Featureless: Bypassing Feature Extraction In Action Categorization

S. L. Pintea*, P. S. Mettes•, J. C. van Gemert*, A. W. Smeulders•

*Computer Vision Lab,
Delft University of Technology, Netherlands

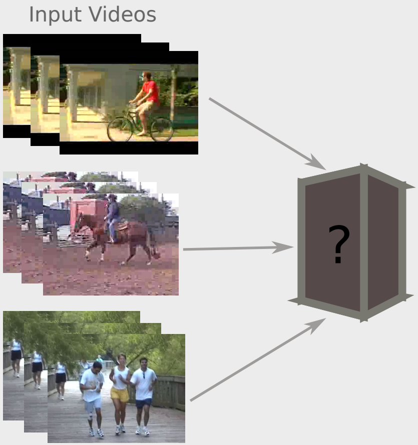
•Intelligent Sensory Information Systems,
University of Amsterdam, Netherlands

What is Action Categorization?

Input Videos

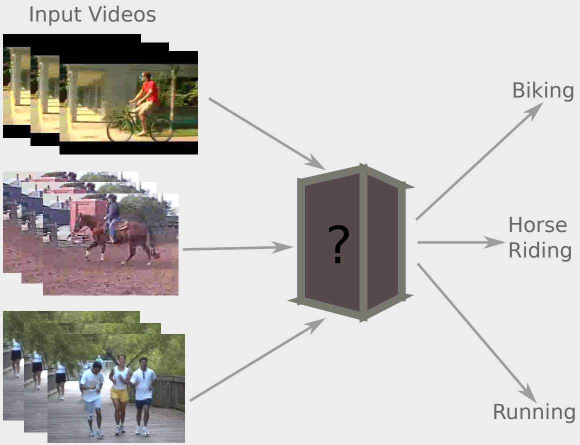
Disregards tasks s.a. action segmentation, action localization.

What is Action Categorization?



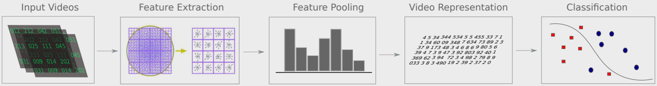
Disregards tasks s.a. action segmentation, action localization.

What is Action Categorization?



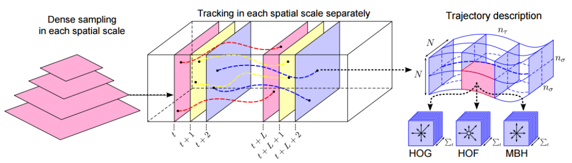
Disregards tasks s.a. action segmentation, action localization.

Standard Classic Approaches

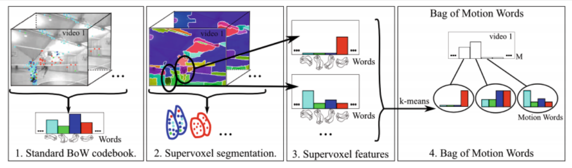


Variations of features over time:

H. Wang, ICCV, 2013.

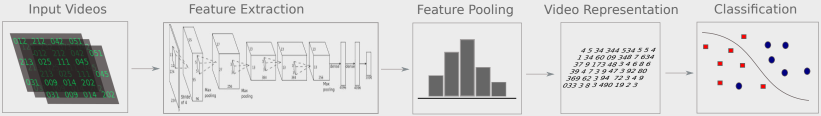


E. Taralova, ECCV, 2014.



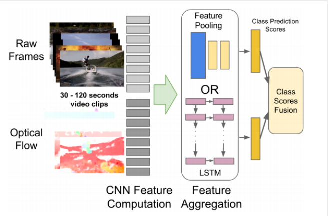
Problem: Feature extraction is slow and takes space to store.

Standard Deep Net Approaches

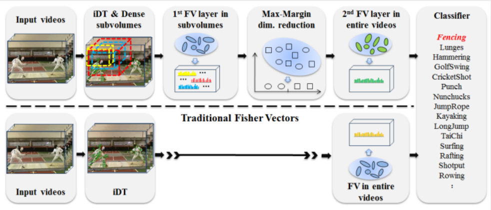


Variations of pooling over frames:

J Yue-Hei Ng, CVPR 2015.



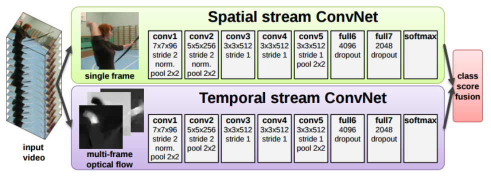
X. Peng, ECCV, 2014.



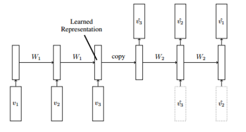
Problem: Still extracts features and aggregates them, but better features.

Deep Learning Approaches

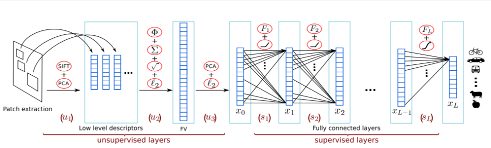
K Simonyan, NIPS 2014.



N Srivastava, JMLR 2015.



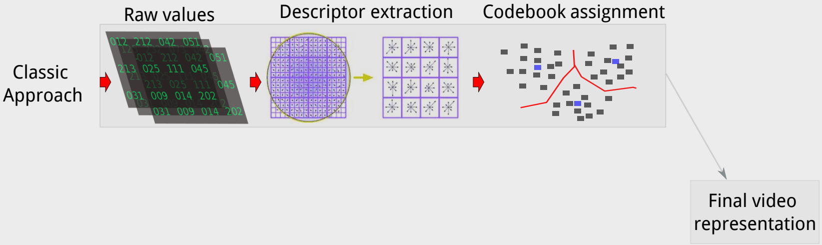
F. Perronnin, Florent, CVPR, 2015.



Slowly bridging the gap in performance.

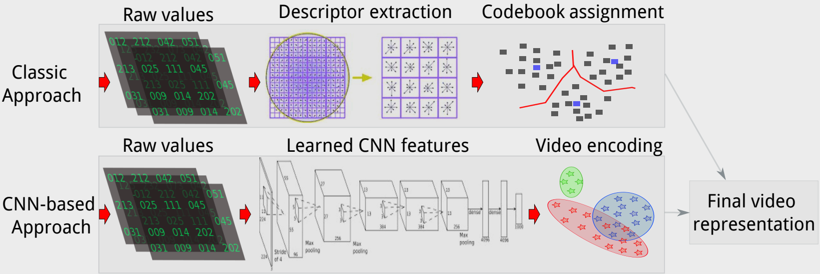
Is Feature Extraction Needed at Test-time?

- Classic: extract handcrafted features and use them in a video representation.



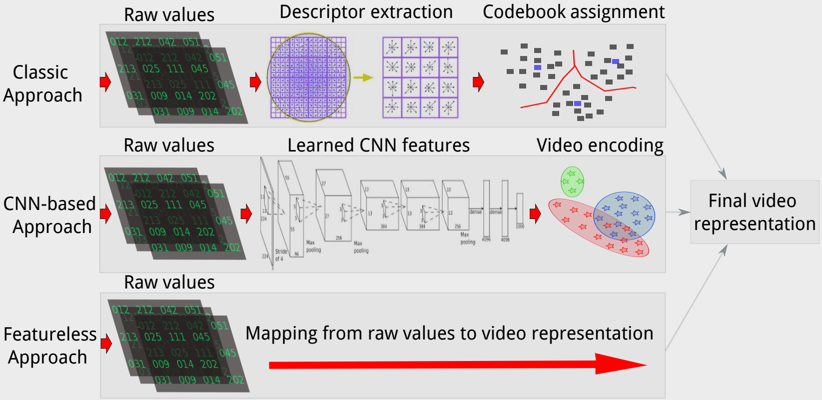
Is Feature Extraction Needed at Test-time?

- CNN-based: extract CNN features and use them in a video representation.



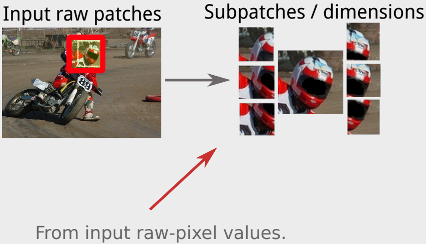
Is Feature Extraction Needed at Test-time?

- Featureless: predict codeword IDs and compute a first-order video representation.



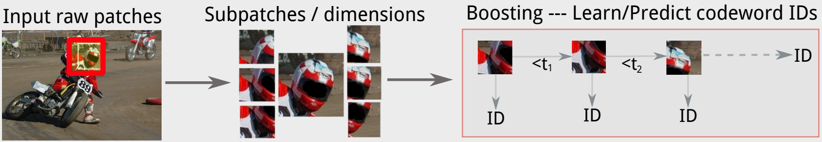
How? — Featureless Boosting

Proof of concept: discard the features and learn their statistics instead.



How? — Featureless Boosting

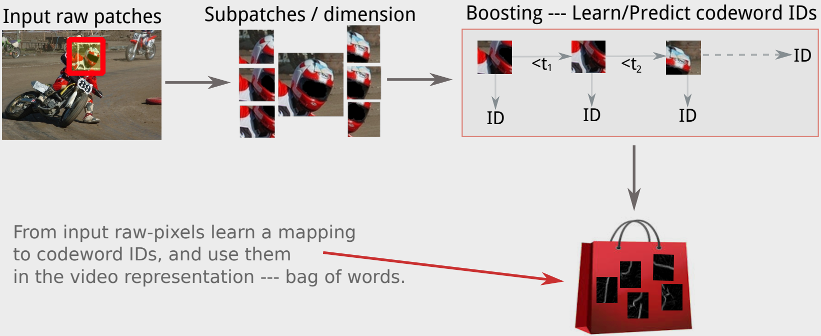
Proof of concept: discard the features and learn their statistics instead.



From input raw-pixels learn
a mapping to codeword ID-s.
(Time Efficient ---> so use Boosting)

How? — Featureless Boosting

Proof of concept: discard the features and learn their statistics instead.



How? — Featureless Boosting

We train the real version of the multiclass Adaboost [J. Zhu, 2009]:

1. *Initialize the observation weights* $w_i = 1/n,\ i = 1, 2, \ldots, n$.
2. *For* $m = 1$ *to* M*:*
   (a) *Fit a classifier* $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i.
   (b) *Obtain the weighted class probability estimates*
   $$p_k^{(m)}(\boldsymbol{x}) = \mathrm{Prob}_w(c = k|\boldsymbol{x}),\ k = 1, \ldots, K.$$
   (c) *Set*
   $$h_k^{(m)}(\boldsymbol{x}) \leftarrow (K-1)\left(\log p_k^{(m)}(\boldsymbol{x}) - \frac{1}{K}\sum_{k'} \log p_{k'}^{(m)}(\boldsymbol{x})\right),\ k = 1, \ldots, K.$$
   (d) *Set*
   $$w_i \leftarrow w_i \cdot \exp\left(-\frac{K-1}{K}\boldsymbol{y}_i^{\mathsf{T}} \log \boldsymbol{p}^{(m)}(\boldsymbol{x}_i)\right),\ i = 1, \ldots, n.$$
   (e) *Re-normalize* w_i.
3. *Output*
$$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} h_k^{(m)}(\boldsymbol{x}).$$

Training data are patch dimensions, labels are codeword IDs.

How? — Featureless Boosting

We train the real version of the multiclass Adaboost [J. Zhu, 2009]:

1. *Initialize the observation weights* $w_i = 1/n,\ i = 1, 2, \ldots, n$.
2. *For* $m = 1$ *to* M*:*
   (a) *Fit a classifier* $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i.
   (b) *Obtain the weighted class probability estimates*
   $$p_k^{(m)}(\boldsymbol{x}) = \mathrm{Prob}_w(c = k|\boldsymbol{x}),\ k = 1, \ldots, K.$$
   (c) *Set*
   $$h_k^{(m)}(\boldsymbol{x}) \leftarrow (K-1)\left(\log p_k^{(m)}(\boldsymbol{x}) - \frac{1}{K}\sum_{k'} \log p_{k'}^{(m)}(\boldsymbol{x})\right),\ k = 1, \ldots, K.$$
   (d) *Set*
   $$w_i \leftarrow w_i \cdot \exp\left(-\frac{K-1}{K}\boldsymbol{y}_i^{\intercal} \log \boldsymbol{p}^{(m)}(\boldsymbol{x}_i)\right),\ i = 1, \ldots, n.$$
   (e) *Re-normalize* w_i.
3. *Output*
   $$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} h_k^{(m)}(\boldsymbol{x}).$$

Multiclass decision trees with probabilistic output.

How? — Featureless Boosting

We train the real version of the multiclass Adaboost [J. Zhu, 2009]:

Decision boundary weighted in each leaf by the sample weight:

$$p_k^m(\mathbf{x}_i) = \frac{\sum_{i\in\mathcal{L}} w_i(y_i^k = 1)}{\sum_{i\in\mathcal{L}} w_i}$$

1. *Initialize the observation weights* $w_i = 1/n,\ i = 1, 2, \ldots, n$.
2. *For* $m = 1$ *to* M*:*
   (a) *Fit a classifier* $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i.
   (b) *Obtain the weighted class probability estimates*
   $$p_k^{(m)}(\boldsymbol{x}) = \mathrm{Prob}_w(c = k|\boldsymbol{x}),\ \ k = 1, \ldots, K.$$
   (c) *Set*
   $$h_k^{(m)}(\boldsymbol{x}) \leftarrow (K-1)\left(\log p_k^{(m)}(\boldsymbol{x}) - \frac{1}{K}\sum_{k'}\log p_{k'}^{(m)}(\boldsymbol{x})\right),\ \ k = 1, \ldots, K.$$
   (d) *Set*
   $$w_i \leftarrow w_i \cdot \exp\left(-\frac{K-1}{K}\boldsymbol{y}_i^{\intercal}\log \boldsymbol{p}^{(m)}(\boldsymbol{x}_i)\right),\ \ i = 1, \ldots, n.$$
   (e) *Re-normalize* w_i.
3. *Output*
$$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} h_k^{(m)}(\boldsymbol{x}).$$

How? — Featureless Boosting

We train the real version of the multiclass Adaboost [J. Zhu, 2009]:

1. Initialize the observation weights $w_i = 1/n,\ i = 1, 2, \ldots, n$.

2. For $m = 1$ *to* M:

(a) Fit a classifier $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i.

(b) Obtain the weighted class probability estimates

$$p_k^{(m)}(\boldsymbol{x}) = \mathrm{Prob}_w(c = k|\boldsymbol{x}),\ k = 1, \ldots, K.$$

(c) Set

$$h_k^{(m)}(\boldsymbol{x}) \leftarrow (K-1)\left(\log p_k^{(m)}(\boldsymbol{x}) - \frac{1}{K}\sum_{k'} \log p_{k'}^{(m)}(\boldsymbol{x})\right),\ k = 1, \ldots, K.$$

(d) Set

Weight updates.

$$w_i \leftarrow w_i \cdot \exp\left(-\frac{K-1}{K}\boldsymbol{y}_i^{\mathsf{T}} \log \boldsymbol{p}^{(m)}(\boldsymbol{x}_i)\right),\ i = 1, \ldots, n.$$

(e) Re-normalize w_i.

3. Output

$$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} h_k^{(m)}(\boldsymbol{x}).$$

How? — Featureless Boosting

We train the real version of the multiclass Adaboost [J. Zhu, 2009]:

1. *Initialize the observation weights* $w_i = 1/n,\ i = 1, 2, \ldots, n$.
2. *For* $m = 1$ *to* M*:*
   (a) *Fit a classifier* $T^{(m)}(\boldsymbol{x})$ *to the training data using weights* w_i.
   (b) *Obtain the weighted class probability estimates*
   $$p_k^{(m)}(\boldsymbol{x}) = \mathrm{Prob}_w(c = k|\boldsymbol{x}),\ k = 1, \ldots, K.$$
   (c) *Set*
   $$h_k^{(m)}(\boldsymbol{x}) \leftarrow (K-1)\left(\log p_k^{(m)}(\boldsymbol{x}) - \frac{1}{K}\sum_{k'} \log p_{k'}^{(m)}(\boldsymbol{x})\right),\ k = 1, \ldots, K.$$
   (d) *Set*
   $$w_i \leftarrow w_i \cdot \exp\left(-\frac{K-1}{K}\boldsymbol{y}_i^{\intercal} \log \boldsymbol{p}^{(m)}(\boldsymbol{x}_i)\right),\ i = 1, \ldots, n.$$
   (e) *Re-normalize* w_i.
3. *Output*
$$C(\boldsymbol{x}) = \arg\max_k \sum_{m=1}^{M} h_k^{(m)}(\boldsymbol{x}).$$

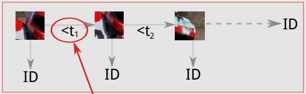
Predicted codeword ID for the input patch.

How? — Featureless Multiclass Waldboost

- Can we make it even faster?

How? — Featureless Multiclass Waldboost

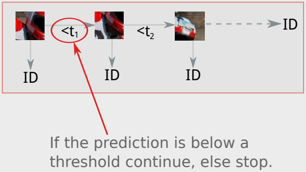
- Can we make it even faster?
- Early stopping — Waldboost [J. Sochman, CVPR, 2005].



If the prediction is below a threshold continue, else stop.

How? — Featureless Multiclass Waldboost

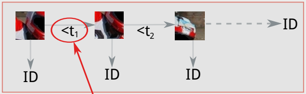
- Can we make it even faster?
- Early stopping — Waldboost [J. Sochman, CVPR, 2005].



- But Waldboost finds stopping thresholds for 2-class Adaboost only.

How? — Featureless Multiclass Waldboost

- Can we make it even faster?
- Early stopping — Waldboost [J. Sochman, CVPR, 2005].



If the prediction is below a
threshold continue, else stop.

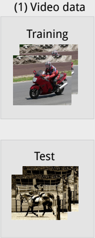
- But Waldboost finds stopping thresholds for 2-class Adaboost only.
- Train on unused training data a **stopping decision tree** that gets as input the prediction of the strong classifier up to now.

$$\max_k \left(\mathrm{Stop}_k^{M'} \left(\sum_{m=1}^{M' \leq M} s_k^m(\bar{\mathbf{x}}_i) \right) \right) \geq \alpha$$

The strong classifier
prediction up to M.

A-to-Z Featureless Approach

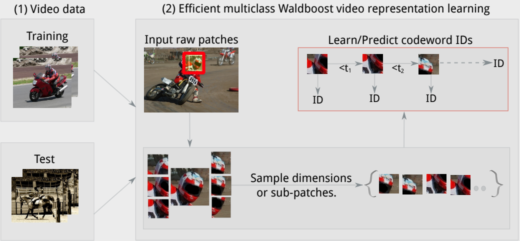
- Get a set of training videos and test videos.



A-to-Z Featureless Approach

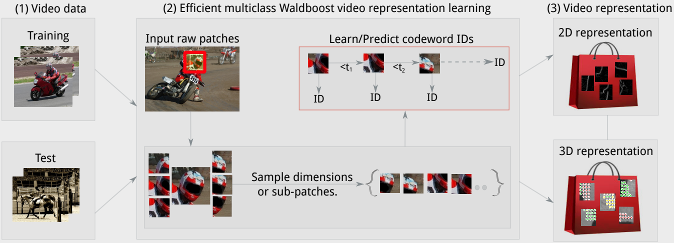
During:

- training: extract (patch, codeword-ID) pairs;
- testing: only patches and predict codeword-IDs in the multiclass Waldboost.



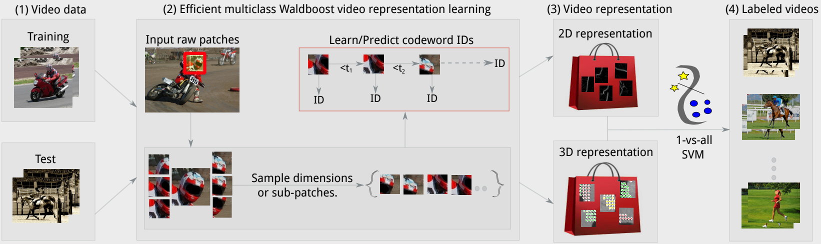
A-to-Z Featureless Approach

- Compute a video representation with the predicted codeword-IDs.



A-to-Z Featureless Approach

- Use the video representation for action categorization.



Experimental Evaluation

Experimental Setup:

- UCF11 dataset. [J. Liu, CVPR, 2009]
- Codebook with K-means over 100 K descriptors.
- Gray-scale patches of 24x24 dimensions.
- 1000 weak classifiers, trained on 24 random dimensions.
- Stopping threshold α set to .97.

Experimental Evaluation

Waldboost word prediction versus other learning algorithms

- Setup:
  - 100 dimensional codebook.
  - HOG descriptors for codebook construction.
- Results:

	Linear SVM	Adaboost	Waldboost
MAP	16%	41%	**41**%
Time/frame	15.00 sec	4.00 sec	0.60 sec

Experimental Evaluation

Learning versus feature extraction

- Setup:
  - 100 dimensional codebook for HOG, HOF.
  - 1000 dimensional codebook for $3D$-HOF.
  - $3D$-HOG descriptors over 8 frames.
- Results:

	HOG		HOF	
	BOW	Waldboost	BOW	Waldboost
MAP	**44%**	41%	**37%**	32%

$3D$ HOF		
BOW	Waldboost	BOW & Waldboost
45%	36%	**50%**

Experimental Evaluation

Learning featureless and codebookless representations

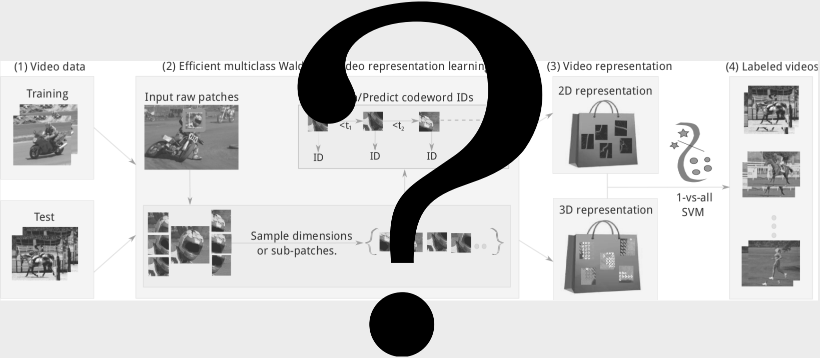
- Setup:
  - From the 100 K patches each is considered to be a data center.
  - Only ≈ 100 patches have test-time patches assigned to them.
- Results:

	BOW	Codebookless	
		Adaboost	Waldboost
MAP	44%	41%	37%

Conclusions

- Present a proof of concept showing that we can bypass feature extraction.
- Still obtain comparable performance with the representation we learn from.
- To this end, a straightforward Waldboost multiclass approach is proposed.
- Finally, we consider both featureless and codebookless representations.

Thank you


(1) Video data
Training
Test
Input raw patches
ID
ID
ID
Sample dimensions
or sub-patches.
(3) Video representation
2D representation
3D representation
1-vs-all
SVM
(4) Labeled videos