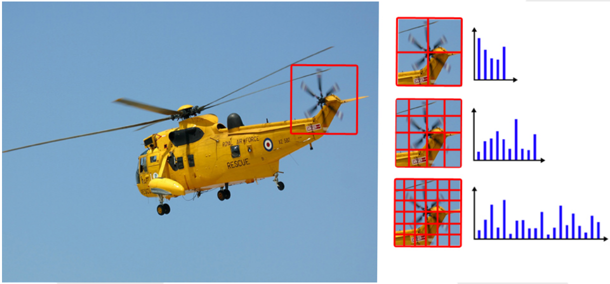
Playing Cello

Ice Dancing

Soccer Penalty

Additional contextual features

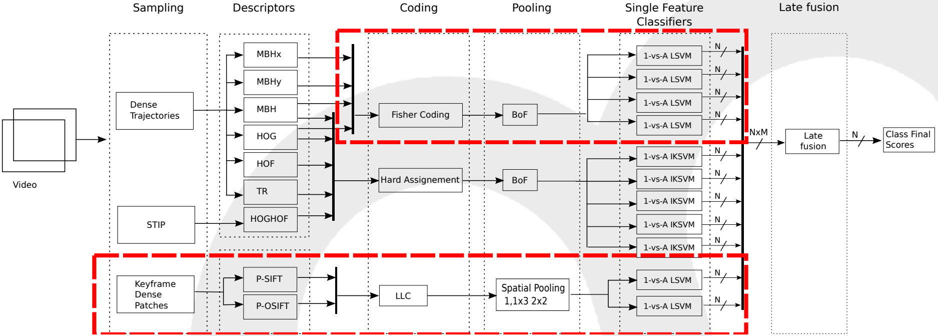
- Dense sampled Pyramidal-SIFT [Seidenari:2013] features (P-SIFT and P-OpponentSIFT) on keyframes
  - Pyramidal-SIFT: three pooling levels, corresponding to 2×2, 4×4, 6×6 pooling regions. Each level has its own dictionary: 1500, 2500 and 3000 words respectively.
  - Spatial pyramid configuration: 1x1, 2x2, 1x3
  - Locality-constrained Linear Coding and max pooling [Wang:2010]



Late fusion with all features (Run-2)

Run-2: more features, better encoding

- The Fisher encoded MBH, MBHx, MBHy, and the LLC encoded P-SIFT and P-OSIFT are fed to Linear 1-vs-all SVMs
- Combined with provided feature histograms: total of $M = 11$ features
- Performance: 82.46% (Split1: 81.47%, Split2: 83.01%, Split3: 82.88%) Run-1: 74.6%



Stacking

- Stacking: learn a classifier on top of the concatenation of expert decisions:

$$S(\mathbf{x}) \quad = \quad [E_i^j], \text{ for } j \in \{1, \ldots M\}, \ i \in \{1, \ldots N\} \qquad (2)$$

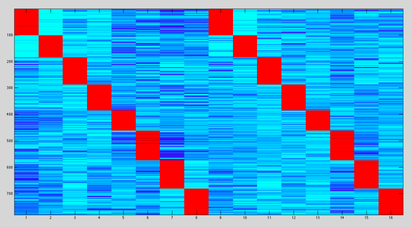
- Having lots of class/feature experts makes THUMOS an excellent playground for this type of fusion approach.
- Our idea: use L1-regularized LR for class/feature expert selection.

Stacking

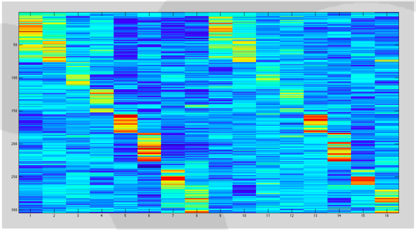
- Stacking: learn a classifier on top of the concatenation of expert decisions:

$$S(\mathbf{x}) \quad = \quad [E_i^j], \text{ for } j \in \{1, \ldots M\}, \; i \in \{1, \ldots N\} \qquad (2)$$

- Having lots of class/feature experts makes THUMOS an excellent playground for this type of fusion approach.
- Our idea: use L1-regularized LR for class/feature expert selection.
- Doing it wrong: decisions values on training samples from classifiers trained on those samples



(a) Train



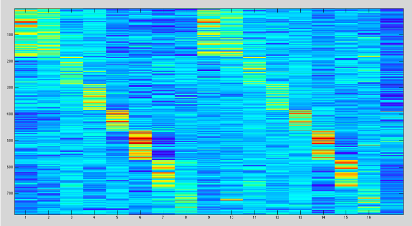
(b) Test

Stacking

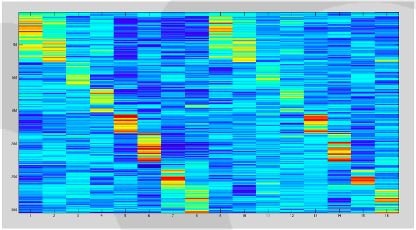
- Stacking: learn a classifier on top of the concatenation of expert decisions:

$$S(\mathbf{x}) \quad = \quad [E_i^j], \text{ for } j \in \{1, \ldots M\}, \; i \in \{1, \ldots N\} \tag{2}$$

- Having lots of class/feature experts makes THUMOS an excellent playground for this type of fusion approach.
- Our idea: use L1-regularized LR for class/feature expert selection.
- Doing it right: reconstruct the decisions on the training samples by running multiple held out training/test folds



(a) Train hold-out



(b) Test

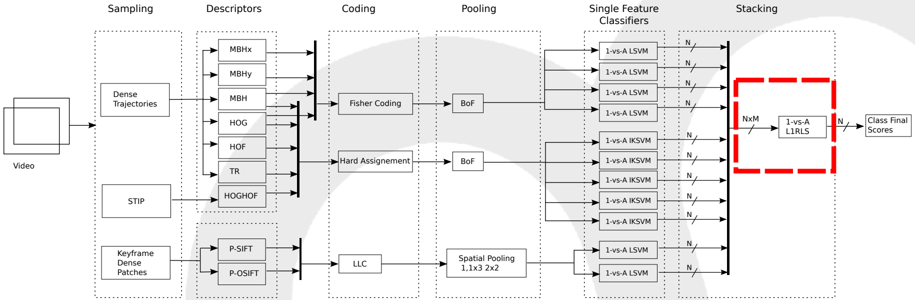
Logistic regression for stacking (Run-3)

Run-3: L1 regularized logistic stacking

- Motivation: smart weighted/selection scheme
- Model $(\boldsymbol{\beta}_c, b_c)$ of class c obtained by minimizing the loss:

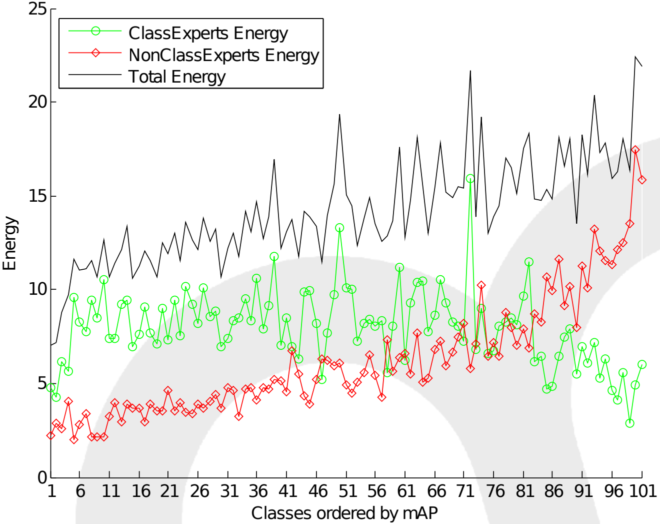
$$(\boldsymbol{\beta}_c, b_c) = \arg\min_{\beta,b} ||\boldsymbol{\beta}||_1 + C \sum_{i=1}^{n} \ln(1 + e^{-y_i \boldsymbol{\beta}^T S(\mathbf{x}_i)+b}) \quad (3)$$

- Performance: 84.44% (Split1: 83.70%, Split2: 85.56%, Split3: 84.07%) Run-2: 82.46%



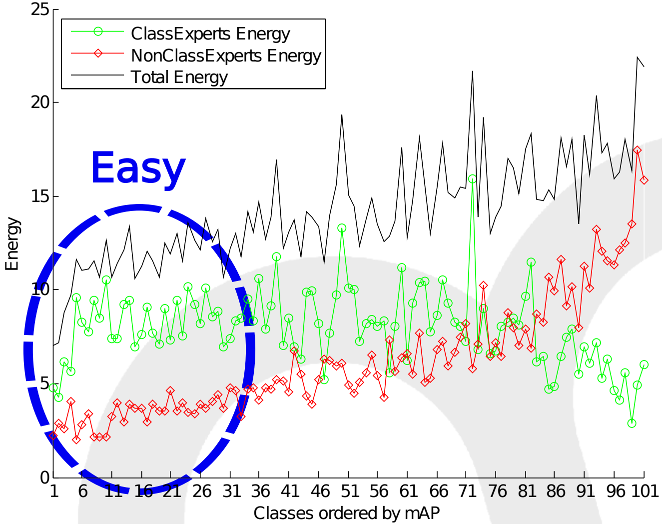
Experts/Non-experts usage analysis

- Analysis: easy/hard classes as mAP of their experts.



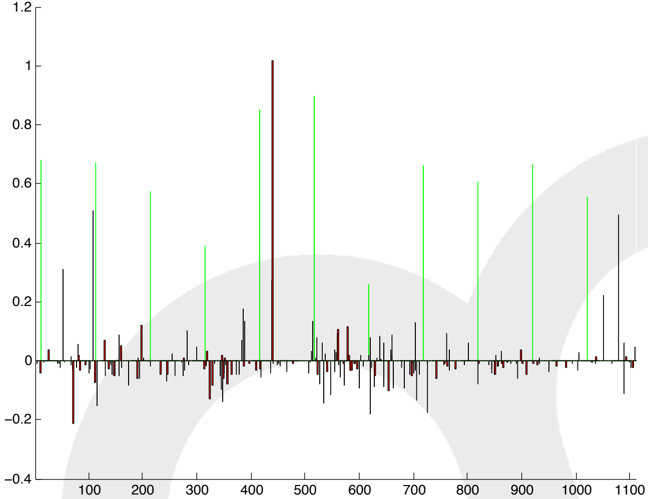
Experts/Non-experts usage analysis

- Easy classes rely more on their own experts, lower total energy



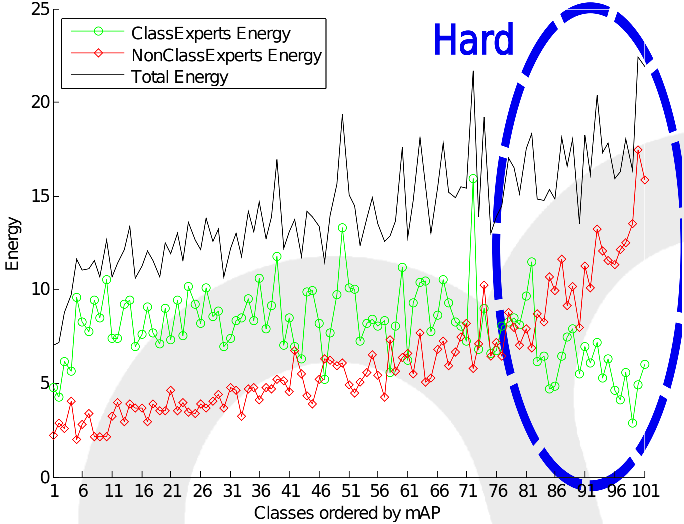
Experts/Non-experts usage analysis

- L1LRS model of "easiest" class: "Billiards"



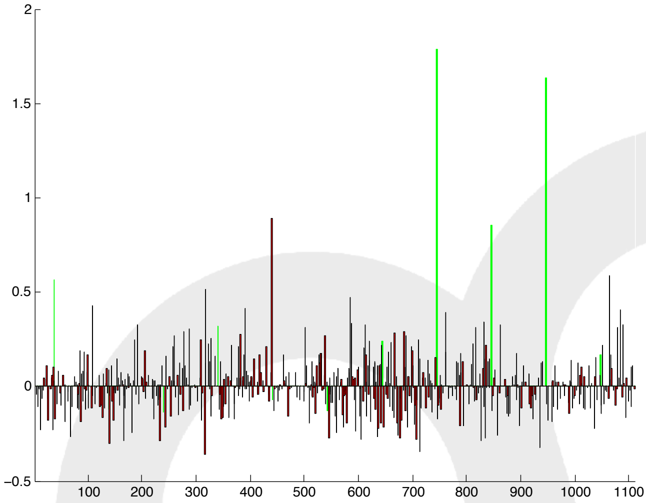
Experts/Non-experts usage analysis

- Hard classes rely more on other classes experts, higher total energy



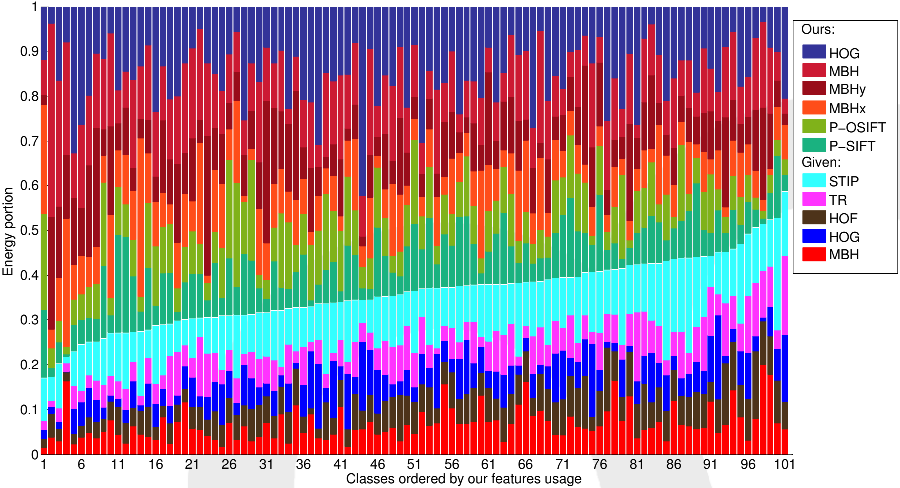
Experts/Non-experts usage analysis

- L1LRS model of "hardest" class: "Handstand Walking"



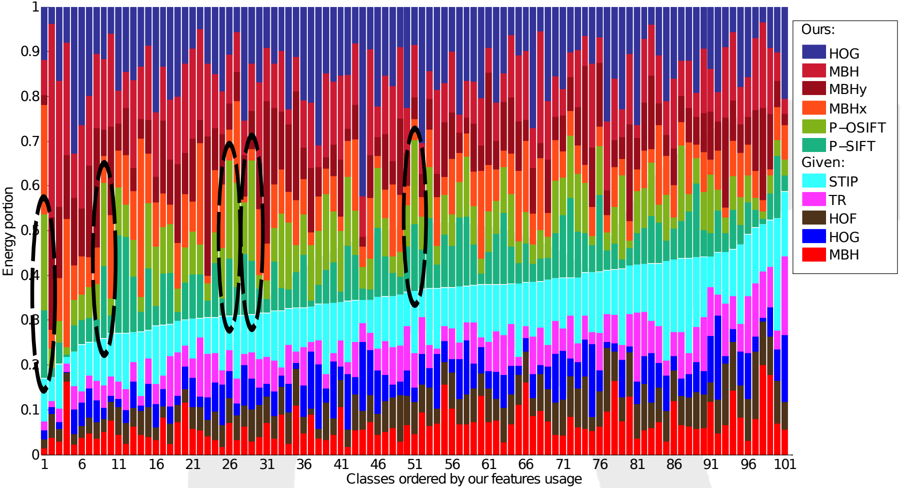
Features/Experts usage analysis

- L1LRS is able to select the most relevant features...



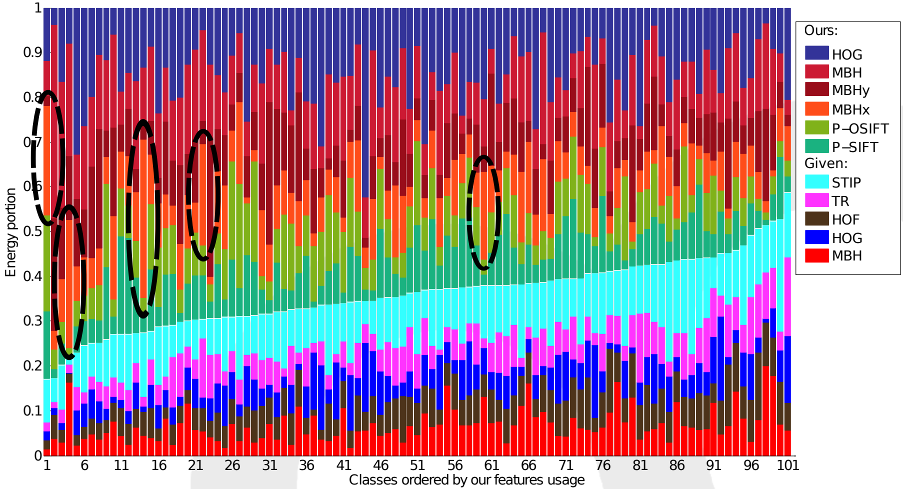
Features/Experts usage analysis

- Classes relying most on contextual features: 1 - "Breaststroke", 26 - "Cutting In Kitchen", 9 - "Field Hockey Penalty", 29 - "Baseball Pitch", 51 - " Playing Piano"



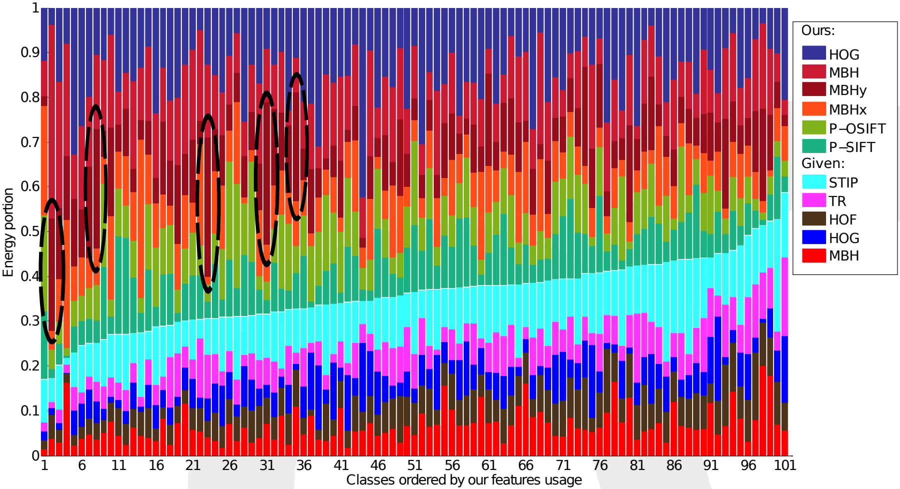
Features/Experts usage analysis

- Classes relying most on MBHx features: 14 - "Hammer Throw", 4 - "Pommel Horse", 1 - "Breaststroke", 22 - "Throw Discus", 60 - "Rowing"



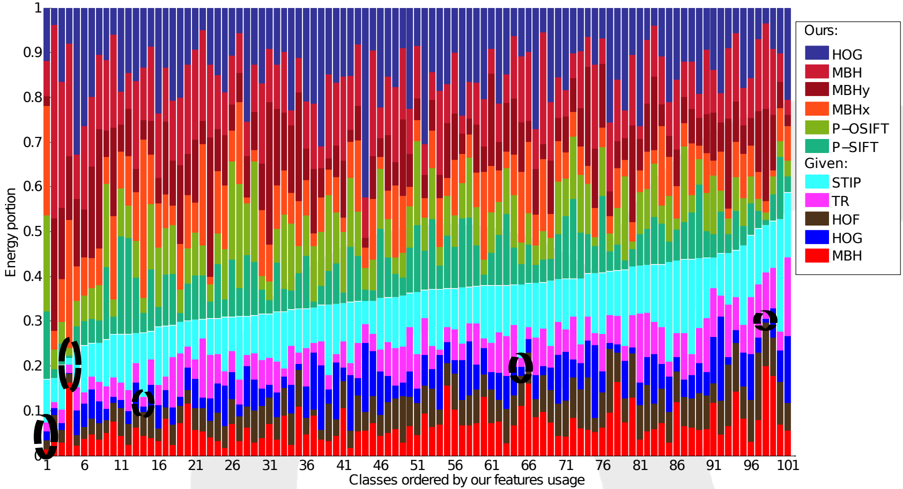
Features/Experts usage analysis

- Classes relying most on MBHy features: 31 - "Soccer Juggling", 23 - "Pole Vault", 8 - "High Jump", 2 - "Body Weight Squats", 35 - "Bench Press"



Features/Experts usage analysis

- ... and L1LRS can also discard the least relevant features



Transductive labelling

- Obtain more consistent labelling using unsupervised local constraints. Previously applied to re-identification [Karaman:2012], first try on another task
- CRF defined as a graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ nodes (all samples) and $\mathcal{E}$ edges of a kNN graph. Energy minimization formulation:

$$W(\hat{\mathbf{c}}) = \sum_{i \in \mathcal{V}} \phi_i(\hat{c}_i) + \lambda \sum_{(v_i, v_j) \in \mathcal{E}} \psi_{ij}(\hat{c}_i, \hat{c}_j), \tag{4}$$

- Data cost uses L1LRS output: $\phi_i(\hat{c}_i) = e^{-\left(\boldsymbol{\beta}_{\hat{c}_i}^T S(\mathbf{x}_i) + b_{\hat{c}_i}\right)}$
- Smoothness cost: $\psi_{ij}(\hat{c}_i, \hat{c}_j) = \psi_{ij}\psi(\hat{c}_i, \hat{c}_j)$
  - Similarities between stacked expert outputs to create and weight the edges of k-NN graph: $\psi_{ij} = \exp\left(-\frac{||S(\mathbf{x}_i) - S(\mathbf{x}_j)||_2}{\sigma_i \sigma_j}\right)$
  - Label cost inversely proportional to confusability between labels: $\psi(\hat{c}_i, \hat{c}_j)$

Transductive labelling (Run-4)

Run-4: the whole shebang

- Energy minimization solved by Graph-Cut [Boykov:2001]
- Performance: 85.71% (Split1: 85.32%, Split2: 86.64%, Split3: 85.16%) Run-3: 84.44%
- Improves labeling of ambiguous samples given similar scores by several classifiers [Karaman:PR]
- Similar training *and* test samples in stacked feature space enable this

	$\mathcal{F}_{\text{org}}$	$\mathcal{F}_{\text{ours}}$	L1LRS	CRF	Accuracy
Run-1	✓				74.6%
Run-2	✓	✓			82.4%
Run-3	✓	✓	✓		84.4%
Run-4	✓	✓	✓	✓	**85.7%**

Table 1 : Summary of our four runs.

Results

#	Participant	Avg.	Split 1	Split 2	Split 3
1	**ID39 INRIA**	**85.900**	84.734	85.862	**87.105**
2	*ID40 Florence*	*85.708*	**85.319**	**86.642**	85.164
3	ID35 Canberra	85.437	*84.761*	*86.367*	*85.183*
4	ID38 CAS-SIAT	84.164	83.515	84.607	84.368
5	ID25 Nanjing	83.979	83.111	84.597	84.229
6	ID34 UCF-BoyrazTappen	82.829	82.640	83.352	82.496
7	ID36 UCSD-MSRA-SJTU	80.895	79.410	81.251	82.025
8	ID28 USC	77.360	76.154	77.704	78.222
9	ID31 NII	73.389	71.102	73.671	75.393
10	ID44 UNITN	70.504	70.446	69.797	71.270

Table 2 : Top 10 results of the challenge.

Discussion

Conclusion

- Better encoding makes a big difference
- Logistic regression for stacking is interesting to leverage the power of several class/features experts
  - automatically adjust sparsity for easy/hard classes
  - select relevant class/features experts
- CRF incorporates local similarity constraints to obtain a more reliable labelling

Future works

- Test logistic regression for stacking with many class/features experts
- Spatial/temporal pooling

References

[Boykov:2001] Y. Boykov, O. Veksler, and R. Zabih.
Fast approximate energy minimization via graph cuts.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222–1239, 2001.

[Karaman:2012] Svebor Karaman and Andrew D Bagdanov.
Identity inference: generalizing person re-identification scenarios.
In *Proceedings of ECCV - Workshops and Demonstrations*, pages 443–452, 2012.

[Karaman:PR] Svebor Karaman, Giuseppe Lisanti, Andrew D. Bagdanov, and Alberto Del Bimbo.
Leveraging local neighborhood topology for large scale person re-identification.
In *Submitted to Pattern Recognition*.

[Seidenari:2013] Lorenzo Seidenari, Giuseppe Serra, Andrew D. Badanov, and Alberto Del Bimbo.
Local pyramidal descriptors for image recognition.
Transactions on Pattern Analisys and Machine Intelligence, in press, 2013.

[Wang:2010] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong.
Locality-constrained linear coding for image classification.
In *Proc. of CVPR*, 2010.

[Wang:2013] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu.
Dense trajectories and motion boundary descriptors for action recognition.
International Journal of Computer Vision, pages 1–20, 2013.

L1-regularized Logistic Regression Stacking and Transductive CRF Smoothing for Action Recognition in Video

Svebor Karaman, Lorenzo Seidenari, Andrew D. Bagdanov,
Alberto Del Bimbo

Media Integration and Communication Center (MICC)
University of Florence, Florence, Italy

{svebor.karaman, lorenzo.seidenari}@unifi.it,
{bagdanov, delbimbo}@dsi.unifi.it

`http://www.micc.unifi.it/vim/people`