Motion representations for visual analysis

Silvia Pintea

Joint work with a wonderful team:

Dr. Xin Liu

(TUDelft)

Dr. Yancong Lin

(TUDelft)

Prof. Arnold Smeulders

(UvA)

Prof. Jan van Gemert

(TUDelft)

Prof. Jouke Dijkstra

(LUMC)

Silvia-Laura Pintea

FOLLOW

GET MY OWN PROFILE

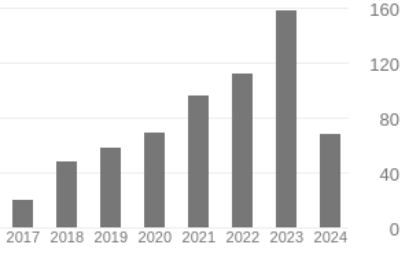
Computer vision researcher
Verified email at lumc.nl - Homepage

Computer Vision Video understanding Motion analysis Future anticipation

TITLE	CITED BY	YEAR
Video acceleration magnification Y Zhang, SL Pintea, JC Van Gemert Proceedings of the IEEE Conference on Computer Vision and Pattern …	117	2017
Deja Vu: Motion Prediction in Static Images SL Pintea, JC van Gemert, AWM Smeulders Proceedings of the European Conference on Computer Vision (ECCV) 2014, 172-187	89	2014
Deep hough-transform line priors Y Lin, SL Pintea, JC van Gemert Proceedings of the European Conference on Computer Vision (ECCV) 2020: 16th …	71	2020
Divide and count: Generic object counting by image divisions T Stahl, SL Pintea, JC Van Gemert	67	2018

Cited by VIEW ALL

	All	Since 2019
Citations	679	561
h-index	13	11
i10-index	16	14

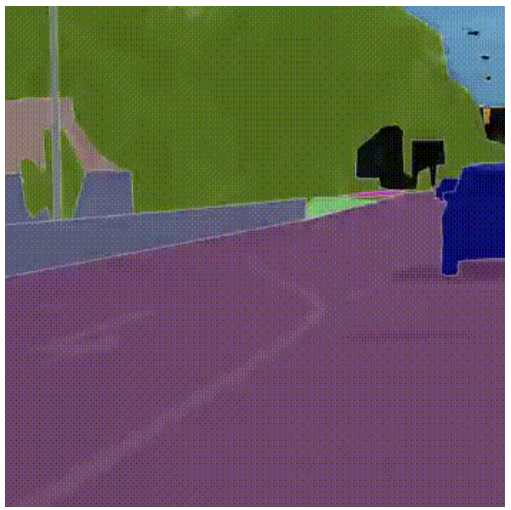


- PhD in Computer Vision at UvA with Arnold Smeulders in 2017
- R&D Engineer at Blippar/Layar on augmented reality
- Researcher at TUDelft from 2017 - 2022
- Senior postdoc researcher at LUMC on surgery video analysis

- **Supervision**: 15 MSc students, 2 PhDs (co-promotor)
- **Teaching**:
  - Course manager: BSc Image Processing @ TUDelft
  - Co-lecturer: MSc Computer Vision by Deep Learning @ TUDelft

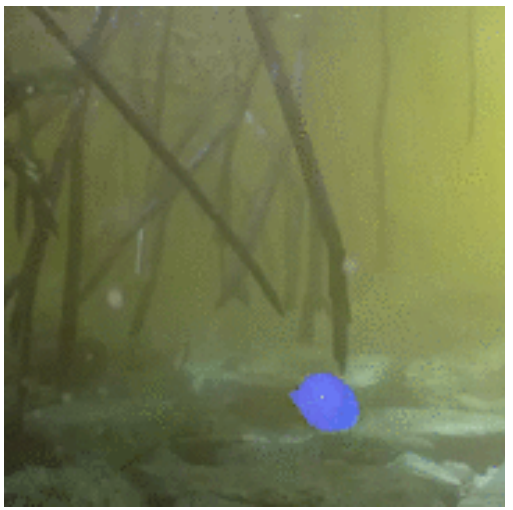
Why motion?

- Everywhere around us and analyzing it enables:



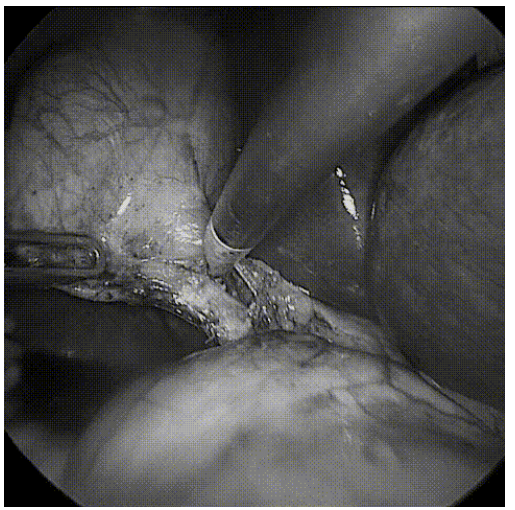
Safer transportation

(Source: Cityscapes dataset)



Ecosystem monitoring

(Source: DeepFish dataset)



Better medical diagnosing

(Source: Cholech80 dataset)

Prior: Anticipating short motions

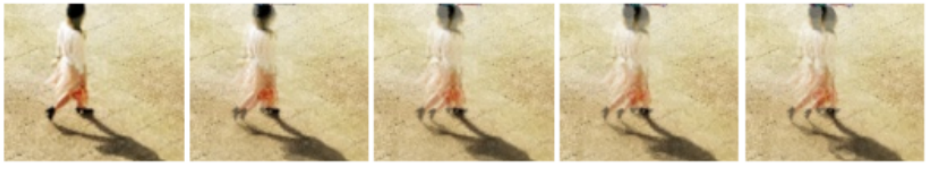
- European Computer Vision Conference (ECCV), 2014

Déjà Vu:
Motion Prediction in Static Images

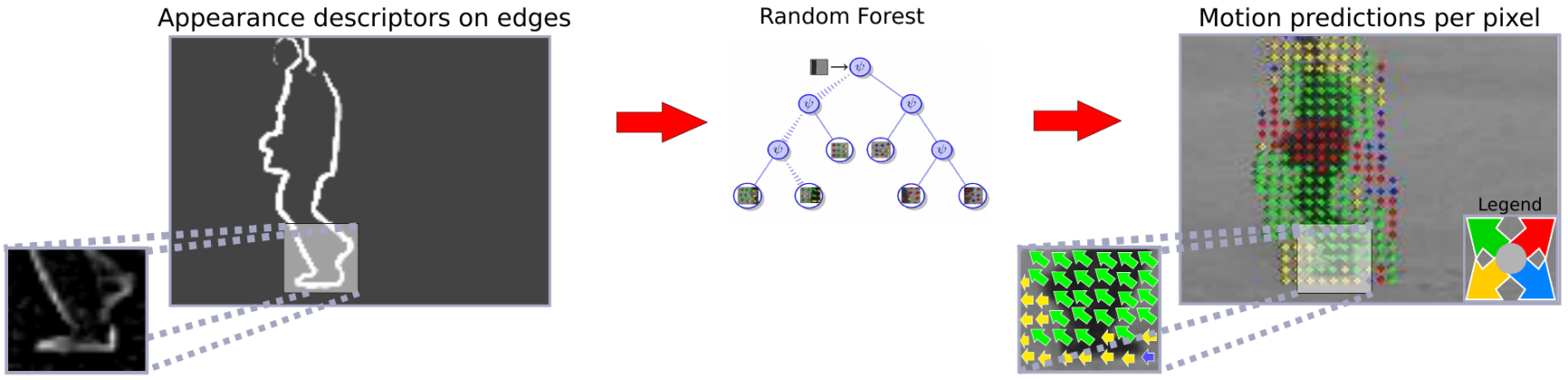
Silvia L. Pintea, Jan C. van Gemert, and Arnold W. M. Smeulders

Intelligent Systems Lab Amsterdam (ISLA), University of Amsterdam
Science Park 904, 1098 HX, Amsterdam, The Netherlands

Abstract. This paper proposes motion prediction in single still images by learning it from a set of videos. The building assumption is that similar motion is characterized by similar appearance. The proposed method learns local motion patterns given a specific appearance and adds the predicted motion in a number of applications. This work (i) introduces a novel method to predict motion from appearance in a single static image, (ii) to that end, extends of the Structured Random Forest with regression derived from first principles, and (iii) shows the value of adding motion predictions in different tasks such as: weak frame-proposals containing unexpected events, action recognition, motion saliency. Illustrative results indicate that motion prediction is not only feasible, but also provides valuable information for a number of applications.

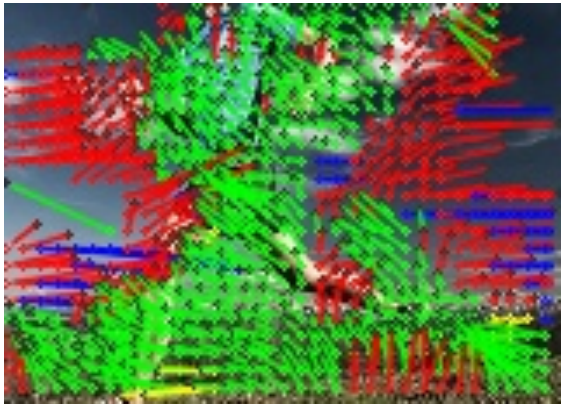


Prior: Anticipating short motions



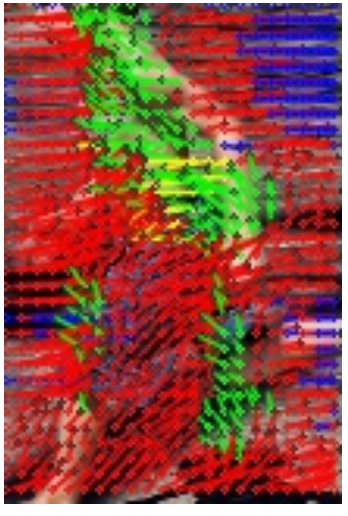
- Hypothesis:
  - Structured random forests to learn to associate input image-patches to motion-patches.
  - Mosaic a motion image using predicted motion-patches from image-patches.

Prior: Anticipating short motions

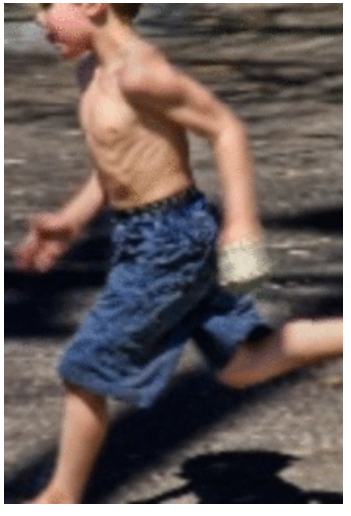


Predicted flow

Animated static image



Predicted flow



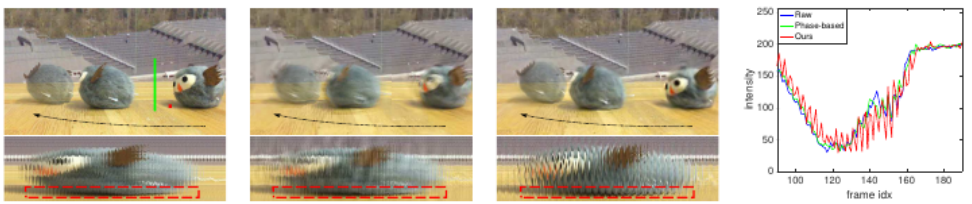
Animated static image

Prior: Magnifying subtle motions

- Computer Vision and Pattern Recognition (CVPR), 2017

Video Acceleration Magnification

Yichao Zhang, Silvia L. Pintea, and Jan C. van Gemert
Vision Lab, Delft University of Technology
Delft, Netherlands



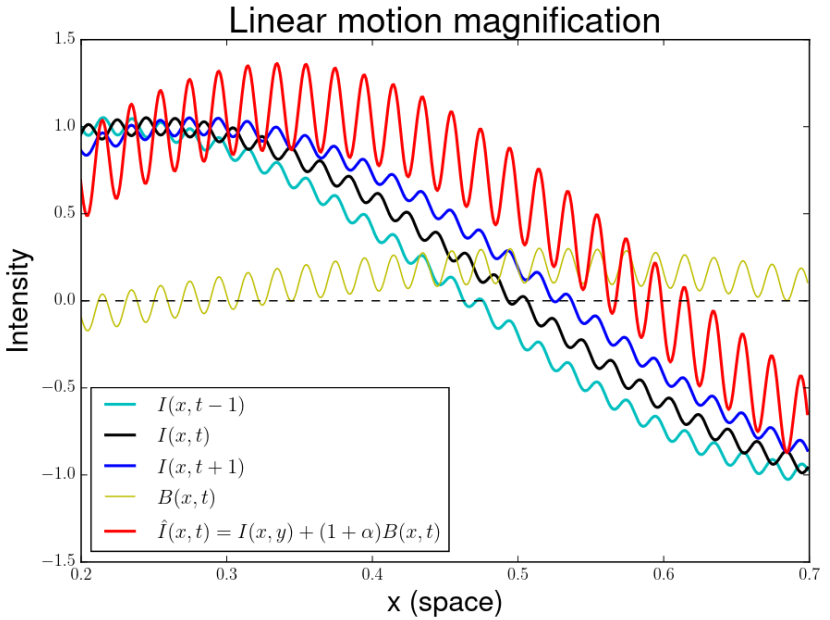
(a) Original video. (b) Phase-based [26]. (c) Ours. (d) Intensity changes.

Figure 1: A toy moving along a trajectory depicted by the black arrow, while vibrating at a high frequency. The top row shows 3 frames overlayed to indicate the toy's trajectory. The bottom row shows a single column of pixels – the green line in (a) – for relevant video frames. (a) Original video. (b) Phase-based motion magnification [26]. (c) Our proposed acceleration magnification. (d) Intensity changes at the location of the red pixel in the top row in (a) — corresponding to a spatio-temporal rectangle in the bottom row. Our method generates sharper results with a greater magnification than the phase-based method in [26]. See the supplementary material for the video result.

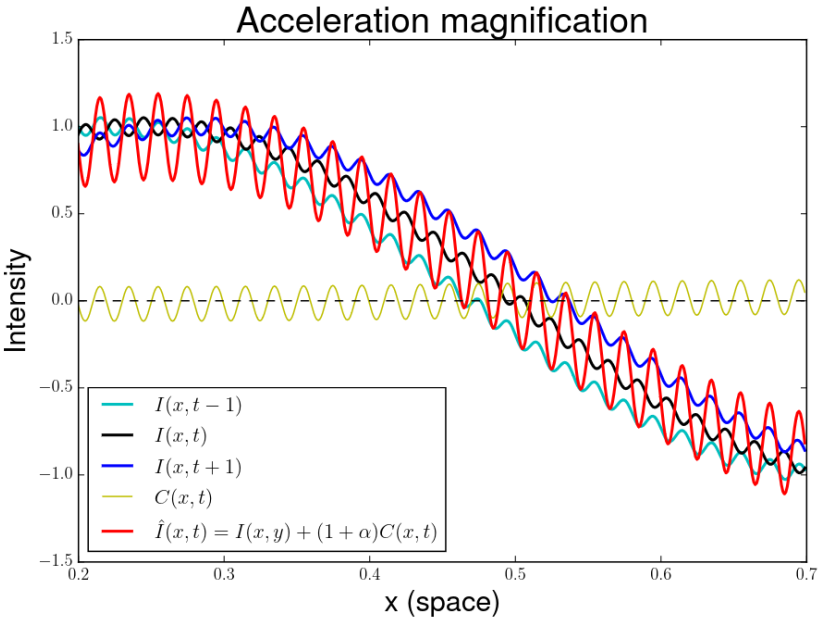
Prior: Magnifying subtle motions

- Hypothesis:
  - If an object has a constant (piece-wise linear motion) the subtle motions are captured in the 2*nd* derivative of the motion (*i.e.* acceleration)
  - Magnify the acceleration of motion

Prior: Magnifying subtle motions



(a)



(b)

- Linear methods: magnify $B(x,t) \propto \frac{\partial I(x,t)}{\partial x}$
- Acceleration magnification magnifies $C(x,t) \propto \frac{\partial^2 I(x,t)}{\partial x^2}$

Prior: Magnifying subtle motions

Prior: Eulerian motion representations

- European Conference on Computer Vision Workshops (ECCVw), 2018

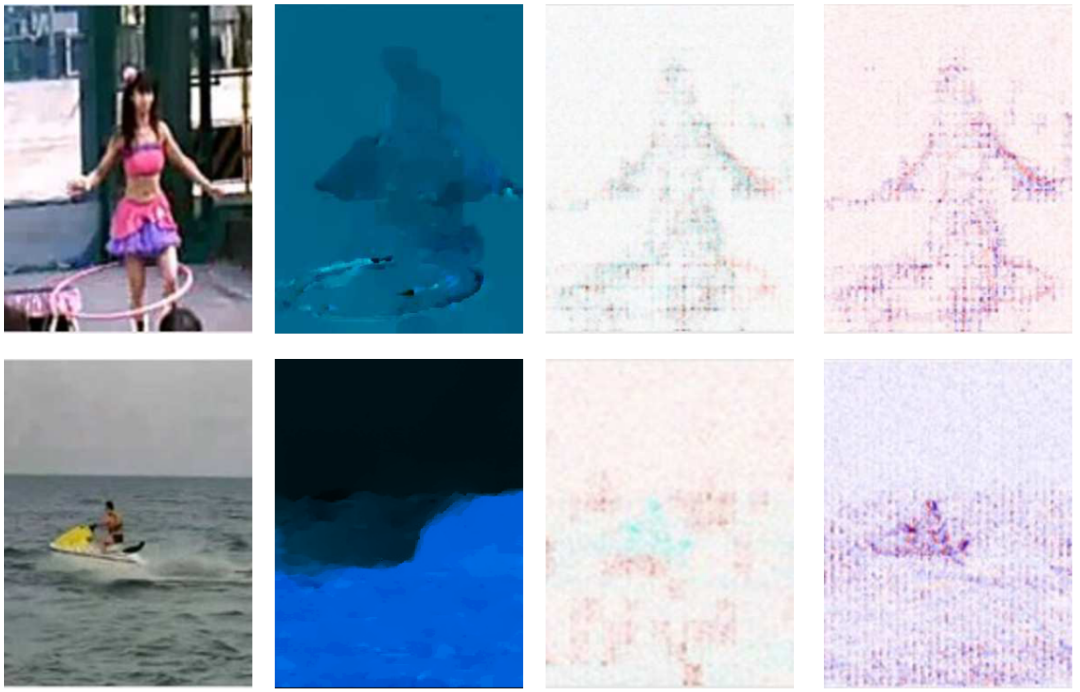
Using phase instead of optical flow for action recognition

Omar Hommos[1], Silvia L. Pintea[1],
Pascal S.M. Mettes[2], Jan C. van Gemert[1]

[1]Computer Vision Lab, Delft University of Technology, Netherlands
[2]Intelligent Sensory Interactive Systems, University of Amsterdam, Netherlands

Abstract. Currently, the most common motion representation for action recognition is optical flow. Optical flow is based on particle tracking which adheres to a Lagrangian perspective on dynamics. In contrast to the Lagrangian perspective, the Eulerian model of dynamics does not track, but describes local changes. For video, an Eulerian phase-based motion representation, using complex steerable filters, has been successfully employed recently for motion magnification and video frame interpolation. Inspired by these previous works, here, we proposes learning Eulerian motion representations in a deep architecture for action recognition. We learn filters in the complex domain in an end-to-end manner. We design these complex filters to resemble complex Gabor filters, typically employed for phase-information extraction. We propose a phase-information extraction module, based on these complex filters, that can be used in any network architecture for extracting Eulerian representations. We experimentally analyze the added value of Eulerian motion representations, as extracted by our proposed phase extraction module, and compare with existing motion representations based on optical flow, on the UCF101 dataset.

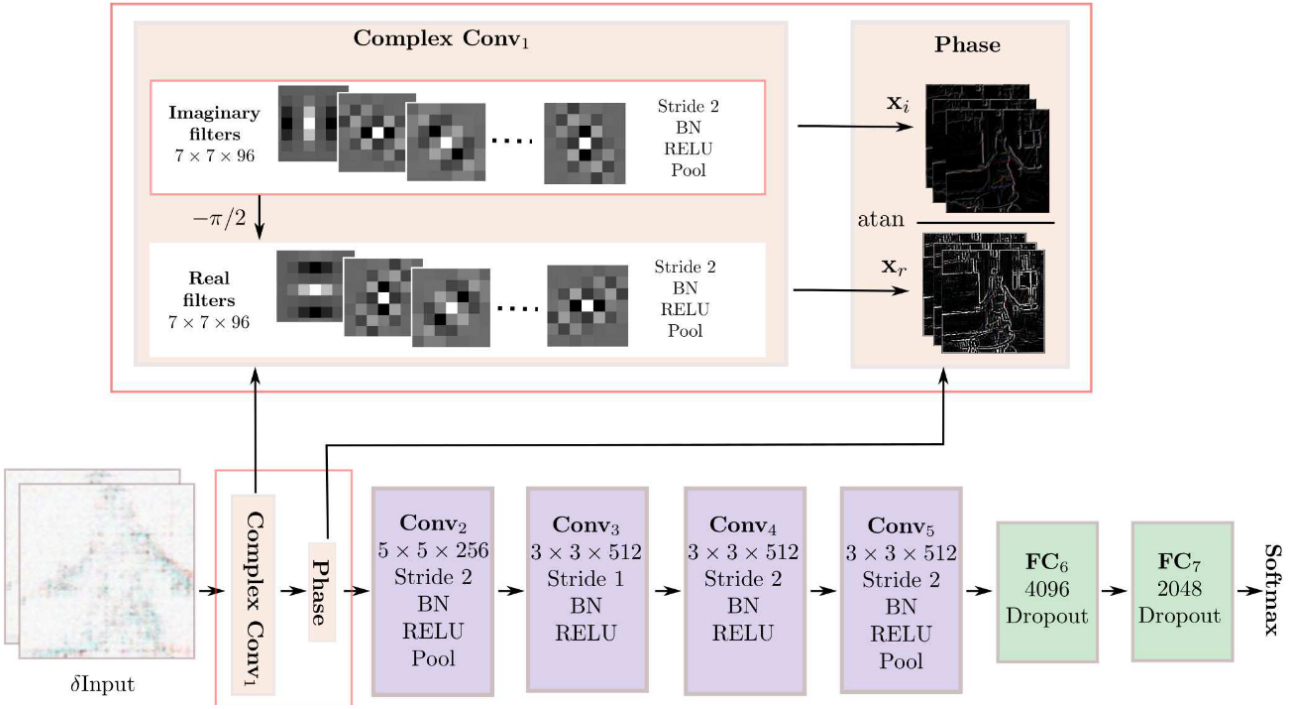
Prior: Eulerian motion representations



(a) Original input. (b) Optical flow. (c) RGB derivative. (d) Phase derivative.

- Eulerian motion representation are more informative at object boundaries

Prior: Eulerian motion representations



- We start from temporal image derivatives
- We use a set of complex filters resembling Gabor-quadrature filters for extracting phase

Prior: Eulerian motion representations

Input	VGG-M [20]	PhaseStream (our)
RGB	**52.3** %	51.3 %
OF	67.7 %	N/A
dRGB	45.5 %	**48.8** %
dGray	74.3 %	**74.4** %
dPhase	65.4 %	**70.1** %

(i) Different inputs.

- Results on the dataset UCF101, where PhaseStream is VGG-M where the first layer is replaced with the complex filters.

Prior: Object localization by motion anticipation

- International Computer Vision and Conference (ICCV), 2023

Objects do not disappear:
Video object detection by single-frame object location anticipation

Xin Liu[1] Fatemeh Karimi Nejadasl[2] Jan C. van Gemert[1] Olaf Booij[1] Silvia L. Pintea[1]
Computer Vision Lab, Delft University of Technology[1]
Institute for Biodiversity and Ecosystem Dynamics, University of Amsterdam[2]

Abstract

Objects in videos are typically characterized by continuous smooth motion. We exploit continuous smooth motion in three ways. 1) Improved accuracy by using object motion as an additional source of supervision, which we obtain by anticipating object locations from a static keyframe. 2) Improved efficiency by only doing the expensive feature computations on a small subset of all frames. Because neighboring video frames are often redundant, we only compute features for a single static keyframe and predict object locations in subsequent frames. 3) Reduced annotation cost, where we only annotate the keyframe and use smooth pseudo-motion between keyframes. We demonstrate computational efficiency, annotation efficiency, and improved mean average precision compared to the state-of-the-art on four datasets: ImageNet VID, EPIC KITCHENS-55, YouTube-BoundingBoxes and Waymo Open dataset. Our source code is available at https://github.com/L-KID/Video-object-detection-by-location-anticipation.

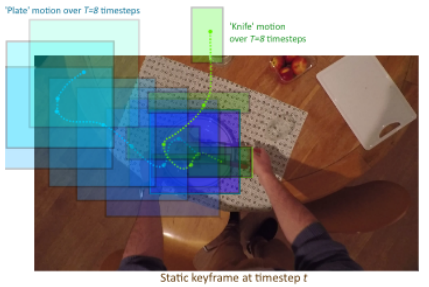
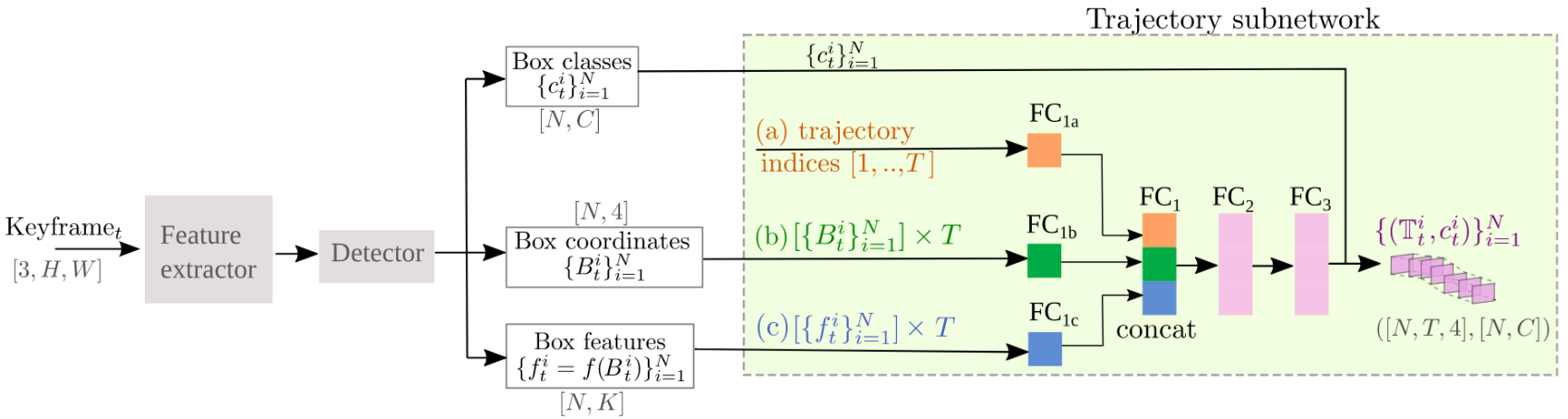


Figure 1. Anticipating future object locations from a static keyframe is *efficient*. We only do the expensive feature extraction on a small subset of keyframes, while still accessing bounding-box locations for all video frames. Moreover, exploiting motion cues as additional supervision *improves* object detection. By sampling a static keyframe at time t and anticipating the object locations over the next T timesteps, we incorporate temporal consistency and smoothness of object motion.

Prior: Object detection by motion anticipation

- Hypothesis:
  - Detect objects on a subset of frames, and anticipate object locations in-between.
  - **After-effect:** Improved object detection accuracy: Gestalt principle of common fate (things that move together belong together)

Prior: Object localization by motion anticipation



- Use an image object detector to obtain: object bounding-boxes, object classes, and object features.
- We combine these in a trajectory subnetwork to predict object locations over the next T frames.

Prior: Object localization by motion anticipation

Methods	Backbone	No Post-proc.	mAP (%)	Train-time (hrs/epoch)	Runtime (FPS)
Faster-RCNN [57]	R101	✓	73.6	1.55	21.2
LWND [36]	R101	✓	76.3	-	20.0
FGFA [80]	R101		78.4	6.59	5.0
THP [78]	R101+DCN	✓	78.6	-	-
ST-Lattice [6]	R101		79.6	1.40	20.0
D&T [18]	R101		80.2	6.56	5.0
MANet [70]	R101		80.3	6.88	4.9
STSN [5]	R101+DCN		80.4	-	-
STMN [73]	R101		80.5	2.49	13.2
TROI [22]	R101		80.5	5.18	6.4
SELSA [72]	R101		80.5	3.15	10.6
OGEMN [13]	R101+DCN		81.6	-	8.9
SparseVOD [29]	R101	✓	81.9	-	14.4
BoxMask [28]	R101	✓	83.2	-	6.1
RDN [14]	R101		83.8	-	-
HVRNet [24]	R101		83.8	-	-
TF-Blender [9]	R101	✓	83.8	-	4.9
MEGA [7]	R101		84.5	6.34	5.3
TransVOD [77] (Def. DETR)	R101	✓	81.9	-	32.3
PTSEFormer [69] (Def. DETR)	R101	✓	88.1	-	-
TransVOD [77] (Def. DETR)	SwinB	✓	90.1	-	14.9
Ours (Faster RCNN)	R101	✓	87.2	0.78	**39.6**
Ours (Def. DETR)	R101	✓	87.9	-	36.4
Ours (Def. DETR)	SwinB	✓	**91.3**	-	18.1

Table 4. **[C1]: Experiments on ImageNet VID.**

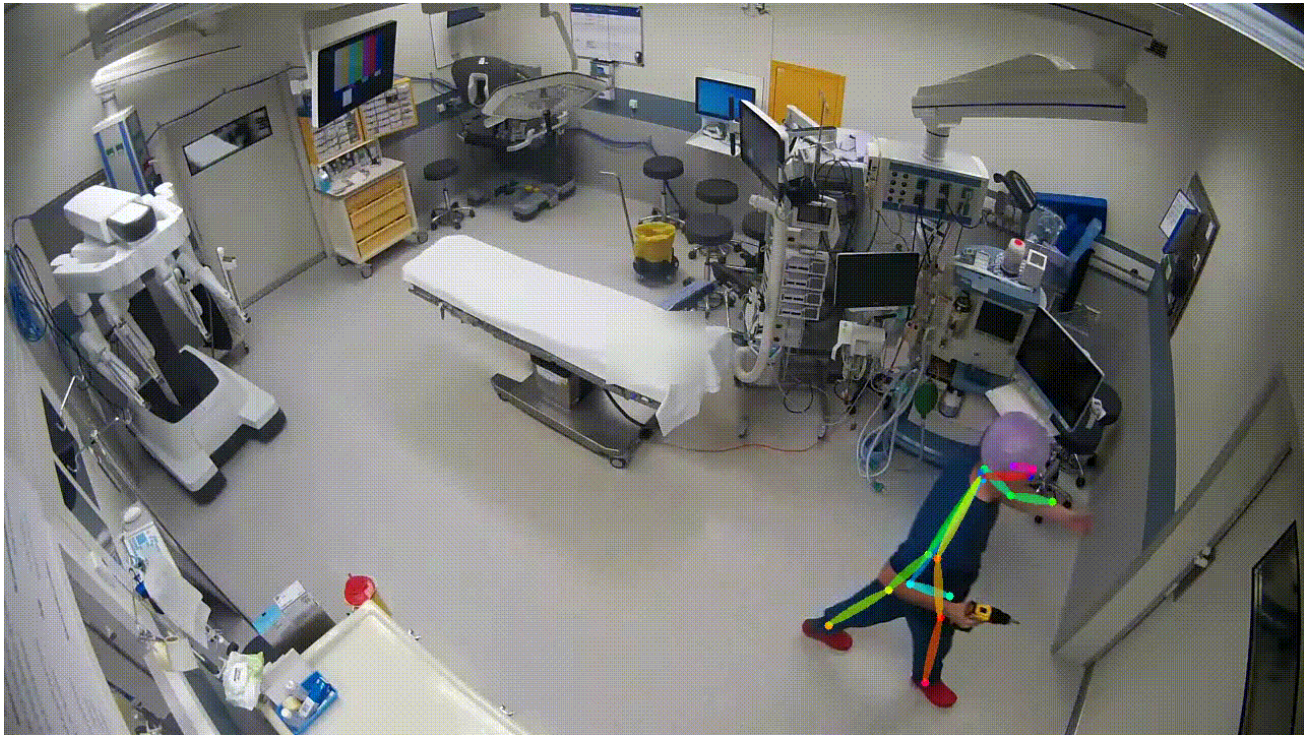
Methods	Backbone	mAP (%)	mAP (%) (slow)	mAP (%) (medium)	mAP (%) (fast)
FGFA [80]	R101	78.4	83.5	75.8	57.6
MANet [70]	R101	80.3	86.9	76.8	56.7
SELSA [72]	R101	80.5	86.9	78.9	61.4
OGEMN [13]	R101+DCN	81.6	86.2	78.7	61.1
HVRNet [24]	R101	83.8	88.7	82.3	**66.6**
IFFNet [38]	R101	79.7	87.5	78.7	60.6
Ours (Faster RCNN)	R101	**87.2**	**92.2**	**86.1**	66.5

Table 5. **[C1]: ImageNet VID across different motion speeds**. Our method improves mAP on different motion speeds.

	S1		S2	
Methods	mAP@.5	mAP@.75	mAP@.5	mAP@.75
EPIC [11]	34.2	8.5	32.0	7.9
Faster-RCNN [72]	36.6	9.9	31.9	7.4
SELSA [72]	37.9	9.8	34.8	8.1
SELSA-ReIm + TROI [22]	42.2	-	39.6	-
BoxMask [28]	44.3	18.5	41.3	15.7
Ours (Faster RCNN)	**44.9**	**18.7**	**41.7**	**16.0**

Table 6. **[C2]: Experiments on EPIC KITCHENS-55.** S1 and S2 represent Seen and Unseen splits, respectively. Our method achieves promising results for both test sets and IoU thresholds.

Currently: Video appearance variations



- Video appearance variations are a challenge:
  - appearance differs per surgery-room,
  - per surgery type,
  - per hospital, *etc.*

Currently: Video appearance variations

- International Computer Vision and Conference Workshop (ICCVw), 2023

Is there progress in activity progress prediction?

Frans de Boer[1] Jan C. van Gemert[1] Jouke Dijkstra[2] Silvia L. Pintea[1,2]

[1] Computer Vision Lab, Delft University of Technology
[2] Division of Image Processing (LKEB), Leiden University Medical Center

Abstract

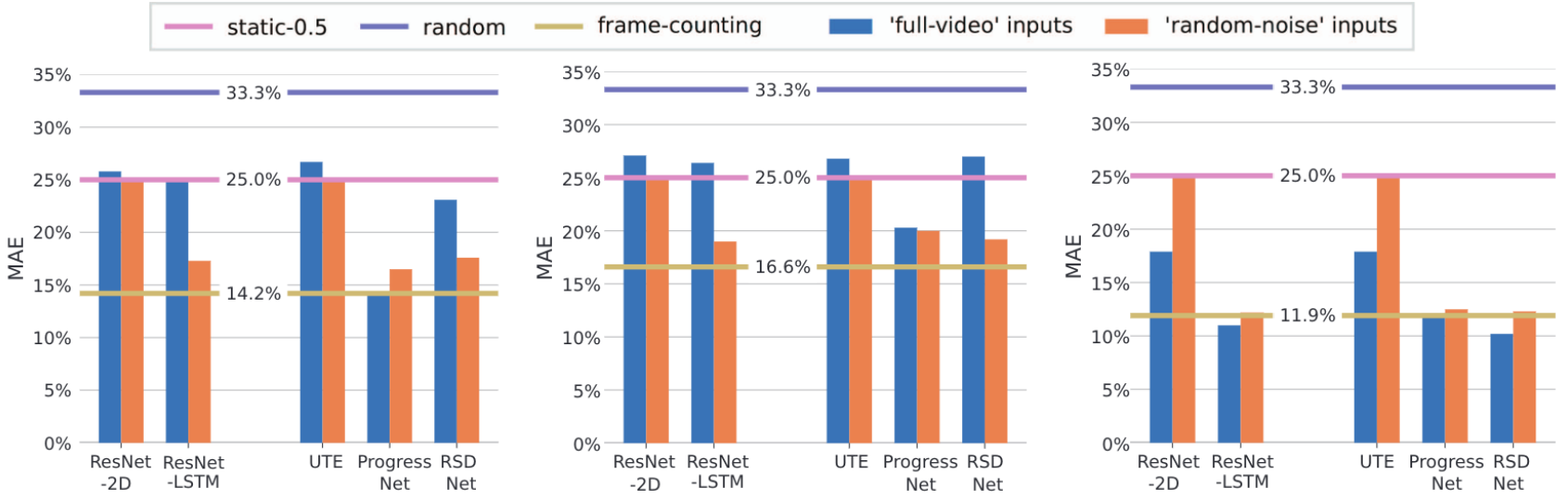
Activity progress prediction aims to estimate what percentage of an activity has been completed. Currently this is done with machine learning approaches, trained and evaluated on complicated and realistic video datasets. The videos in these datasets vary drastically in length and appearance. And some of the activities have unanticipated developments, making activity progression difficult to estimate. In this work, we examine the results obtained by existing progress prediction methods on these datasets. We find that current progress prediction methods seem not to extract useful visual information for the progress prediction task. Therefore, these methods fail to exceed simple frame-counting baselines. We design a precisely controlled dataset for activity progress prediction and on this synthetic dataset we show that the considered methods can make use of the visual information, when this directly relates to the progress prediction. We conclude that the progress prediction task is ill-posed on the currently used real-world datasets. Moreover, to fairly measure activity progression we advise to consider a, simple but effective, frame-counting baseline.

is generally the case in real-world scenarios. The main challenge for progress prediction is extracting meaning from the visual inputs, which, ideally relates to the specific phases of the activity and, thus, enables predicting progress.

To address this challenge, current methods rely on deep networks, such as *VGG-16* [23], *ResNet* [9], *YOLOv2* [22], or *I3D* [4] to extract visual information. Furthermore, to remember information over time, current progress prediction methods [3, 26] rely on memory blocks and recurrent connections [12]. While these embeddings and recurrent connections are useful for extracting visual information and keeping track of the activity progression over time, they may also overfit to uninformative artifacts. Here, we aim to analyze if such undesirable learning strategies are occurring when performing progress prediction.

To this end, we consider the state-of-the-art progress prediction methods [3, 17, 26], as well as two more simple learning-based methods: a $2D$-only *ResNet*, and a *ResNet* model augmented with recurrent connections. We evaluate all these learning methods across three video datasets used for progress prediction: *UCF101-24* [24], *Breakfast* [15, 16], and *Cholec80* [25]. Additionally, we compare the learning-based methods with simple non-learning baseline methods such as simply frame counting.

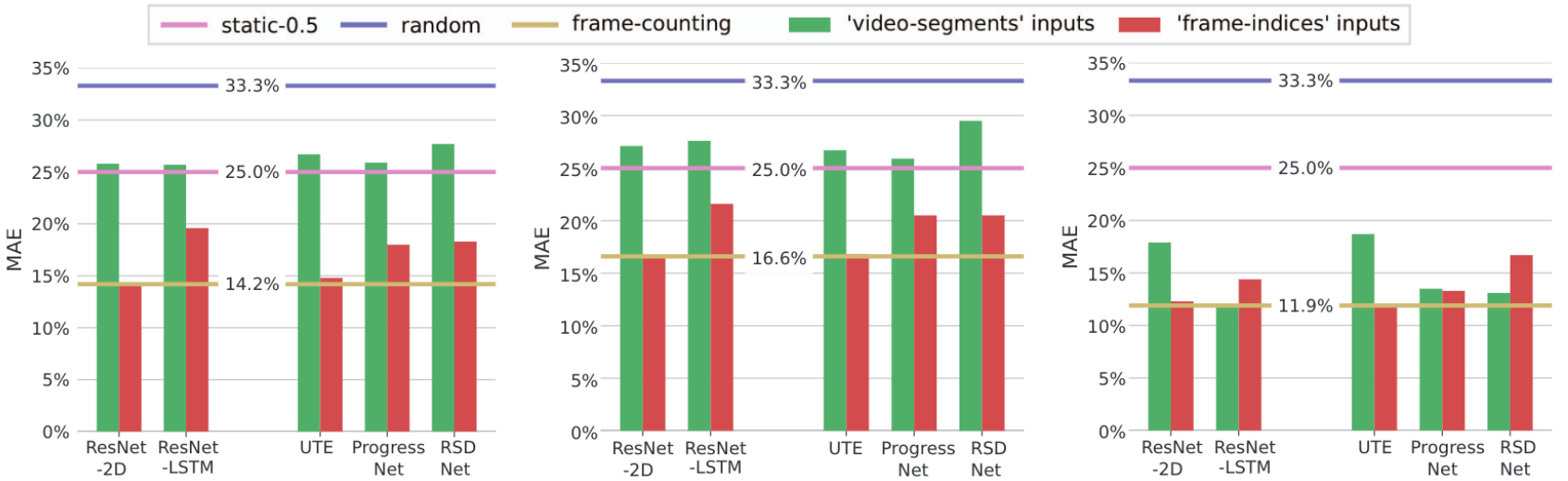
Currently: Video appearance variations



(a) *UCF101-24* on *full-videos*. (b) *Breakfast* on *full-videos*. (c) *Cholec80* on *full-videos*.

- When the full video is input to the deep models, they perform on par with simple counting baselines.

Currently: Video appearance variations



(a) *UCF101-24* on *video-segments*. (b) *Breakfast* on *video-segments*. (c) *Cholec80* on *video-segments*.

- When the video segments with random starting points are input to the deep models, they are outperformed by simple counting baselines.
- Inputting the video visual information seems detrimental rather than useful

Currently: Video appearance variations

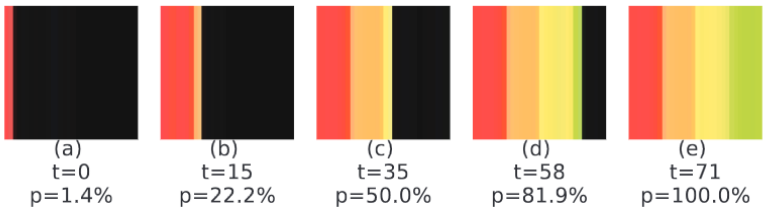
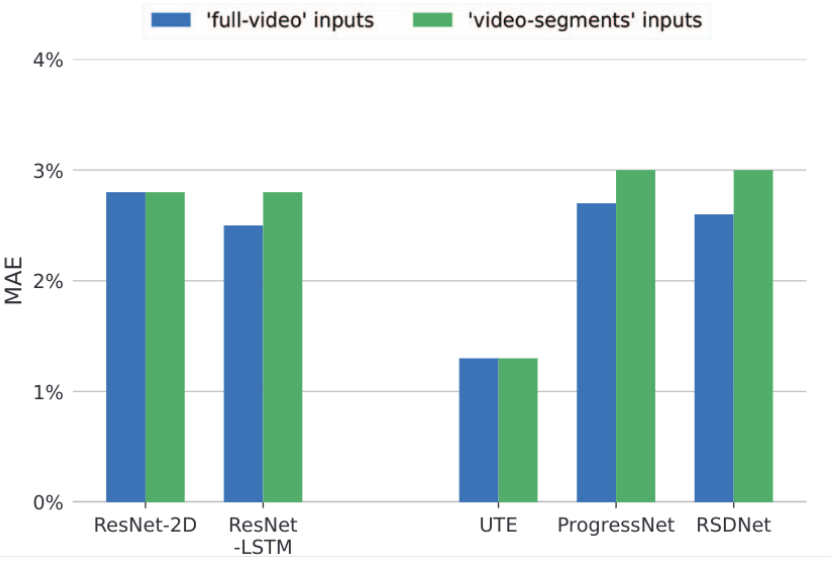


Figure 4. Visualisation of a progress bar from our synthetic *Progress-bar* dataset at timesteps t=0, t=15, t=35, t=58, and t=71. Each coloured section indicates visually a 25% section, but due to variance in the speed, the actual video progress may differ at these points.



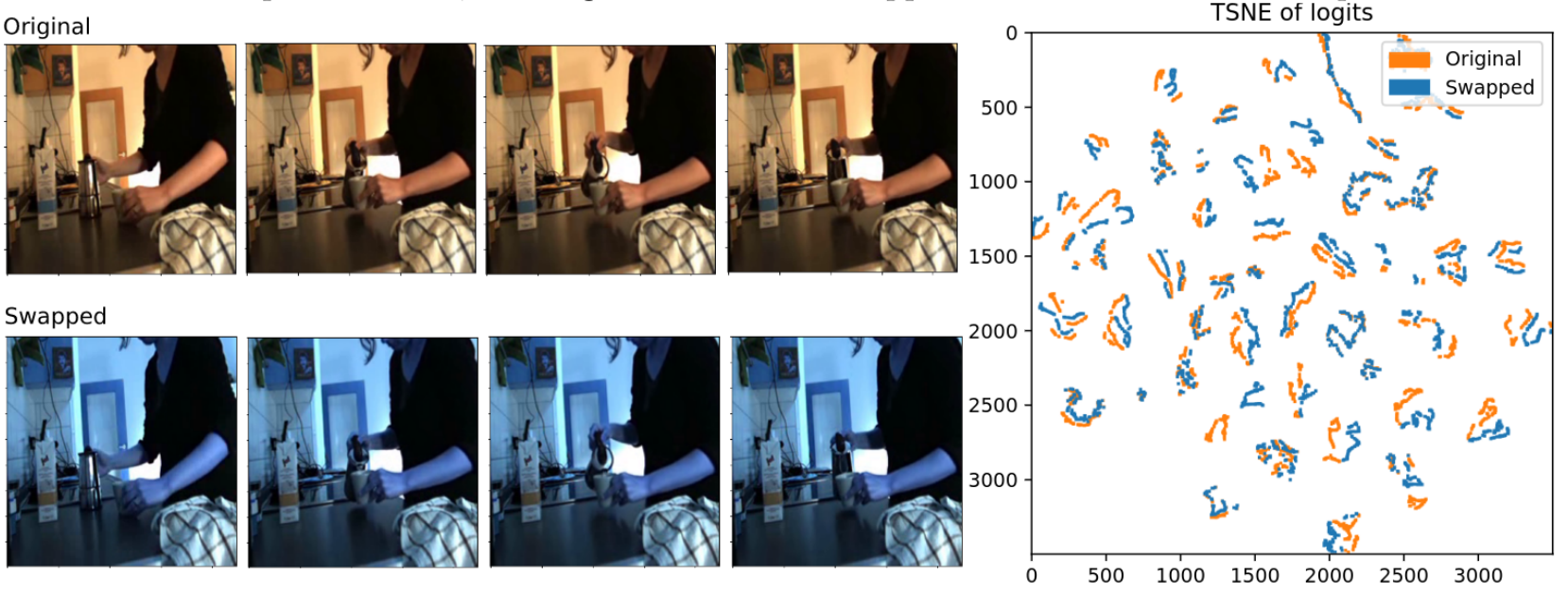
- If there is a direct correlation between the video appearance and the progress, all methods can outperform the simple counting baselines.

Conclusions:

- Predicting video progression is challenging because of the appearance variations
- Current learning based methods are outperformed by non-learning baselines.

Next: Appearance-free motion

Fig: TSNE plot of I3D features pretrained on Kinetics (each squiggle is a video).



- **Problem:** Video (motion) representations are tied to video appearance.
- **Consequence:** Models need to be refit to every new setting, and the data must contain an exhaustive set of appearance variations.

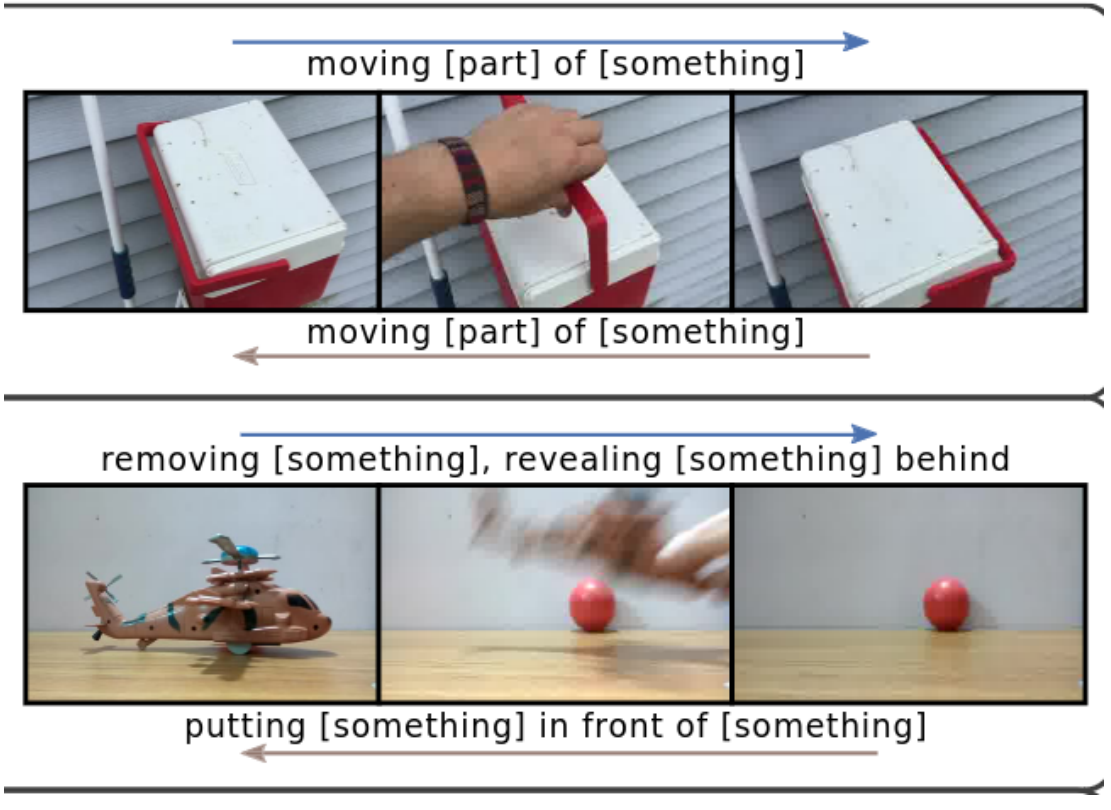
Next: Appearance-free motion

What's the problem with the appearance?

Next: Appearance-free motion

What's the problem with the appearance?

- Appearance is not sufficient to differentiate visually similar actions:



[*Retro-Actions: Learning 'Close' by Time-Reversing 'Open' Videos*. W. Prince and D Damen, ICCVw, 2019]

Next: Appearance-free motion

- **(1) How to define implicit motion representations independent of appearance?**
  - Without defining the motion geometry explicitly, how can we extract motion features that are decorrelated from appearance?

Next: Appearance-free motion

- **(1) How to define implicit motion representations independent of appearance?**
  - Without defining the motion geometry explicitly, how can we extract motion features that are decorrelated from appearance?
- **(2) How to define motion geometries?**
  - Defining complex motion patterns from a set of motion primitives: *e.g.* grad, div, curl.

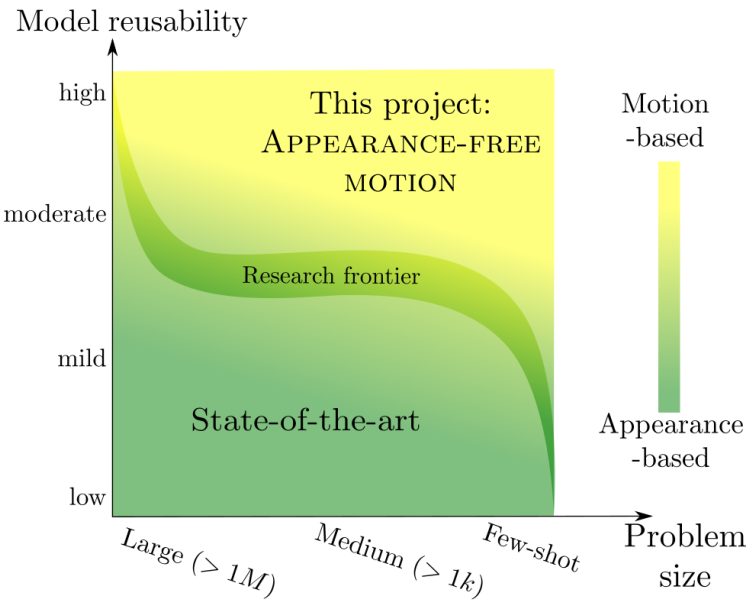
Next: Appearance-free motion

- **(1) How to define implicit motion representations independent of appearance?**
  - Without defining the motion geometry explicitly, how can we extract motion features that are decorrelated from appearance?
- **(2) How to define motion geometries?**
  - Defining complex motion patterns from a set of motion primitives: *e.g.* grad, div, curl.
- **(3) How to synthesize new data from appearance-free motion?**
  - How to create a new privacy-respecting dataset that retains the motion patterns but has a new appearance?

Next: Appearance-free motion

- **(1) How to define implicit motion representations independent of appearance?**
  - Without defining the motion geometry explicitly, how can we extract motion features that are decorrelated from appearance?
- **(2) How to define motion geometries?**
  - Defining complex motion patterns from a set of motion primitives: *e.g.* grad, div, curl.
- **(3) How to synthesize new data from appearance-free motion?**
  - How to create a new privacy-respecting dataset that retains the motion patterns but has a new appearance?
- **(4) How to benchmark appearance-free representations?**

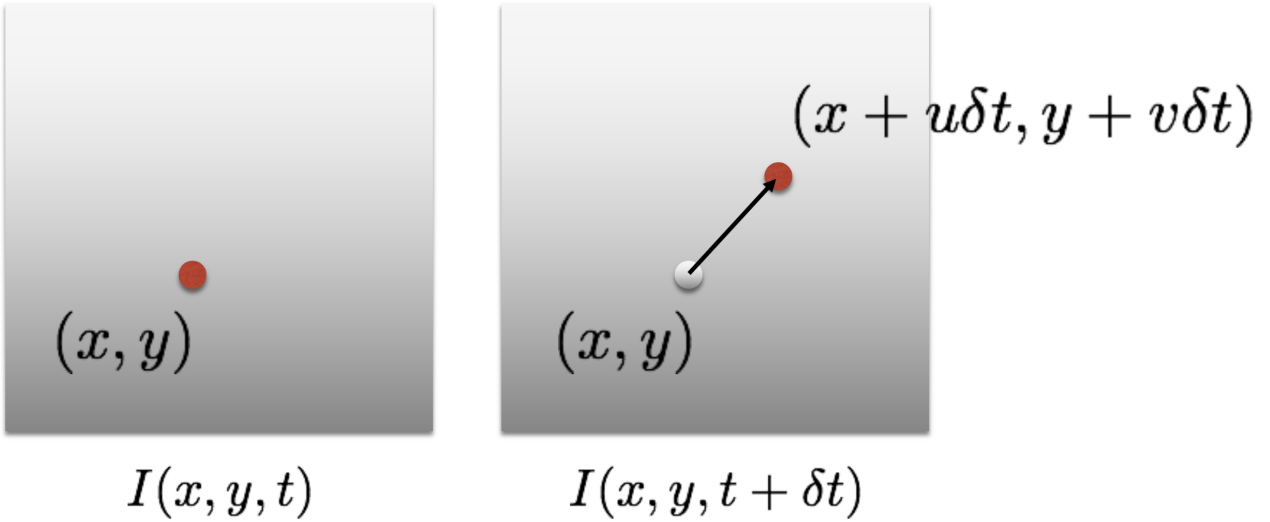
Next: Appearance-free motion



Expected gains:

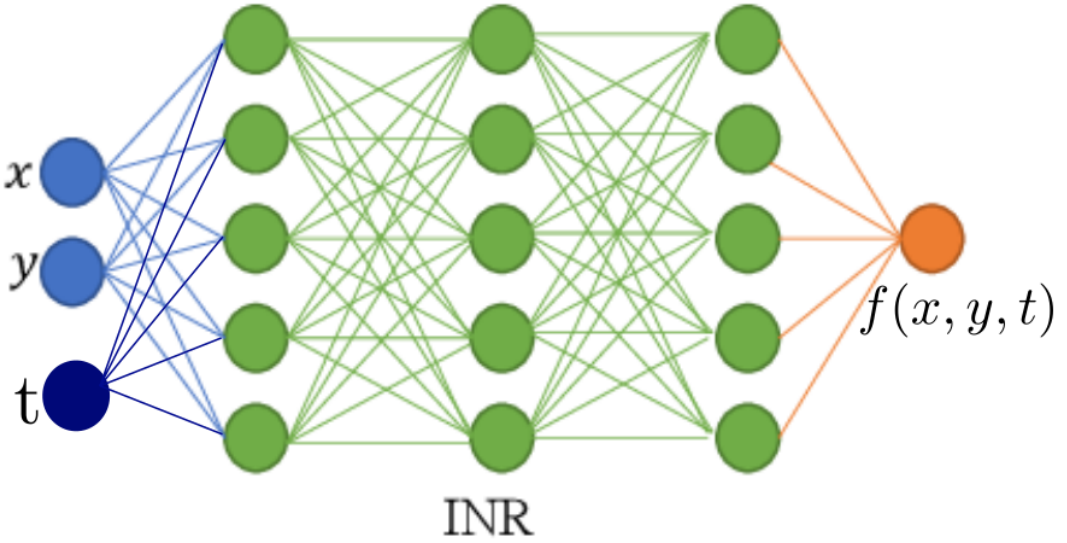
- Model re-usability without refitting across hospitals/cities/kitchens, etc.
- Learning from limited data (Since not all appearance settings need to be seen)
- Data sharing while preserving privacy, by synthesizing motion-consistent data.

Implicit motion features



- Analyze videos based on motion features (u, v) at every frame pixel (x, y, t)
- Learned to predict the motion in an implicit neural representation (INR)

Implicit motion features



- Represent the motion as an implicit function that maps input video positions $f(x, y, t) = (u, v)$

Implicit motion features

- Analyze videos based on motion features (u, v) at every frame pixel (x, y, t)
- Learned to predict the motion in an implicit neural representation (INR)
- Use the learned motion in downstream video analysis tasks:
  - video segmentation,
  - video progress prediction.

Questions?