Computer Vision Lab
Bitdefender Bucharest

Elena Burceanu
eburceanu@bitdefender.com

Emanuela Haller
ehaller@bitdefender.com

Iulia Duță
iduta@bitdefender.com

Andrei Nicolicioiu
anicolicioiu@bitdefender.com

Marius Leordeanu
marius.leordeanu@cs.pub.ro

UNIVERSITATEA DIN BUCUREȘTI
VIRTUTE ET SAPIENTIA

Bitdefender

Our Focus

We want to better **Understand Video**

Observations regarding current models and data:

- Deeply rooted in **traditional frame by frame** video processing
  - **Aggregate** in **time** over individual frame representation

Our approach: **space**-**time** connections in **earlier stages, graph** architecture

- A huge **quantity of unlabelled data**
  - Labelling is expensive

Our approach: **space**-**time** video **consistency** as a **self-supervision** signal

Our Projects

All of our models can be designed as **space-time graphs**

1. **Unsupervised Object Discovery** in **Video**
   - Based on learning from **Highly Probable Positives (HPPs)**

2. **Object Tracking**
   - Tracking models both **robust** and **adaptive,** also learning from **HPPs**
   - Refine the per frame **segmentation masks** using **time**

3. **Recurrent Space-time Graphs** for **Video Understanding**
   - A general **graph** based computational model for **video processing**
   - This **architecture** is **specifically designed** for **addressing space-time tasks** (classification, detection, segmentation, video captioning, action recognition)

1. Unsupervised Object Discovery in Video

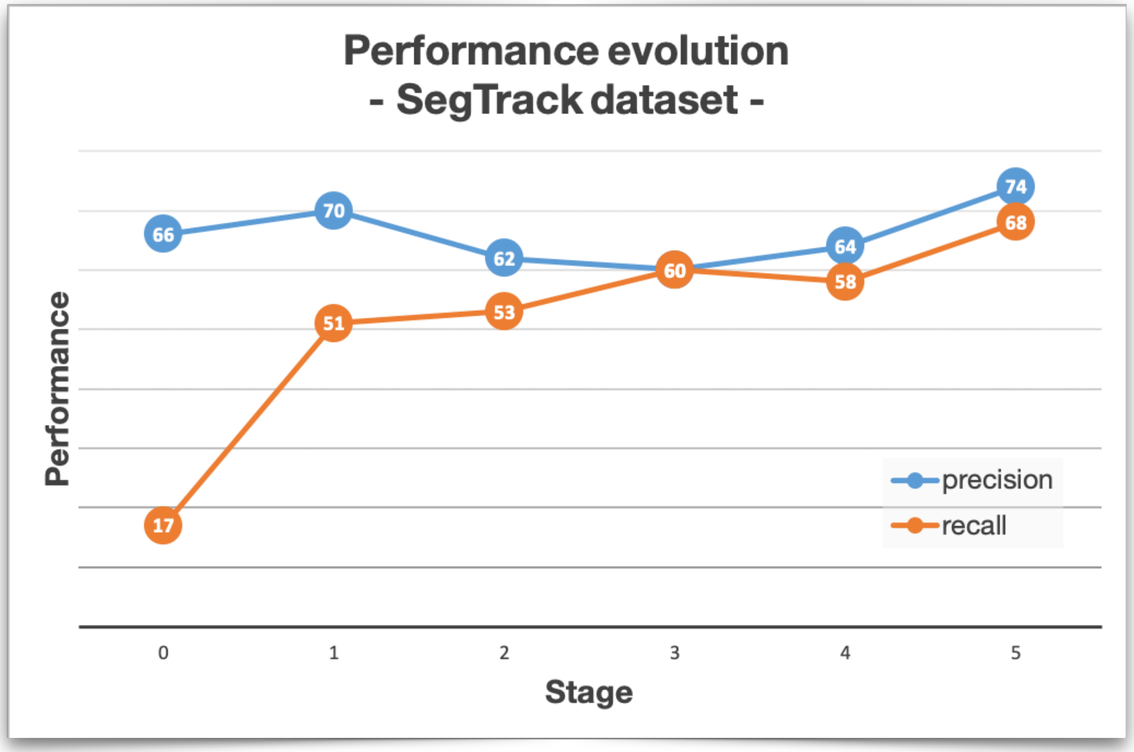
Emanuela Haller and Marius Leordeanu

Efficient selection of HPP features

1. Exploit *foreground-background* **contrasting properties**
   - Efficiently select **Highly Probable Positives** (*foreground* **pixels** with high precision)
   - Use them as supervision in the next step
   - Simple features: colors and motion cues

2. Model created over several stages
   - **Keep** the **precision high (HPPs)** and **increase recall**

3. SOTA results over both supervised and unsupervised solutions, **one order of magnitude faster** than competitors



1. Unsupervised Object Discovery in Video

[E. Haller and M. Leordeanu, ICCV 2017]

Bitdefender

Results

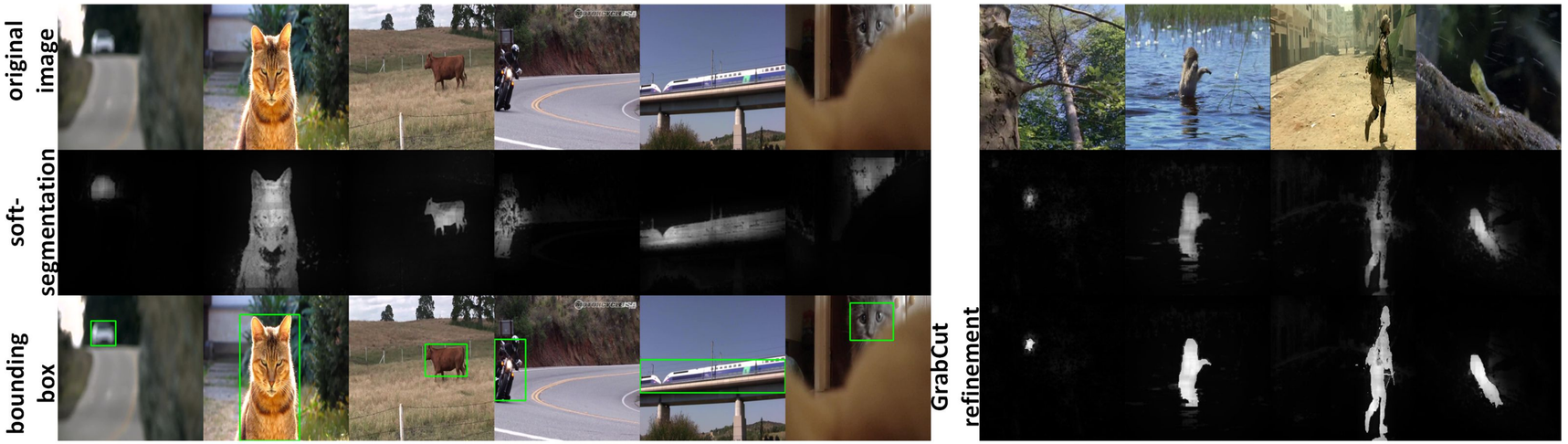
	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	Avg
[Haller and Leordeanu, ICCV 2017]	**76.3**	**71.4**	**65.0**	58.9	**68.0**	**55.9**	**70.6**	33.3	**69.7**	42.4	**61.1**
[Stretcu and Leordeanu, BMVC 2015]	38.3	62.5	51.1	54.9	64.3	52.9	44.3	43.8	41.9	**45.8**	49.9
[Papazoglou and Ferraru, ICCV 2013]	65.4	67.3	38.9	**65.2**	46.3	40.2	65.3	**48.4**	39.0	25.0	50.1
[Prest et al., CVPR 2012]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	**48.4**	39.0	25.0	30.4
[Jun et al., CVPR 2016]	64.3	63.2	73.3	68.9	44.4	62.5	71.4	52.3	78.6	23.1	60.2
[Zhang et al., CVPR 2015]	75.8	60.8	43.7	71.1	46.5	54.6	55.5	54.9	42.4	35.8	54.1

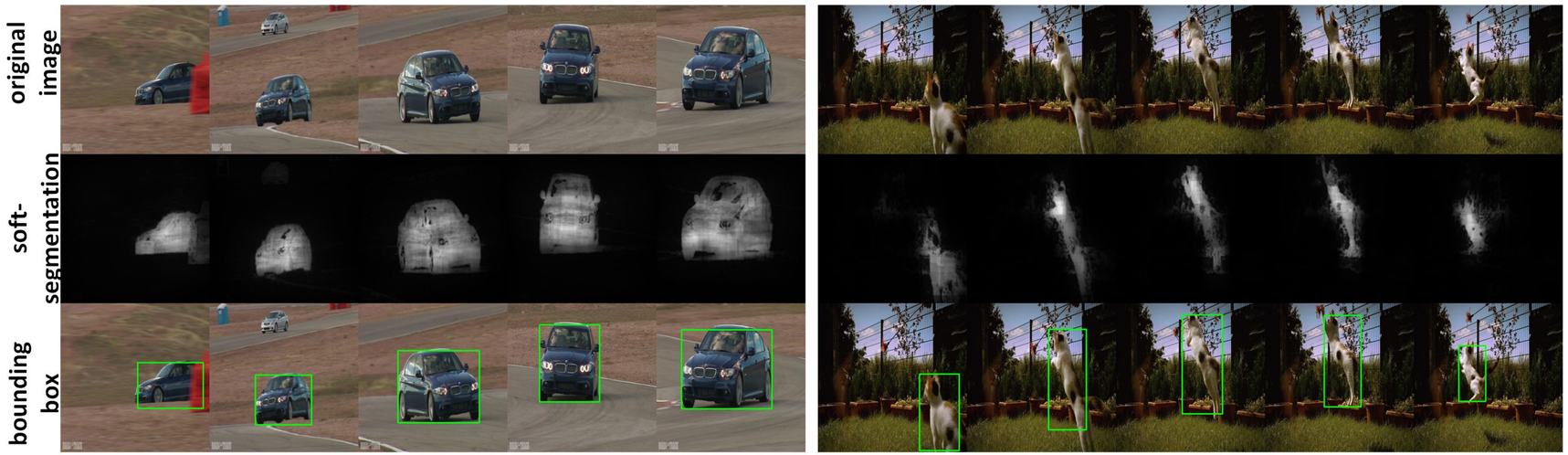
- YouTube-Objects dataset (bbox prediction)
- 4 unsupervised solutions (best in **bold**)
- 2 supervised solutions (best underlined)
- Mean IoU score

Qualitative Results

YouTube-Objects Dataset

SegTrack Dataset





1. Unsupervised Object Discovery in Video

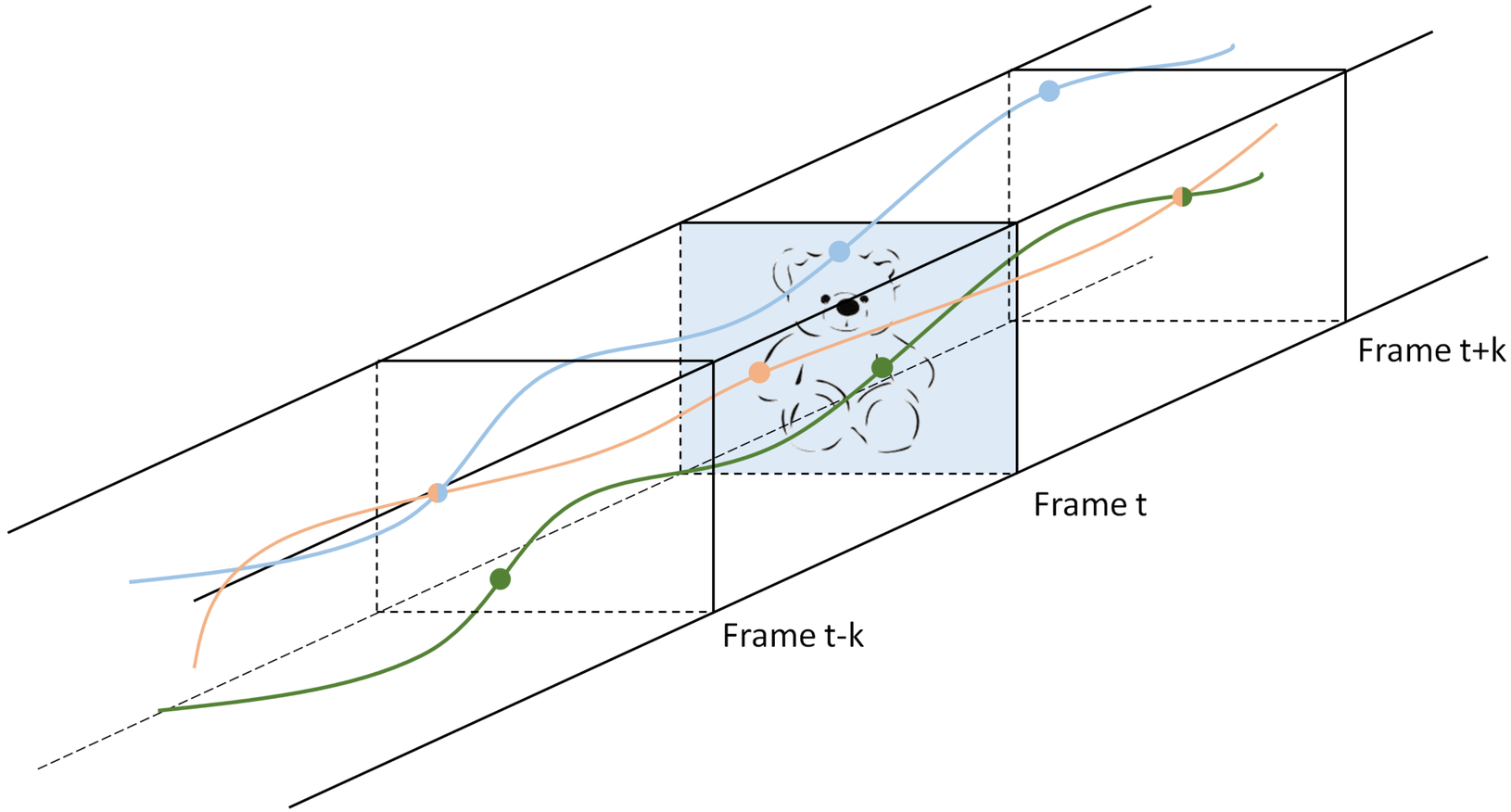
[E. Haller and M. Leordeanu, ICCV 2017]

What's next?

1. Consider more **complex motion models** - **Optical Flow Chains**
2. Further exploit **spatial-temporal** inherent **consistencies** and **co-occurences**
3. Adapt to **shorter** and **harder video sequences** (like the ones in DAVIS)

Bitdefender

Optical Flow Chains



Optical Flow Chains

Refine object **segmentation masks** by **following pixels** along the **Optical Flow**

Step 1 - pixel based (unary)

- Why? **pixels** along one **chain** should have **similar labels**
- How? **propagate foreground probabilities** along the **chains**
- Voting scheme

Step 2 - relations between pixels (pairwise)

- Why? pixels that **move in similar ways** should have **similar labels**
- What **moves together belongs together**
- How? each pixel is described by its left and right **directions** along the **flow**
- Regression model

Iterative formulation

one iteration

$$\begin{cases} y^{(iter,1)} \leftarrow My^{(iter-1,2)}, & \textbf{power iteration step} \\ M(i,j) = is\_chain(i,j)k(i,j), & \textbf{similarity between } i,j \textbf{ pixels} \\ y^{(iter,2)} \leftarrow regress(flow\_descriptor, y^{(iter,1)}), & \textbf{projection over the solution space} \end{cases}$$

$y^{(iter,s)}$ **- foreground map**

M **- adjacency matrix of the graph**

k **- kernel distance function**

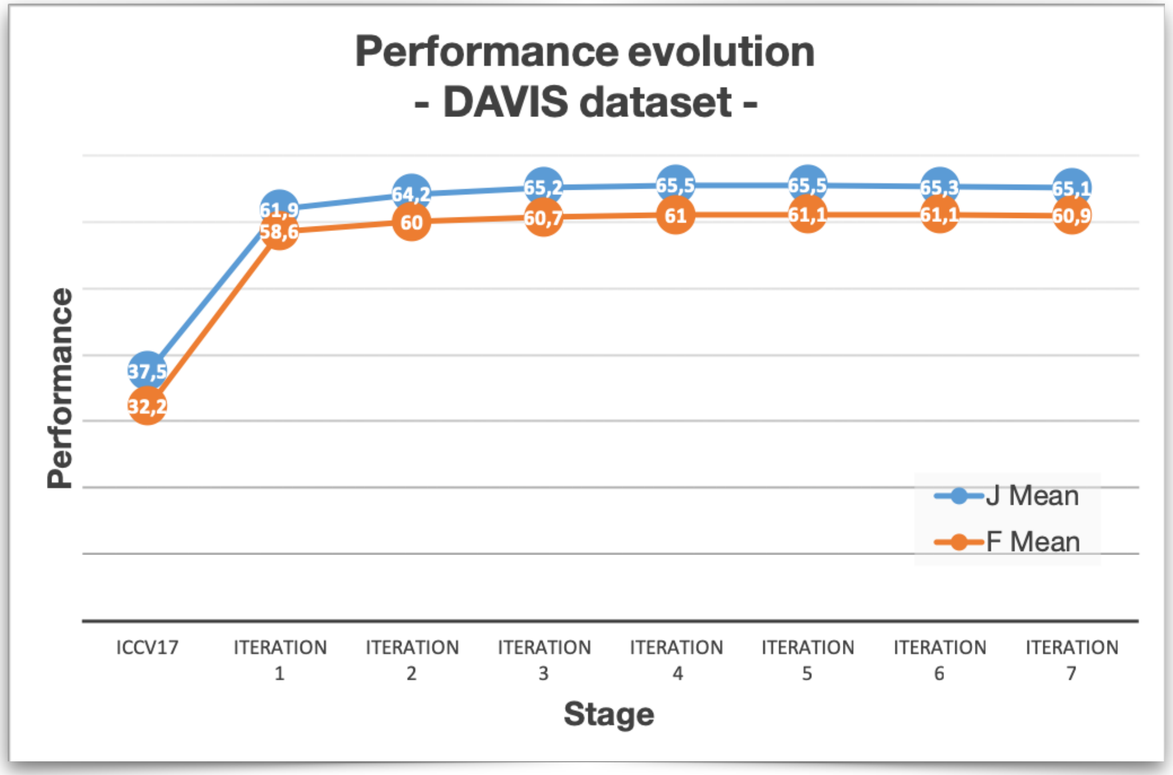
$$is\_chain(i,j) = \begin{cases} 1 & \textbf{if there is a path between } j \textbf{ and } i, \textbf{ along a flow chain} \\ 0 & \textbf{otherwise} \end{cases}$$

$regress(i)$ **- regression function for a faster convergence**

$flow\_descriptor(i)$ **- left and right flow displacemnets for pixel** i

Preliminary Results

	J Mean	F Mean
[Haller and Leordeanu, ICCV 2017]	37.5	32.2
[Haller and Leordeanu - New]	**65.5**	61
[Lao and Sundaramoorthi, ECCV 2018]	61.8	61.2
[Papazoglou and Ferrari, ICCV 2013]	55.8	51.1
[Keuper et al., ICCV 2015]	55.2	55.2
[Faktor and Irani, BMVC 2014]	55.1	52.3

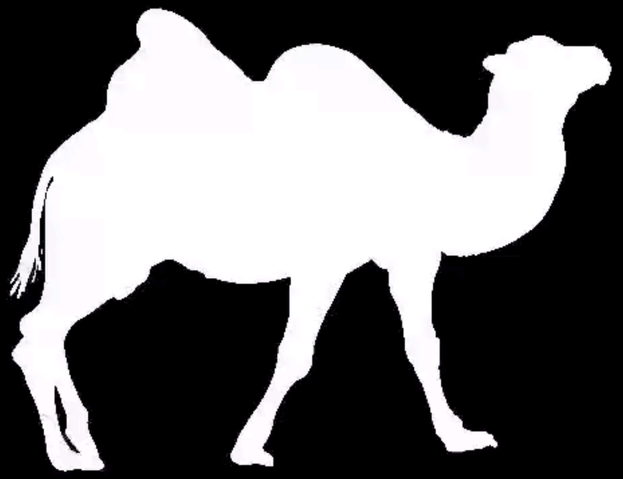


- DAVIS dataset
  - **Fully unsupervised** solutions

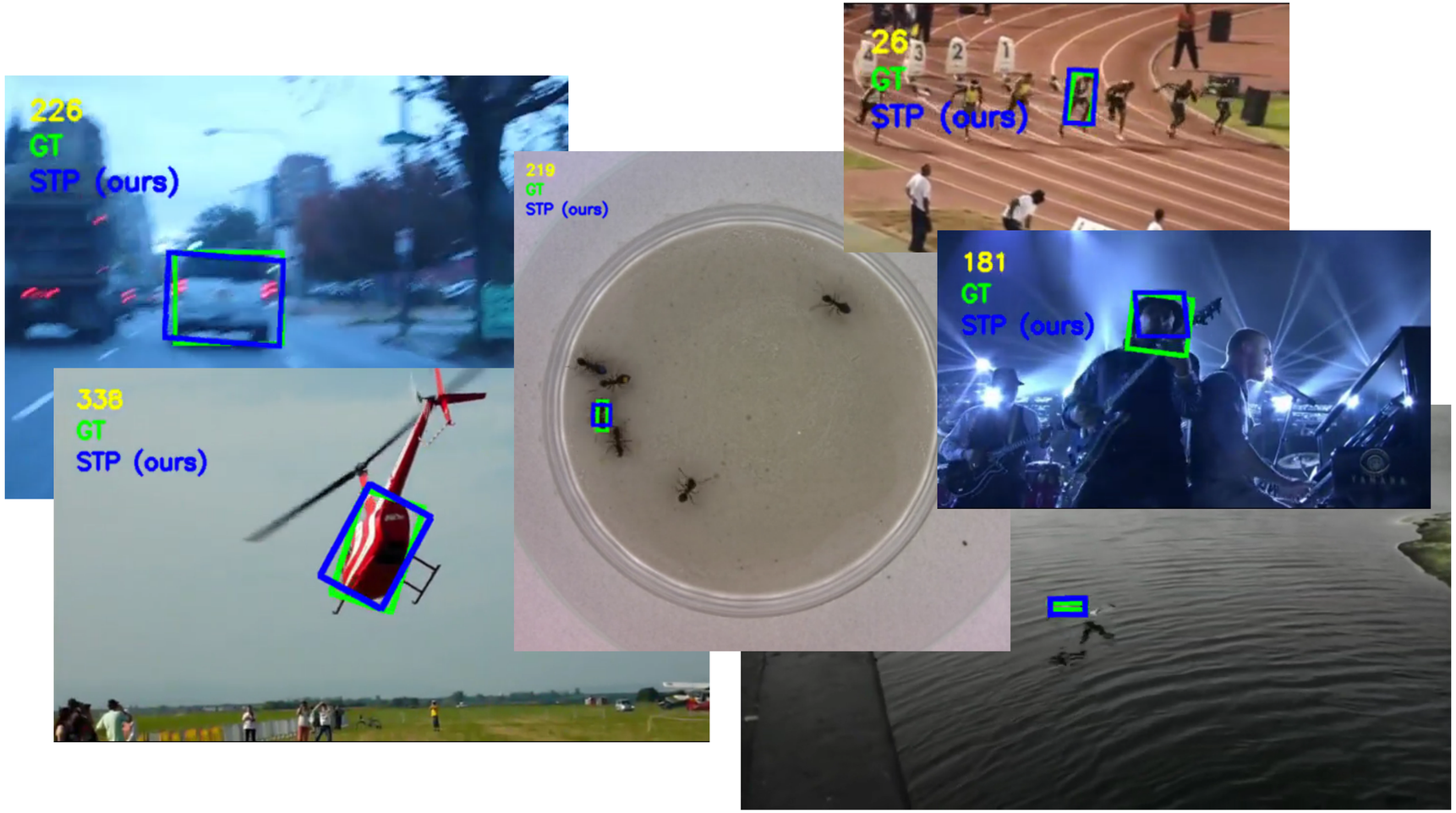
original image

[Haller and Leordeanu – ICCV 2017]

ground truth



[Haller and Leordeanu – New]



2. Object Tracking

Elena Burceanu and Marius Leordeanu

Bitdefender

Our Approach

1. **Society of Tracking Parts (STP)** combines votes from two pathways: **FilterParts pathway** (robust and conservative) with a more adaptive one, the **ConvNetPart pathway**.

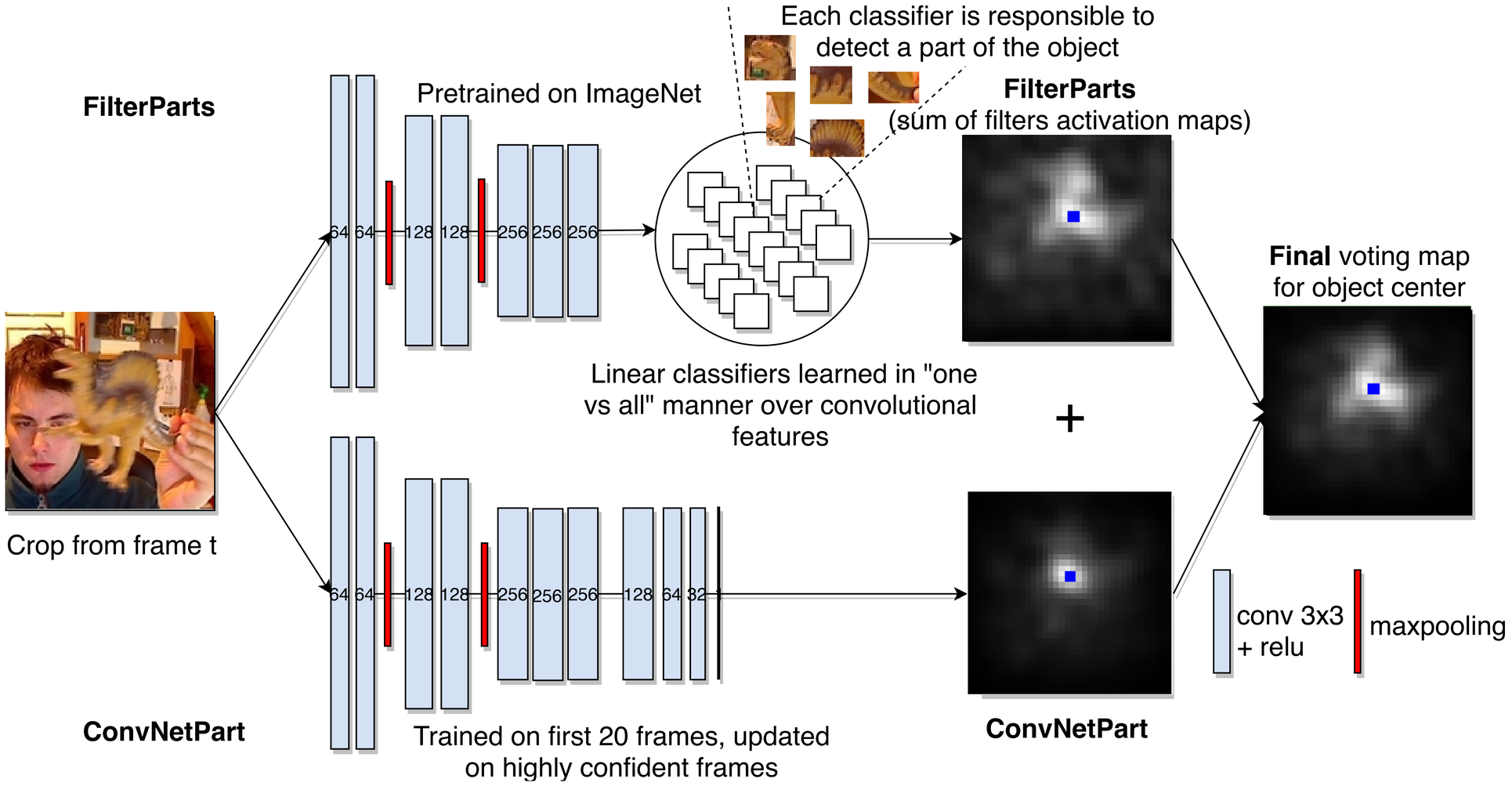
2. **FilterParts pathway**: Parts play different **reliability roles** in voting, based on their value proved in time.

3. **ConvNetPart pathway** uses **co-occurrences guided supervision**. Model is updated online by training only on Highly Confident Frames.

4. Mathematical novelty: **Efficient strategy** for learning **many "one vs all"** **FilterParts** classifiers.

 [E. Burceanu and M. Leordeanu, ECCV VOT Workshop 2018]

Society of Tracking Parts (STP)



 [E. Burceanu and M. Leordeanu, ECCV VOT Workshop 2018]

Bitdefender

STP Pathways

- **FilterParts** pathway
  - Parts differ by **size**, **location** and **reliability** role
  - **Promote** parts if **consistently in agreement** with the final vote
  - Parts have different roles: **candidate**, **reliable** and **golden**

- **ConvNetPart** pathway
  - **Robust adaptability** to change
  - Updates based on **co-occurrences** between pathways
  - Online training on **Highly Confident Frames** - frames where the two complementary pathways agree

 [E. Burceanu and M. Leordeanu, ECCV VOT Workshop 2018]

Fast learning of many "one sample vs all" classifiers

- Weighted least squares needs **n matrix inversions** (expensive)

- **Theoretical novelty**:
  - Our solution requires **a single matrix inversion** for computing **all classifiers simultaneously**
  - Weighted solution has the **same direction**, but a **different magnitude**:

$$c_{weighted} = \frac{n}{1+(n-1)\mathbf{d}_i^\top \mathbf{c}_i} * c_i$$

- Additional optimisation: **Matrix Inversion Lemma** to invert a two orders of magnitude smaller matrix

[E. Burceanu and M. Leordeanu, ECCV VOT Workshop 2018]

State of the Art on VOT17

- **EAO**: 1st on VOT17, 3rd on VOT16 (with the **same** set of **hyper-parameters**)
- **Robustness**: STP outperforms others by a large margin
- **Difficult cases**: 1st on **occlusion** - the most difficult case
  - we also have top results on all the other difficult cases (camera motion, illumination change, motion change, size change)

Tracker (VOT17)	EAO %	R↓	A↑
CFWCR [2]	30.3	1.2	0.48
ECO [3]	28	1.13	0.48
CCOT [4]	26.7	1.31	**0.49**
STP (ours)	**30.9**	**0.765**	0.44

[E. Burceanu and M. Leordeanu, ECCV VOT Workshop 2018]

Bitdefender

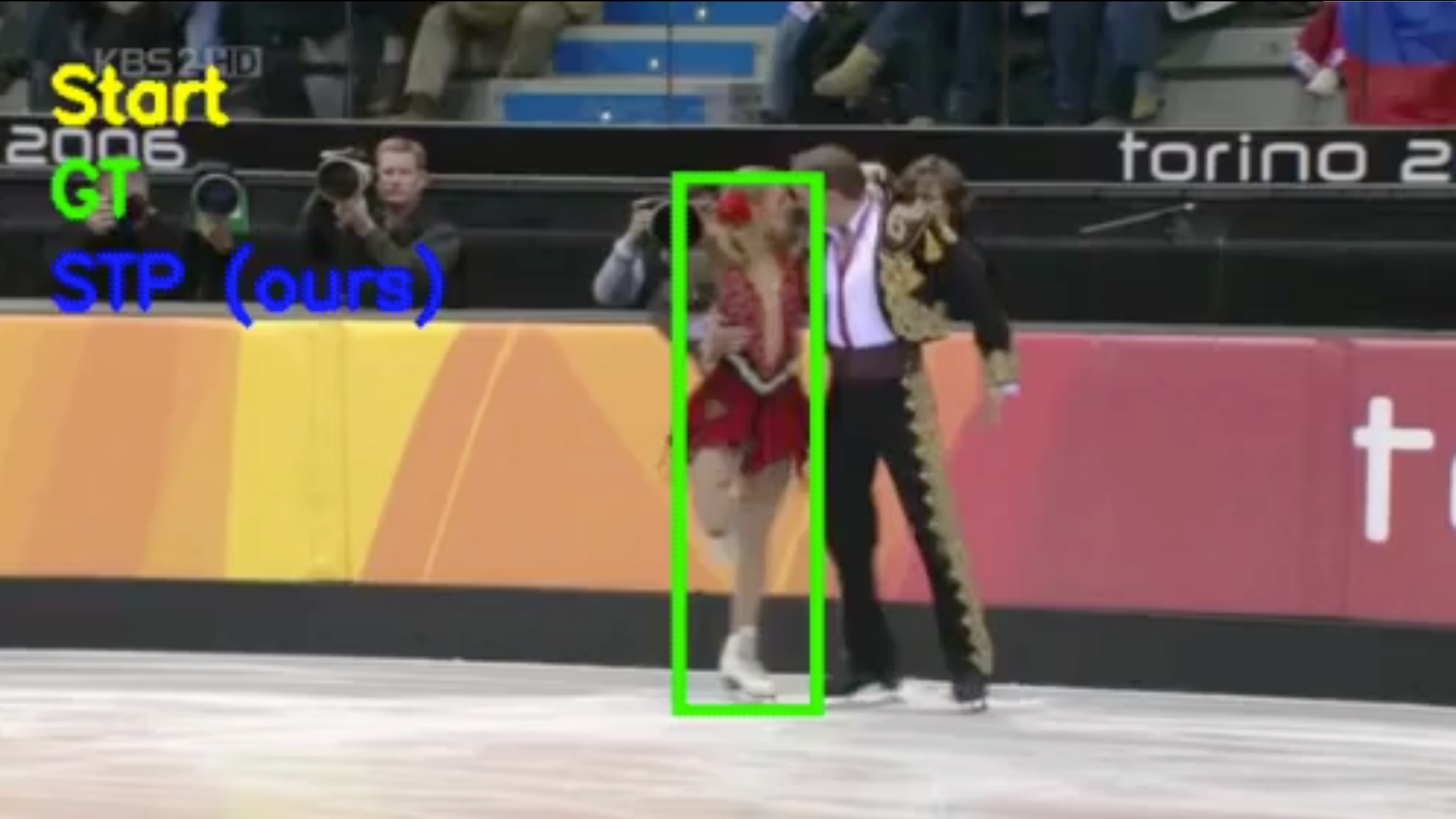
Ablation study

1. **Combining** the two **complementary pathways**
   - 6% boost in EAO
   - compared versions: “**FilterParts** pathway”, “**ConvNetPart** pathway”, “**both pathways**”

2. **Reliability roles** when voting
   - 4% boost in EAO
   - compared versions: “**only reliable parts vote**”, “all parts voting”

3. **Update only on Highly Confident Frames**
   - 2.5% boost in EAO
   - compared versions: “**HCF update**”, “no update”, “all frames update”

 [E. Burceanu and M. Leordeanu, ECCV VOT Workshop 2018]


KBS 2 HD
Start
2006
GT
torino 2
STP (ours)

What’s next? Segmentation

We don’t keep a segmentation model for the object

- Refine **per frame** segmentation masks
  - take into account the **temporal** **consistency**
  - can be **applied** over **any image segmentation** solution

- Formulate the **segmentation task** as a **graph problem**
  - Nodes -> pixels (>20M nodes for a DAVIS-2016 video)
  - Connections -> local relations between pixels (sparse adjacency matrix)
  - Still very expensive operations

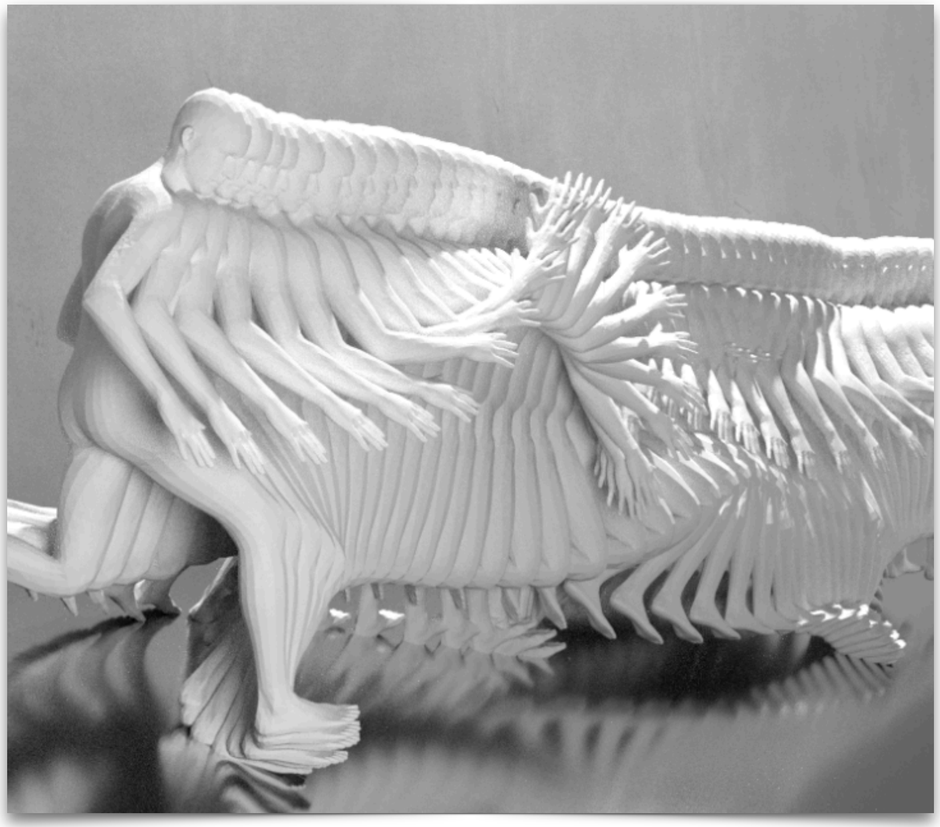
Video as a Graph of Pixels

A slice from the **space**-**time** volume is a **segmentation mask for one frame**

- See how the **time** **consistency** can **refine** the mask

The **strongest cluster** in **space**-**time**

- Find it with a spectral method
- **Eigenvector** of the **adjacency matrix** of the graph
- This is the **refined segmentation**



MoSculp [Zhang et al., UIST 2018]

Video as a Graph of Pixels

one iteration

$$\begin{cases} y^{iter} \leftarrow My^{iter-1}, & \textbf{power iteration step} \\ M(i,j) = c_i^p c_j^p [N - |f_i - f_j|_2^2], & \textbf{unitary and pairwise (similarity) terms} \\ testing: y^{iter} \leftarrow ipfp(y^{iter}), & \textbf{discretization step} \end{cases}$$

y^{iter} **- foreground map**

M **- adjacency matrix of the graph, with only local connections**

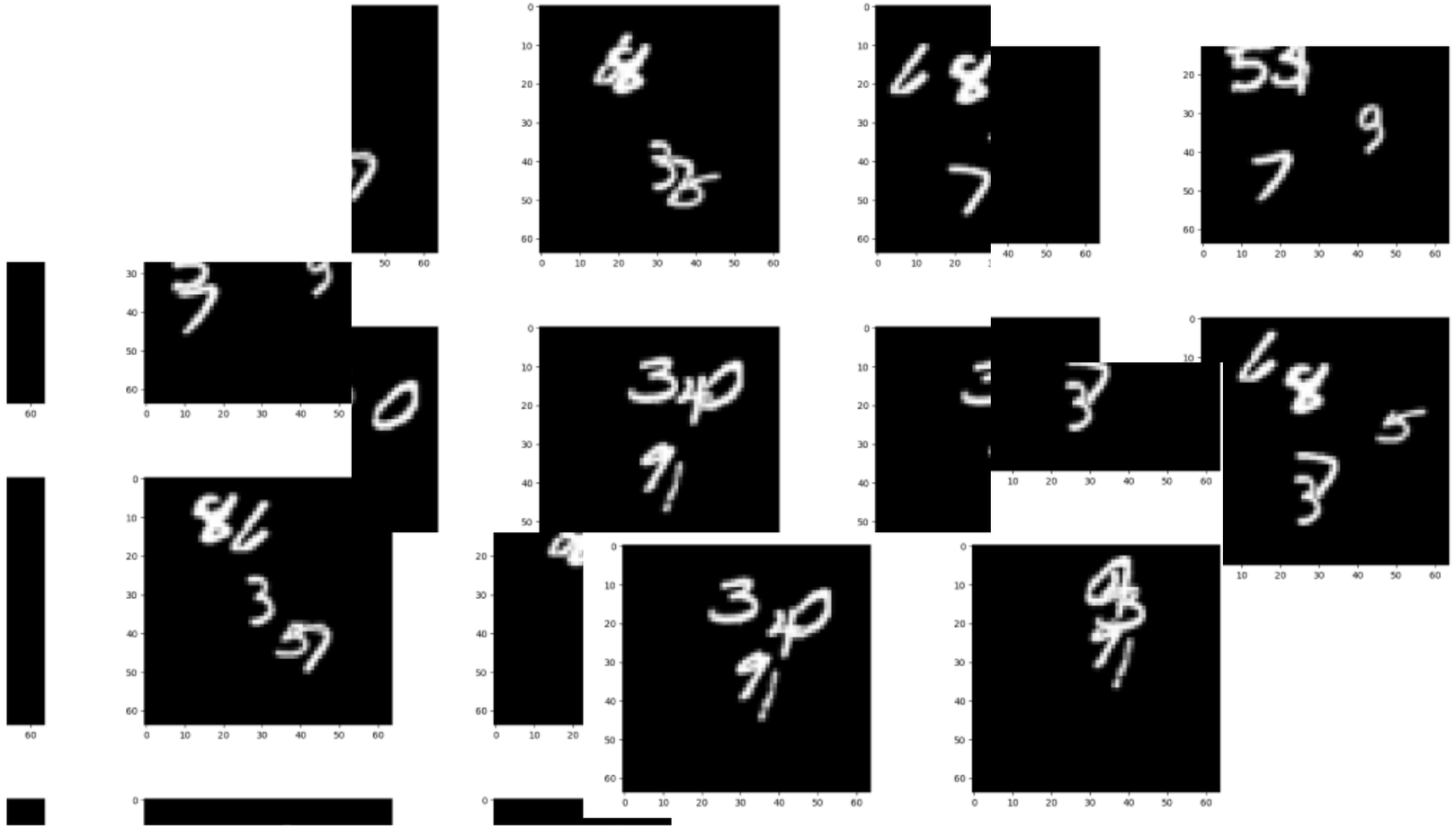
c_i **- initial foreground map for pixel** i

f_i **- N features for pixel** i **(e.g. gray level, optical flow, edges)**

$ipfp$ **- algo for faster power iteration convergence***

- Rewrite it using only several **convolutions** with **simple** filters (very **efficient**)
- Introduce a **custom neural net layer**
  - Learn new filters
  - Inspired from power iteration equations
- **Preliminary results**: maximum 5% (segmentations over DAVIS-2016)

* **I**nteger **P**rojected **F**ixed **P**oint **[M. Leordeanu et al. IJCV 2012]**

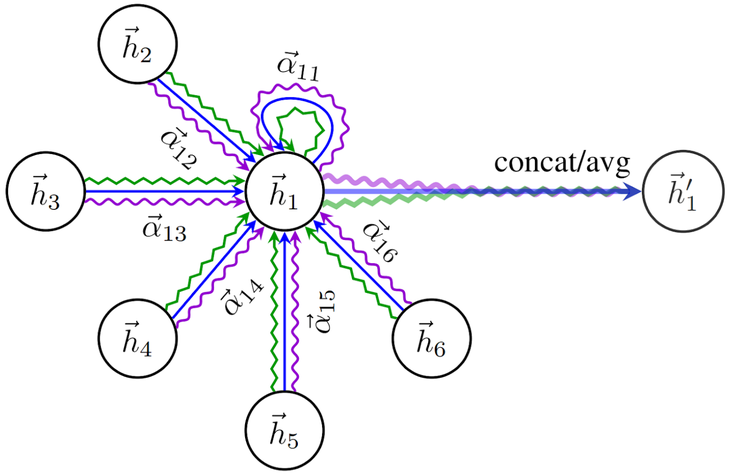


3. Recurrent Space-time Graphs for Video Understanding

Iulia Duță, Andrei Nicolicioiu and Marius Leordeanu

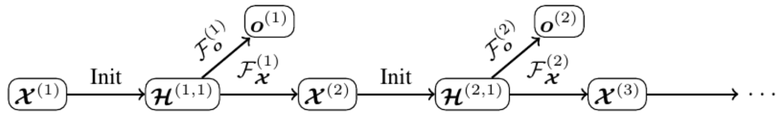
submitted to ICML 2019

Related Work



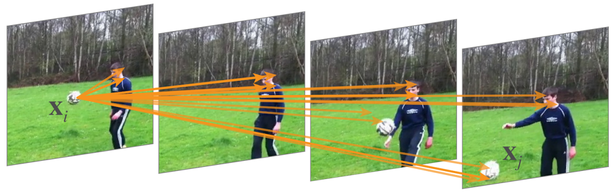
- Input has the **explicit graph structure**
- Nodes are updated using an attention mechanism for weighting neighbouring nodes information

GAT: [Velickovic et al., ICLR 2018]



- Applied on symbolic data (bAbI, program verification) with **sequential output**
- Each graph message-passing is followed by a **recurrent update** to obtain a sequential output

Gated: [Li et al., ICLR 2016]



- Activity Recognition task
- All feature maps points are nodes in the graph
- A **completely-connected** graph for the entire **video** (**fixed** number of frames)
- Processing all nodes: **space** or **time** in the **same way**

Non-Local: [Wang et al., CVPR 2018]

Overall approach

1. A computational model for learning in **spatio**-**temporal** data
   - **space** and **time** are treated **differently** (**complementary** recurrent units in both of them)
   - use **nodes** at **different scales**

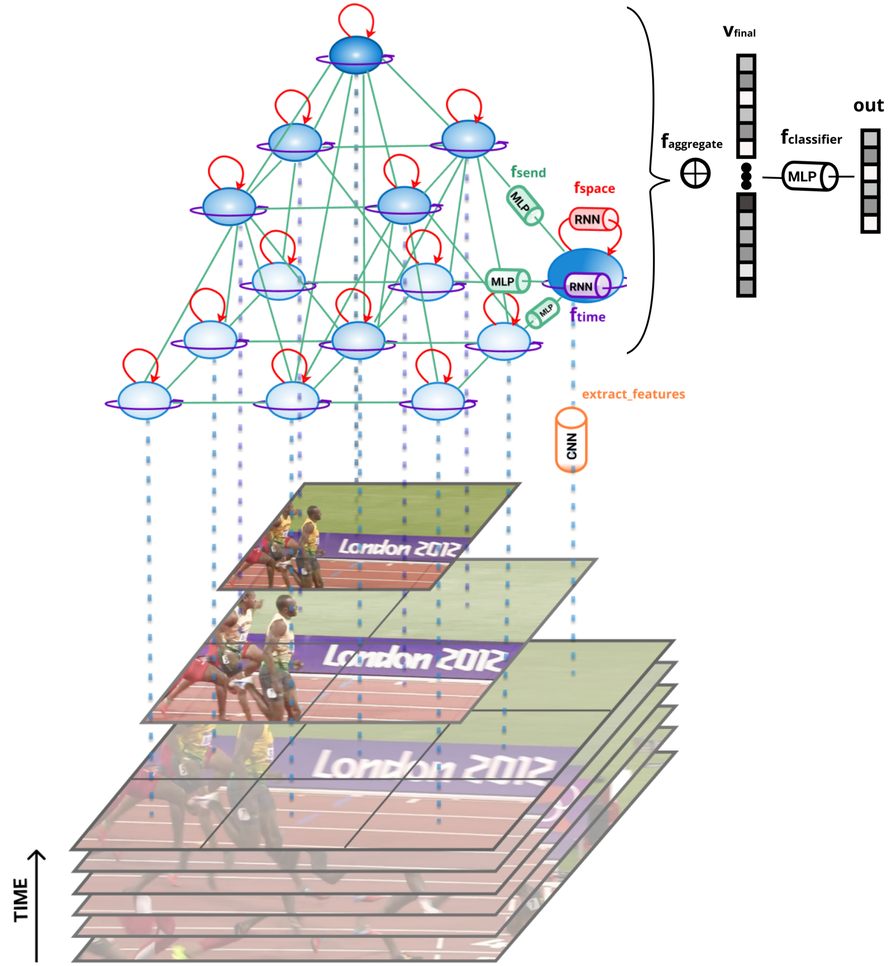
2. **Recurrent Space-time Graph (RSTG)** model is **general** and could be applied to **various visual learning** problems in the **spatio**-**temporal** domain
   - build a graph from an **unstructured** video
   - the recurrent processing allows **input** of **variable length**

3. **Tested** it on **two video classification tasks**
   - **complex interactions**
   - **superior** performance to several **powerful baselines**

Graph Creation

- Divide a frame in **regions** using a **spatial grid over multiple scales**
- Extract **spatio-temporal** features using **2D or 3D CNN** for **each region**
- **Each node** receives information from a **single region**

- Two nodes are **connected** if:
  - they are **neighbours** in the **same grid** or
  - their **regions** at **different scales intersect**



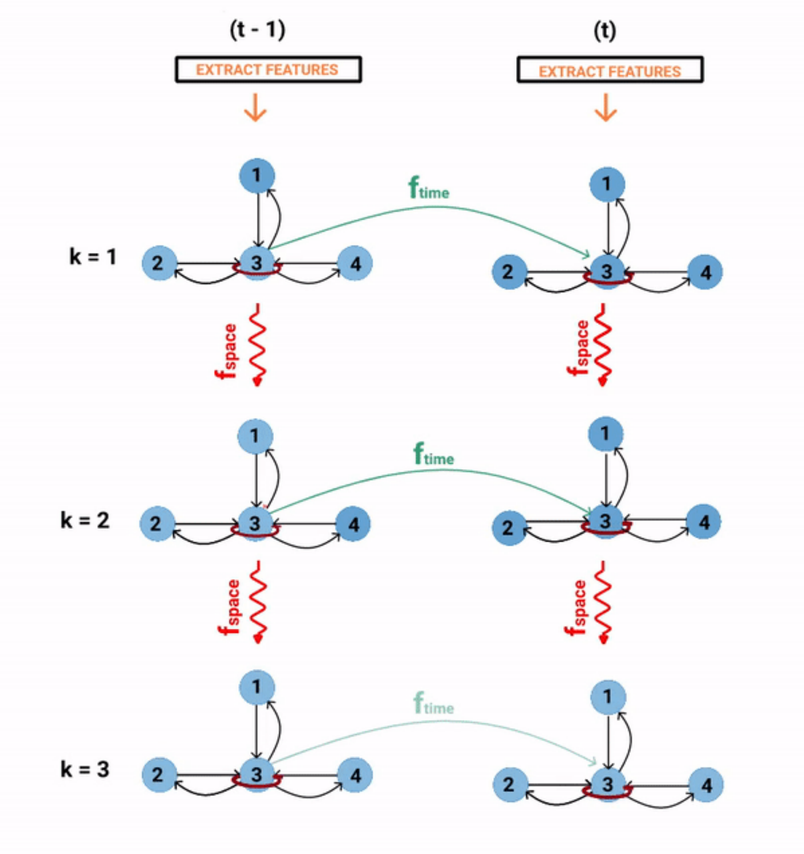
Graph Processing

- The **nodes** receive **local features** pooled from their **regions** and integrate information **globally** (at every time step) by:

  1. **Space Processing Stage**: **recurrent** message passing at the **spatial** level between the connected nodes

  2. **Time Processing Stage**: **recurrent** in time: each node at the **present time** receives a message from its **past state**

- Capture both **local** and **long range spatio-temporal interactions**



Bitdefender

Space Processing Stage

1. **Message sending function**: send messages between all connected nodes

$$f_{send}(\mathbf{v}_j, \mathbf{v}_i) = \mathbf{MLP}_s([\mathbf{v}_j \,|\, \mathbf{v}_i]) \in \mathbb{R}^D$$

2. **Gather function:** Gathers information (all received messages from all neighbours)

$$f_{gather}(\mathbf{v}_i) = \sum_{j \in \mathcal{N}(i)} \alpha(\mathbf{v}_j, \mathbf{v}_i) f_{send}(\mathbf{v}_j, \mathbf{v}_i) \in \mathbb{R}^D$$

3. **Update function:** Update the internal node representation (**local** identity + **global** info)

$$\mathbf{v}_i \leftarrow f_{space}(\mathbf{v}_i) = \mathbf{MLP}_u([\mathbf{v}_i \,|\, f_{gather}(\mathbf{v}_i)]) \in \mathbb{R}^D$$

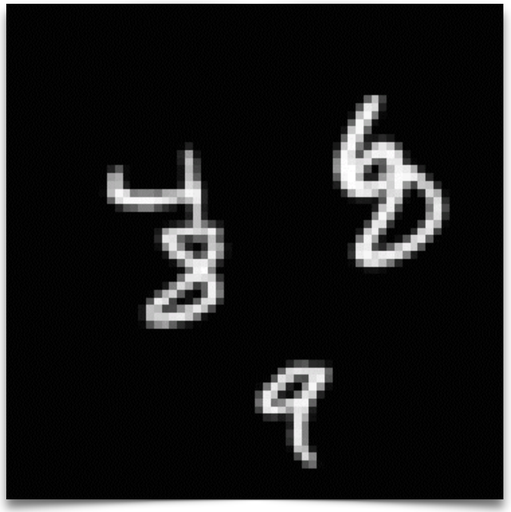
Time Processing Stage

1. **Aggregate** the **spatial** representation with its **time** representation from the previous step using a **recurrent** function

$$\mathbf{v}_{i,time}^{t} \leftarrow f_{time}(\mathbf{v}_{i,space}, \mathbf{v}_{i,time}^{t-1})$$

2. Use a **Time Processing Stage** after each **Space Processing Stage**
   - each stage **aggregates information** representing **interactions** between **increasingly distant nodes**

Results



Random

Sync pair (4, 2)

Model	3 Digits	5 Digits
Mean + LSTM	77.0	-
Conv + LSTM	95.0	39.7
I3D	-	88.2
RSTG: Space-Only	61.3	-
RSTG: Time-Only	89.7	-
RSTG: Homogenous	95.7	58.3
RSTG: 1-temp-stage	97.0	74.1
RSTG: all-temp-stages	**98.9**	**90.9**

Model	Kinetics 400 validation
I3D	72.26
RSTG	72.50

- Take 5 MNIST digits that move
- Find the pair that **moves synchronously**

- **Next**: applying it for **Video Captioning**, building over their previous paper **[I. Duță et al. BMVC 2018]**

Thank you!

Bitdefender®

Elena Burceanu
eburceanu@bitdefender.com

Emanuela Haller
ehaller@bitdefender.com

Our full team and projects:

bit-ml.github.io

Iulia Duță
iduta@bitdefender.com

Andrei Nicolicioiu
anicolicioiu@bitdefender.com

Marius Leordeanu
marius.leordeanu@cs.pub.ro