
SARAS CALL h2020-ICT-2016-2017
INFORMATION AND COMMUNICATION TECHNOLOGIES

# SARAS

"Smart Autonomous Robotic Assistant Surgeon"

## D6.1 – Real-time surgeon action detection and recognition

Due date of deliverable: 30/06/19
Actual submission date: 28/06/19

**Grant agreement number:** 779813
**Start date of project:** 01/01/2018
**Revision**

**Lead contractor:** Università di Verona
**Duration:** 36 months

| Project funded by the European Commission within the EU Framework Programme for Research and Innovation HORIZON 2020 | |
|---|---|
| Dissemination Level | |
| PU = Public, fully open, e.g. web | ✓ |
| CO = Confidential, restricted under conditions set out in Model Grant Agreement | |
| CI = Classified, information as referred to in Commission Decision 2001/844/EC. | |
| Int = Internal Working Document | |

# D6.1 – Real-time surgeon action detection and recognition

**Editor**
Fabio Cuzzolin

**Contributors**
Fabio Cuzzolin
Elettra Oleari
Alice Leporini

**Reviewers**
Elettra Oleari
Alice Leporini

The work described in this document has been conducted within the project SARAS, started in January 2018. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No.779813

## Executive Summary

The SARAS - Smart Autonomous Robotic Assistant Surgeon - project aims at developing a next-generation cognitive autonomous system for solo surgery, allowing a single surgeon to execute Robotic Minimally Invasive Surgery (R-MIS) without the need of an expert assistant surgeon.
Within SARAS, WorkPackage 6 concerns the detection and recognition (in real time) of what actions the main surgeon performs using standard laparoscopic tools, either manually or in tele-operation via a da Vinci master robot available at University of Verona. In particular, this Deliverable relates the results of the activities conducted under Task 6.1 (Online surgeon action recognition).
Starting from the existing codebase at partner OBU (Oxford Brookes University), SARAS' work on this Task has so far led to two publications in major computer vision conferences, proposing a tracking-inspired framework for the online detection of actions in videos and a transition matrix neural network for the flexible detection of such actions, respectively, as well a paper currently under review in the top computer vision journal (IEEE Transactions on Pattern Analysis and Machine Intelligence). In close collaboration with partner OSR (Ospedale San Raffaele) and UNIVR (University of Verona), groundbreaking work has been conducted to create the first dataset on surgeon action detection in laparoscopic videos, via the annotation of four endoscopic videos captured by OSR during RARP procedures on real patients. The models so learned will be transferred to the same tasks on the two SARAS demonstrator platforms.
Further work is ongoing on the design of networks able to detect entire action instances in a single go, through novel causal 3D convolutional neural network architectures, and the detection and recognition of complex activities, such as a laparoscopic procedure, a problem feeding directly into SARAS' Tasks 6.2 (Current procedure stage recognition) and 6.3 (Predicting future surgeon actions).

# Contents

## List of acronyms

OBU – Oxford Brookes University

OSR – Ospedale San Raffaele

CNN – Convolutional Neural Network

SSD – Single-Shot Detector

RNN – Recurrent Neural Network

LSTM – Long-Short Term Memory

RARP – Robotic Assisted Radical Prostatectomy

OR – Operating Room

RGB – Red/Green/Blue colour space

OF – Optical flow

## List of tables

## List of figures

# Introduction

## 1.1 *Purpose of the document*

This document aims at describing the progress made by the SARAS project under Task 6.1 (Online surgeon action recognition), part of WorkPackage 6, which concerns the detection and recognition in real time of what actions the main surgeon performs using standard laparoscopic tools, either manually or in tele-operation via a da Vinci master robot available at University of Verona.

The input to this component of SARAS is the video streaming in from the available laparoscopic camera. Its output is, for each video frame, a number of bounding boxes showing where the various actions of interest are taking place, with attached scores (produced by a neural network) for each action class. This is exemplified by Figure 1, where two bounding boxes are shown, that relate to two actions of interest ('bladder anastomosis' and 'pulling tissue').

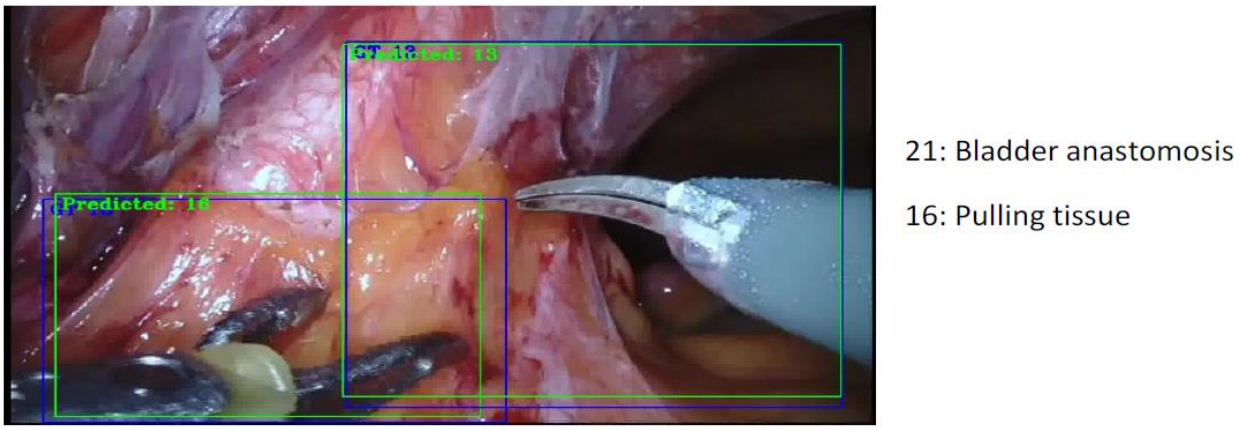


*Figure 1: Two example detections for a video frame provided by OSR, related to action classes "bladder anastomosis" and "pulling tissue". Ground truth detections, manually provided to train the neural network model are shown in blue. Network predictions are shown in green.*

# Deep learning for real-time action detection

## 2.1 *The problem*

The problem of recognising human actions (such as those performed by the surgeon through their tele-operated tools, see Figure 1) from video streams (such as those coming from an endoscope) is in fact very challenging, as different surgeons may have different styles of execution, while viewpoint variations (due to the camera shooting the cavity from a slightly different position) and occlusions may further complicate recognition.

Traditionally, within computer vision, videos are processed as a whole, so that no decision upon what class of movement is observed can be drawn in real time (instant by instant). This is completely unacceptable for robotic assistant systems such as SARAS', which require a prompt, real-time interpretation of what is taking place within the surgical cavity. Although some work was done in this sense in the past, the systems that were available up until the end of 2017 were only tailored for the recognition of a single type of action per video frame. Some, while able to tackle incremental, online recognition, could not concurrently locate the action of interest in each video frame (in particular in the form of a bounding box around the action of interest, as shown in Fig. 1).

As of 2016-17, before SARAS started, the state of the art computer vision methods in action detection were still inherently offline, as they relied on detecting instances of actions of interest in a frame-by-frame fashion, using, for instance, a Faster R-CNN Region Proposal Network [1], to then link them up into the desired 'action tubes' by means of a post-processing step.

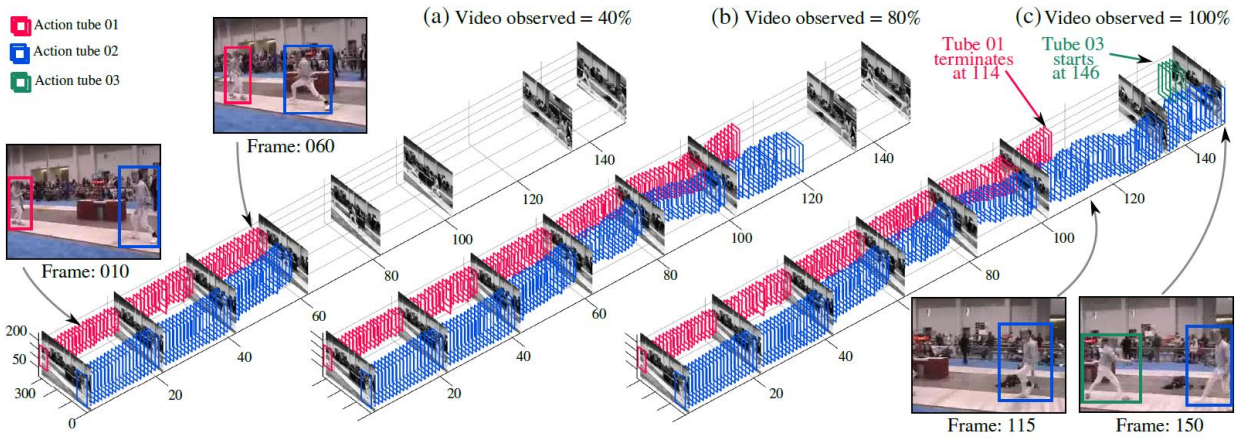


*Figure 2: Example of action tubes generated from a video, and some sample frame-level detections. As the fraction of the video observed increases from 40% (a) to 80% (b) to 100% (c) the system incrementally builds the action tubes (series of bounding box detections) associated with the various action instances, and determines their starting and ending times.*

## 2.2 *Deep learning for detecting action tubes*

Current best practice in action detection is based on 'deep' neural networks [2], i.e., artificial neural networks composed by a significant number of layers and architectures geared to efficiently support learning at various levels of abstraction. In particular, this is achieved thanks to a specialised connectivity structure in which (unlike in traditional artificial neural networks) all

neurons are not connected to all others, but (as is the case for convolutional neural networks, CNNs) follow a local pattern.

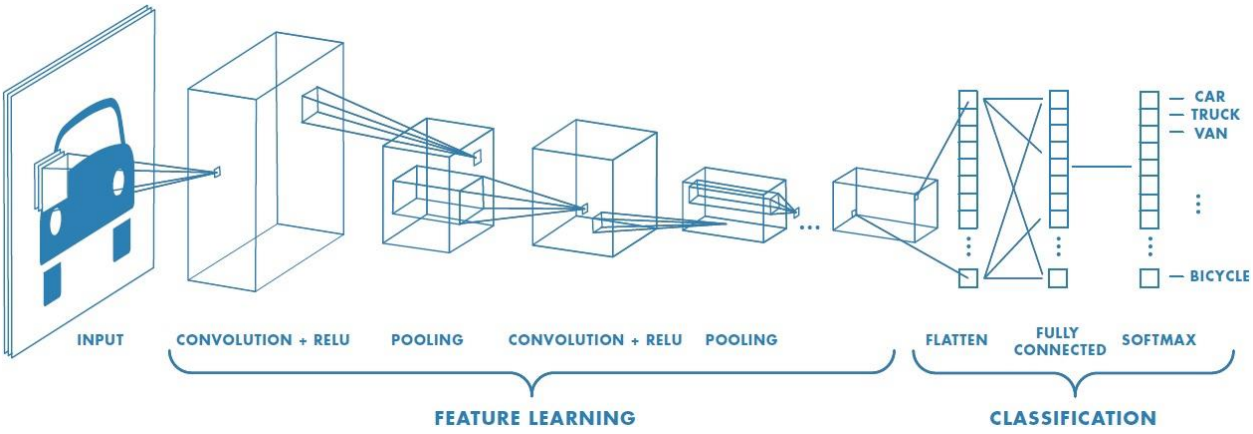


*Figure 3: General structure of a convolutional neural network.*

For instance, in CNNs (see Figure 3) the main structures are *convolutional layers* in which each local patch in the output of the previous layer (feature map) is processed using the mathematical operation of convolution. Crucially, all local patches at the same level of the hierarchy are processed with the same convolution kernel (in network terminology, they "share weights"). Convolutional layers are alternated with *max pooling* ones which summarise each local patch with a single real number, and non-linear ones associated with *activation functions*.

Common activation functions are the *sigmoid*

$$f(x) = \frac{1}{1 + e^{-x}}$$

and the *rectifier linear unit (ReLU)* ones:

$$f(x) = \max(0, x).$$

As the number of network parameters is drastically reduced, all these desirable features make it possible to train the resulting deep network, consistently achieving state of the art results.

Given a number of example videos (with bounding boxes showing where each action takes place), appropriate deep networks can indeed be designed to learn to both regress the location of bounding boxes containing actions of interest, and to provide a score for each action, as desired.

The dominant paradigm until 2017 was for the deep network to perform the detection separately for each video frame, to then link up these detections in time to form what researchers term 'action tubes' (see Figure 2).

## 2.3 *OBU's existing real time platform*

In recent years OBU has built a leadership position in the field of deep learning for real-time action detection, localisation and recognition, with the best detection accuracies to date and the only system able to localise multiple actions on the image plane in (better than) real time [3]. Superseding recently proposed methodologies, the approach in [3] is able to handle simultaneous localisation in space and time of multiple action instances and to perform detection and recognition in a completely real time fashion. Figure 4 shows the components of the real-time platform for action detection from streaming videos, developed by OBU in 2017 and published at ICCV'17, the International Conference of Computer Vision [3].

The approach processes the video input in two separate streams, one associated with the raw, RGB video frames ('appearance'), and the other with *optical flow* [4], a vector field which, for each pixel location, expresses where the pixel is going to move to in the next video frame (see Figure 4 (d)). Subsequently, SSD (Single Shot Detector) [5] is used as detection network (e) to generate bounding box detections and class scores (f). Detections associated with the two streams are fused using a simple strategy (g). Finally, an online action tube generation algorithm (h) incrementally grows multiple tubes, for each action, over time. Tubes are temporally trimmed (their start and end instant are determined) using an online Viterbi approach [3].

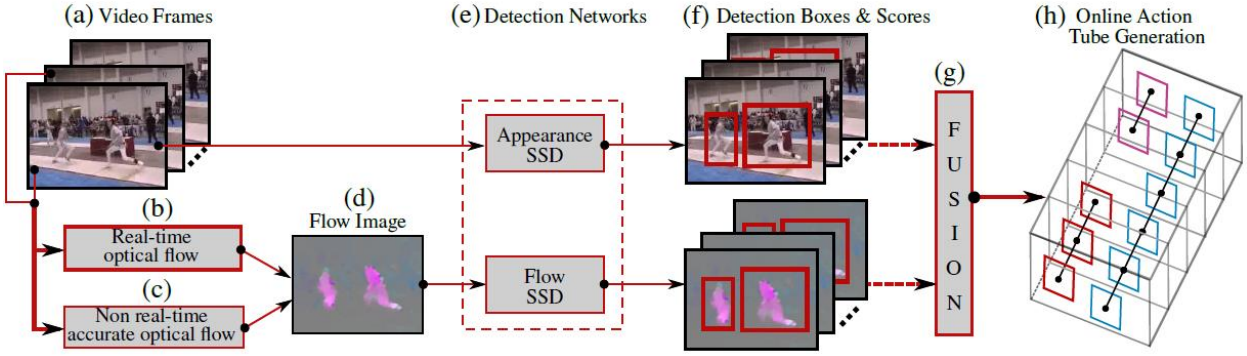


*Figure 4: OBU's deep learning module for the real time localisation and recognition of human actions. At test time, the input to the framework is a sequence of RGB video frames (a). A real-time optical flow (OF) algorithm (b) takes the consecutive RGB frames as input to produce flow images (d). As an option, (c) a more accurate optical flow algorithm can be used (although not in real time). (e) RGB and OF images are fed to two separate SSD detection networks). (f) Each network outputs a set of detection boxes along with their class-specific confidence scores. (g) Appearance and flow detections are fused. Finally (h), multiple action tubes are built up in an online fashion by associating current detections with partial tubes.*

This methodology is still somewhat unsatisfactory, for it relies on frame-by-frame detections, rather than estimating entire action instances (tubes) at any given moment in time. However, it exhibits some early prediction abilities, i.e., the network is able to predict with a good degree of confidence what the final label of the action instance is going to be, after observing only a fraction of the constituting video frames (Figure 6). Some example action localisation results on UCF-101 are shown in Figure 5.


*Figure 5: Sample action localisation results on UCF-101. Each row represents a UCF-101 test video clip. Ground-truth bounding boxes are drawn in green and detection boxes are in red.*

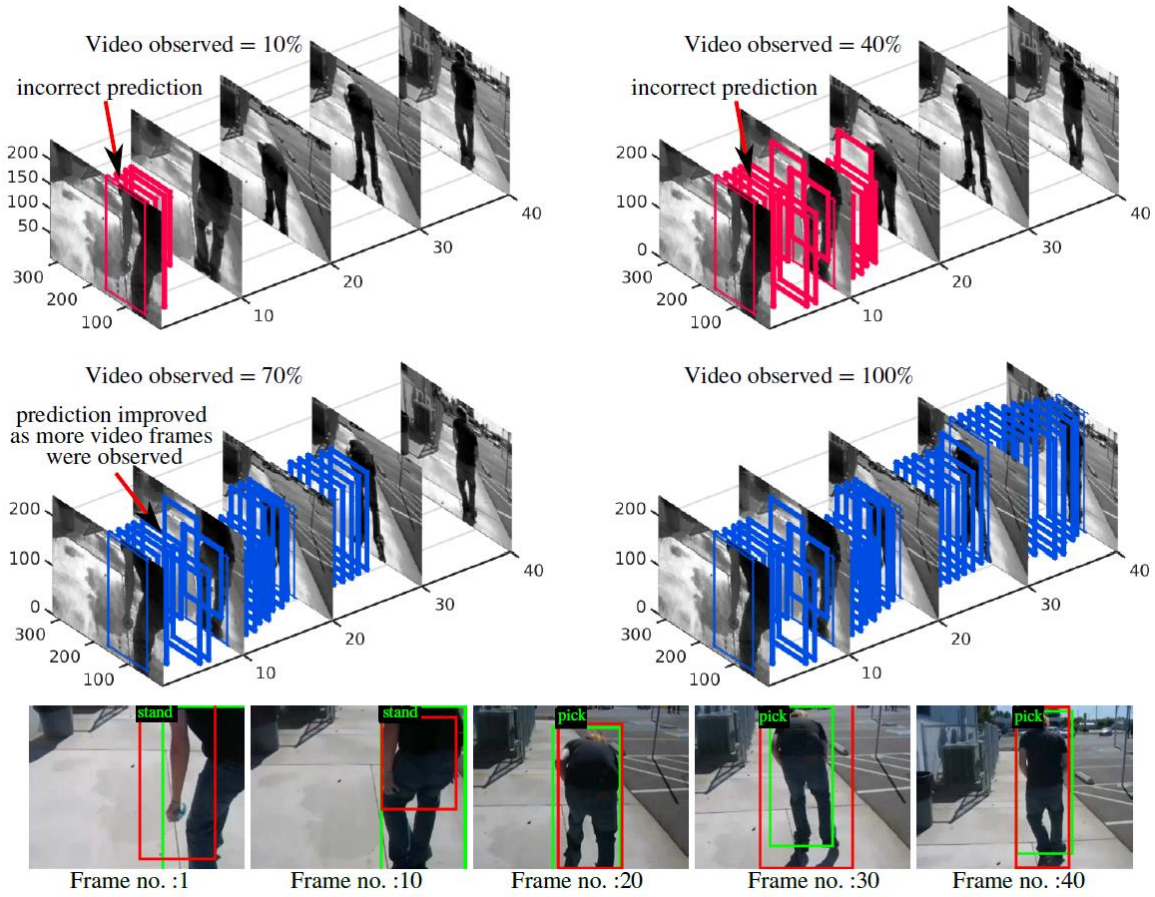


*Figure 6: Sample early action label prediction and online action localisation results of [3] on the J-HMDB-21 dataset. The test video contains an instance of the 'pick' action class. The video and its corresponding space-time detection tube are plotted in 3D at different time points (i.e., % of video observed). Detection tubes are drawn in two different colours to indicate a wrong early label prediction and the improved prediction, respectively, as more video frames are observed in time. At the bottom, the predicted action labels for the same video at different time points are overlaid on the corresponding video frames. Green boxes depict the ground-truth, while red ones depict the predicted bounding boxes.*

## 2.4 *From frame-level detections to micro-tubes*

The first step towards the above objective consists in moving from single-frame detections to *pairs* of detections – which in [6] we termed 'micro-tubes'.

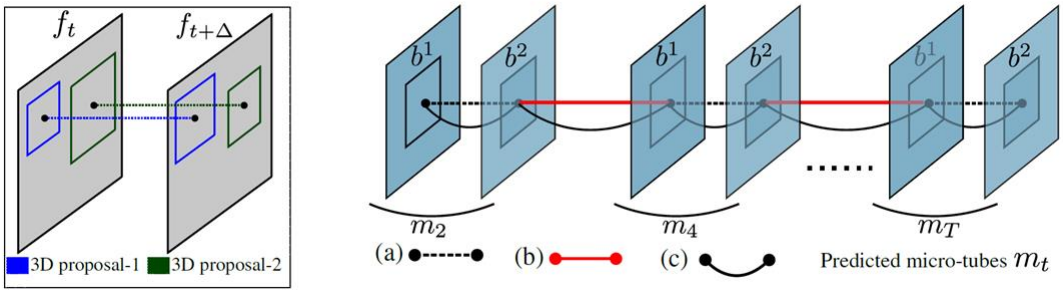


*Figure 7: Left: 3D region proposals. Right: linking of successive micro-tubes in time.*

In another paper OBU published at ICCV'17 [6], the team proposed a newly designed 3D Region Proposal Network able to regress such micro-tubes. The concept of 3D region proposal as a pair of bounding boxes lying within two successive video frames is shown in Figure 7 – Left. As shown in Figure 7 – Right, at test time micro-tubes still needs to be linked up in time for form a complete

action tube. In [6], this was done through a tube generation algorithm similar to the one proposed in [3]. The pipeline of the approach is illustrated in Figure 8.

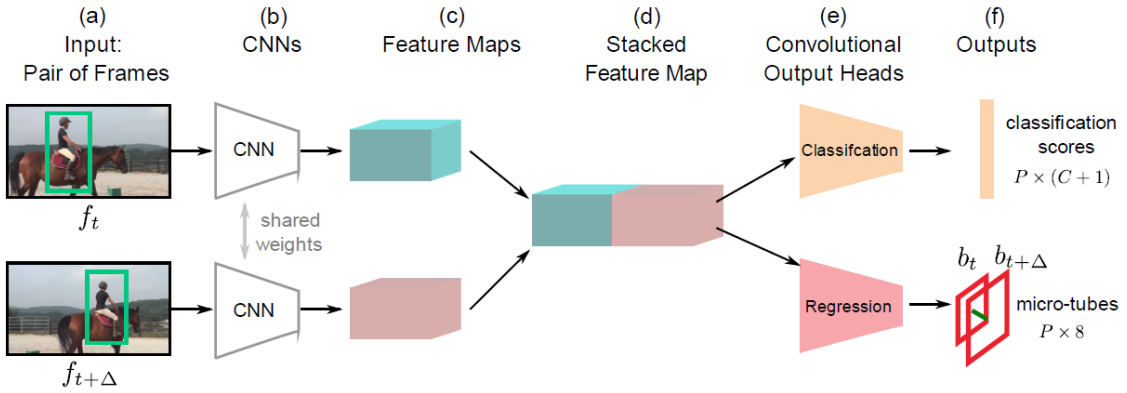


*Figure 8: Action micro-tube detection.*

Input pairs of frames (a) are each processed by two standard 2D CNNs (b) (adopting the popular VGG architecture [7]), generating two feature maps (c). The latter are combined (stacked) into a single overall feature map (d), which is sent to two convolutional heads, one for classification and one for regression (e). These output, respectively, $P \times (C + 1)$ classification scores and the corresponding micro-tubes (f).

The architecture is illustrated in more detailed in Figure 9 – more technicalities can be found in [6]. One can note the central role of the proposed 3D-RPN (Region Proposal Network) in (d).

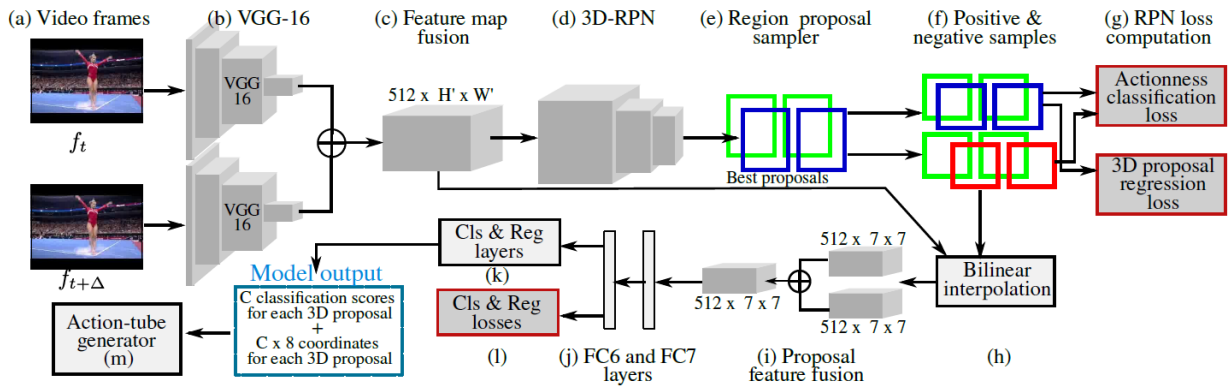


*Figure 9: Details of the AMTnet architecture – details can be found in [6].*

## 2.5 *Benchmark datasets*

A number of standard benchmark datasets are commonly employed for evaluating action detection results.

*UCF101-24* is a subset containing 24 classes of the UCF101 [8] dataset, which itself encompasses 101 classes. Its initial spatial and temporal annotations provided in the THUMOS-2013 action detection challenge [9] were later corrected by Singh et al. [3] – we use this version in all experiments reported here. Each UCF101 video contains a single action category; sometimes multiple action instances of the same category are present in the same video. Each action instance covers, on average, 70% of the video duration.

J-HMDB-21 is a subset of the HMDB-51 dataset [10] comprising 21 action categories and 928 videos, each containing a single action instance and trimmed to the action's duration.

The *DALY dataset* was released by Weinzaepfel et al. [11] for 10 daily activities, and contains 520 videos (200 for test and the rest for training) for a total of 3.3 million frames. Videos in DALY are much longer, and the action duration to video duration ratio is only 4% compared to UCF101-24's 70%, making the temporal labelling of action tubes very challenging. The most interesting aspect of this dataset is that it is not densely annotated, as at max 5 frames are annotated per action instance, and 12% of the action instances only have one annotated frame. As a result, annotated frames are 2.2 seconds apart on average.


*Figure 10: Example frames and detection from the LIRIS-HARL dataset.*

The *LIRIS-HARL dataset* [13] contains 10 action categories, including human-human interactions and human-object interactions (e.g., 'discussion of two or several people', and 'a person types on a keyboard'2). In addition to containing multiple space-time actions, some of which occurring concurrently, the dataset contains scenes where relevant human actions take place amidst other irrelevant human motion.

## 2.6 *Performance metrics*

The following performance measures are standard in action detection for assessing the results of an approach at test time (i.e., when the model/network trained on a certain amount of training data is employed to detect and classify actions in new, test sequences).

*Accuracy* is simply the percentage of correctly classified instances, expressed in %.

*Average precision* (AP) does not simply focus on the percentage of misclassified examples, but calculates, for each class, both the percentage of instances correctly classified as positive over the total of those classified as positive, known as *precision*, and the rate of positive instances correctly recognised as such, known as *recall*. Formally, let us define:

- TP = true positives, as the number of times a positive class prediction is attached to an actual positive instance;
- FP = false positives, as the number of instances whose class is incorrectly predicted as positive when its true value is negative;
- FN = false negatives, as the number of times the class value is incorrectly predicted as negative when its true value is positive;
- TN = true negatives, as the number of times a negative class prediction is correctly attributed to an actual negative class instance.

Precision and recall are then defined as follows (see Figure 11):

$$Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN}.$$

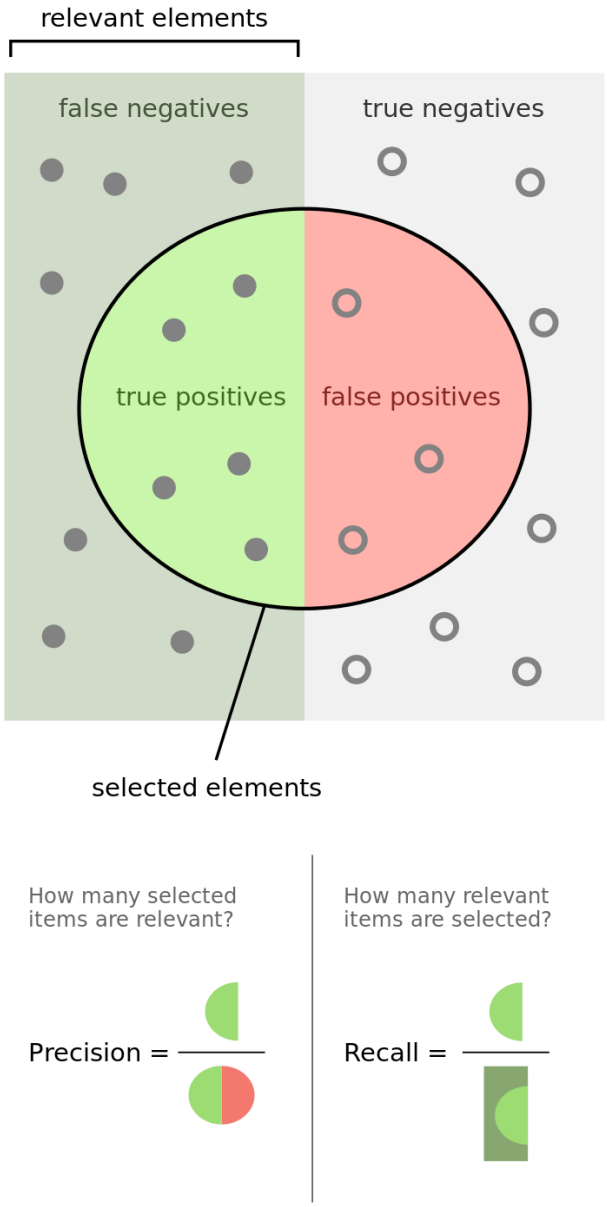


*Figure 11: Precision versus recall in binary classification.*

*Average precision*(AP) is created by plotting precision against recall, yielding a precision-recall curve, and then integrating the area under the curve.

*Mean Average Precision* (mAP) over a set of query points, is the mean of the AP scores for each query. For the action tube detection problem, therefore, the *Frame-mAP* value over an action tube is simply the mean of the AP values for each of the individual frames. The *Video-mAP* value, instead, Is computed over action tubes intended as instances.

## Progress made by SARAS

Since 2018, as part of SARAS, OBU has made further significant progress towards (1) the design of a deep neural network framework able to regress any number of entire action tubes in real time, and (2) the testing of our architectures on laparoscopic videos depicting relevant surgeon actions as they appear in the two procedures of interest to SARAS (nephrectomy and prostatectomy).

The technical progress made is described in this section. The generation of a SARAS dataset of endoscopic videos depicting laparoscopic prostatectomy procedures, and the results obtained so far on the new dataset, are described in the following section.

### 3.1 *Incremental tube construction approach*

In joint work between OBU's Visual AI Lab and Oxford University's Torr Vision Group, Behl et al. [12] proposed a novel linking algorithm called OJLA (*Online Joint Labelling and Association*), that is able to construct and update action tubes as each new frame is added (see Figure 12).

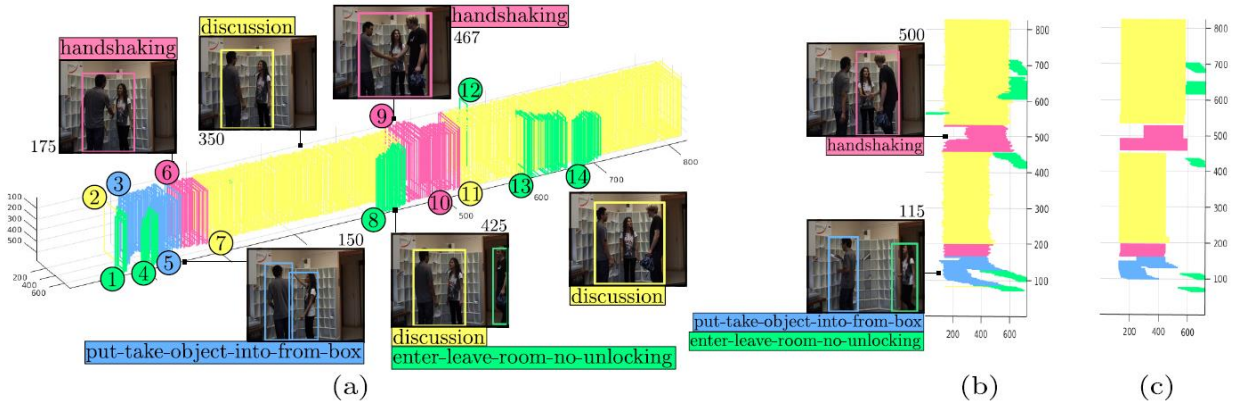


*Figure 12: (a) Illustrative results of OJLA on a video sequence from the LIRIS-HARL dataset [13]. Two people enter a room and put/take an object from a box (frame 150). They then shake hands (frame 175) and start having a discussion (frame 350). In frame 450, another person enters the room, shakes hands, and then joins the discussion. Each action tube instance is numbered and coloured according to its action category. We selected this video to show that our tube construction algorithm can handle very complex situations in which multiple distinct action categories occur in sequence and at concurrent times. (b) Action tubes drawn as viewed from above, compared to (c) the ground truth action tubes.*

Unlike [3,6], the approach is based on the formulation of a novel cost function which solves all of these tasks jointly and incrementally in a single pass, in a multi-target tracking framework. This implies that the algorithm does not perform action detection separately for each class. For scenarios where only one human action is taking place in a space-time location, which is the case in the standard action detection benchmarks UCF-101, JHMDB-21 and LIRIS-HARL [13], the approach outputs several human-centered (non-overlapping) action tubes, where each tube can take a single label. This avoids the problem of detecting multiple co-located action tubes with different classes.

Qualitative results of OJLA can be appreciated in Figure 13, which support the ability of the method to discriminative between very similar activities, which differ by a small but important detail (*fine grained* discrimination).


*Figure 13: Sample qualitative results of OJLA on the LIRIS-HARL dataset [12]. First (top), a woman walks into a room, whilst a man stands in front of a whiteboard. The two people then 'shake hands' and start a 'discussion'. Notice how our algorithm is able to handle situations in which multiple actions occur concurrently and/or sequentially. Next (middle) a person 'enters/leaves a room without unlocking', then 'puts-takes an object from a box', and again 'enters/leaves a room without unlocking'. Finally (bottom) a man holds a 'telephone conversation'; again the system mislabels the beginning and end of the action by detection a 'put/take object into/from box' action immediately preceding and following the 'telephone conversation'.*

## 3.2 *A transition matrix network*

Further progress was made in November 2018, with the publication by OBU of "TraMNet - Transition Matrix Network for Efficient Action Tube Proposals" [14], as a direct evolution of AMTnet [6]. Compared to the latter, the paper introduces the concept of learning, at training time, how likely are action instances (in the form of bounding boxes) to shift from one location in the video frame to another. This is done through a new Transition Matrix Network, in which features are pooled from different regions of the pair of video frames considered according to the (transition) probability of an action instance shifting from one image location to another.

As a result, when compared to AMTnet, TraMNet is able to model more flexibly micro-tubes of any arbitrary shape (see Figure 14), as opposed to those of cuboidal shape handled by the former (and similar state of the art papers [15,16]). In Figure 14, a horse rider changes its location from frame $f_t$ to $f_{t+\Delta}$ (a), as shown by the ground truth bounding boxes (in green). As the micro-tube proposal is constrained by the location of the anchor box in the first frame, the overall spatiotemporal IoU overlap between the ground-truth micro-tube and the proposal is relatively low. (b) In contrast, the anchor micro-tube proposal generator in [14] is much more flexible, as it efficiently explores the video search space via an approximate transition matrix estimated based on a hidden Markov model (HMM) formulation. As a result, the anchor micro-tube proposal (in blue) generated by

TraMNet exhibits higher overlap with the ground-truth. (c) For "static" actions (such as "clap") in which the actor does not change location over time, anchor cuboid and anchor micro-tubes have the same spatiotemporal bounds.

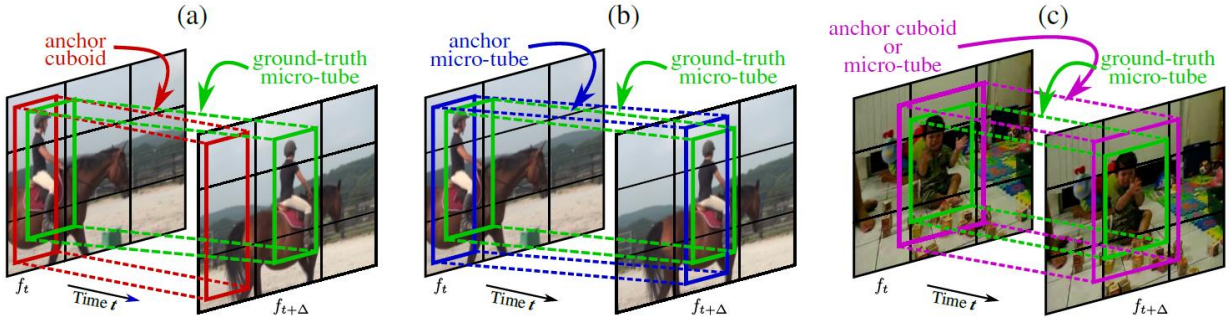


*Figure 14: TraMNet can model arbitrary pairs of detections, as opposed to just cuboidal ones.*

This is made possible by the architecture depicted in Figure 15.

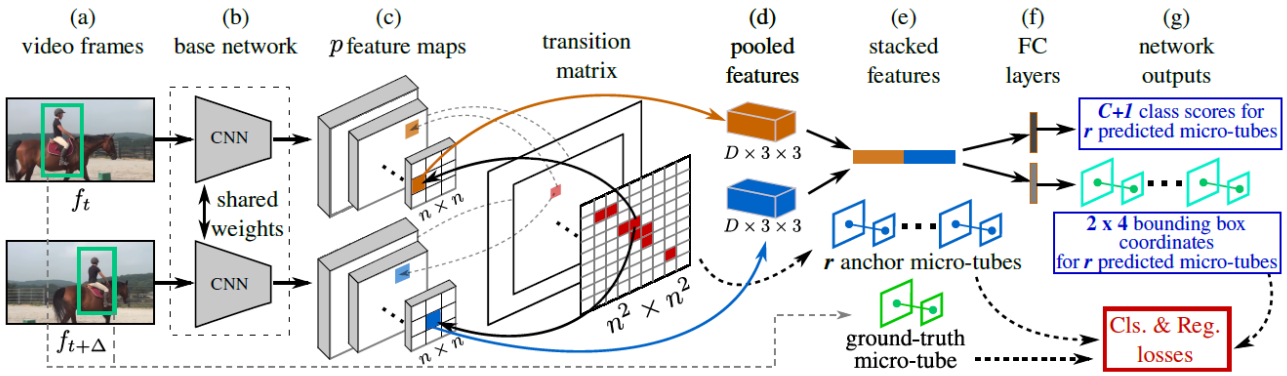


*Figure 15: TraMNet architecture. Details are described in the text.*

The proposed network takes as input (a) a pair of successive video frames $f_t$ and $f_{t+\Delta}$ (where $\Delta$ is the inter-frame distance) and propagates them through a base network comprised of two parallel CNN networks (b), which produce two sets of $p$ convolutional feature maps forming a pyramid (i.e., grid locations at various scales are considered when computing features). These feature pyramids are used by a reconfigurable pooling layer (d) to pool features based on the transition probabilities learned at training time, collected in a transition matrix $A$. The pooled conv features are then stacked (e), just as in AMTnet, and the resulting feature vector is passed to two parallel fully connected linear layers (one for classification and another for micro-tube regression (f)), which predict the output micro-tube and its classification scores for each class $C$ (g).

We call TraMNet "reconfigurable" because the configuration of the pooling layer (d) depends on the transition matrix $A$.

Experimental results on the standard UCF-101-24 action detection dataset (Table 1) show that the approach outperforms existing methods based on individual or pairs of frames [14]. The numbers reported are video-mAP figures, except for the last column (which reports accuracy in %), and the

second last column which reports the average video-mAPs achieved by the various methods for detection overlap (IoU) between 0.5 and 0.95.

| Methods | IoU=0.2 | IoU=0.5 | IoU=0.75 | IoU=0.5 : 0.95 | Acc % |
|---|---|---|---|---|---|
| T-CNN [16] | 47.1 | — | — | — | — |
| MR-TS | 73.5 | 32.1 | 2.7 | 7.3 | — |
| [1] | 66.6 | 36.4 | 7.9 | 14.4 | — |
| [3] | 73.2 | 46.3 | 15.0 | 20.4 | — |
| AMTnet [6], RGB only | 63.0 | 33.1 | 0.5 | 10.7 | — |
| ACT [15] | 76.2 | 49.2 | 19.7 | 23.4 | — |
| Gu et al [17] | — | **59.9** | — | — | — |
| TraMNet | **79.0** | 50.9 | **20.1** | **23.9** | 92.4 |

*Table 1: Action localisation results on untrimmed videos from UCF101-24. Top performance highlighted in bold.*

As reported in Table 2, results on the DALY dataset show that TraMNet is able to handle sparse annotations better than AMTnet, which uses anchor cuboids, strengthening the argument that learning transition matrices helps generate better micro-tubes. For instance, TraMNet's performance on the action class 'CleaningFloor' at IoU equal to 0.5 highlights the effectiveness of general anchor micro-tubes for dynamic classes. 'CleaningFloor' is one of DALY's classes in which the actor moves spatially while the camera is mostly static.

| Methods | IoU=0.5 | IoU=0.5 : 0.95 | Acc % | Clearning Floor |
|---|---|---|---|---|
| [11] | 63.9 | — | — | — |
| [3] | 63.9 | 38.2 | 75.5 | 80.2 |
| AMTnet [6] | 63.7 | 39.3 | 76.5 | 83.4 |
| TraMNet | **64.2** | **41.4** | **78.5** | **86.6** |

*Table 2: Action localisation results (video-mAP) on the DALY dataset.*

## 3.3 *Two-stream AMTnet*

In a paper under review by the IEEE Transactions of Pattern Analysis and Machine Intelligence [18], OBU's team has proposed a further evolution of AMTnet approach, by introducing:

1. the integration of optical flow in the AMTnet architecture (which originally only considered appearance in the form of RGB frames);
2. the end-to-end training of both appearance and motion streams;
3. the train time (as opposed to test time) feature fusion of RGB and optical flow information;
4. an online (as opposed to offline, as in [6]) algorithm which incrementally builds action tubes by linking micro-tubes in time;
5. a network architecture geared, in general, towards faster detection speed.

In particular, AMTnet's original Faster R-CNN-based [19] network architecture is replaced by a comparatively faster SSD [20] network design. The SSD architecture minimises computational cost by eliminating the need for a separate region proposal network. As in previous work, optical flow fields relating successive video frames are computed using [4].

We termed the overall architecture "Two-Stream AMTnet" - illustrated in Figure 16.

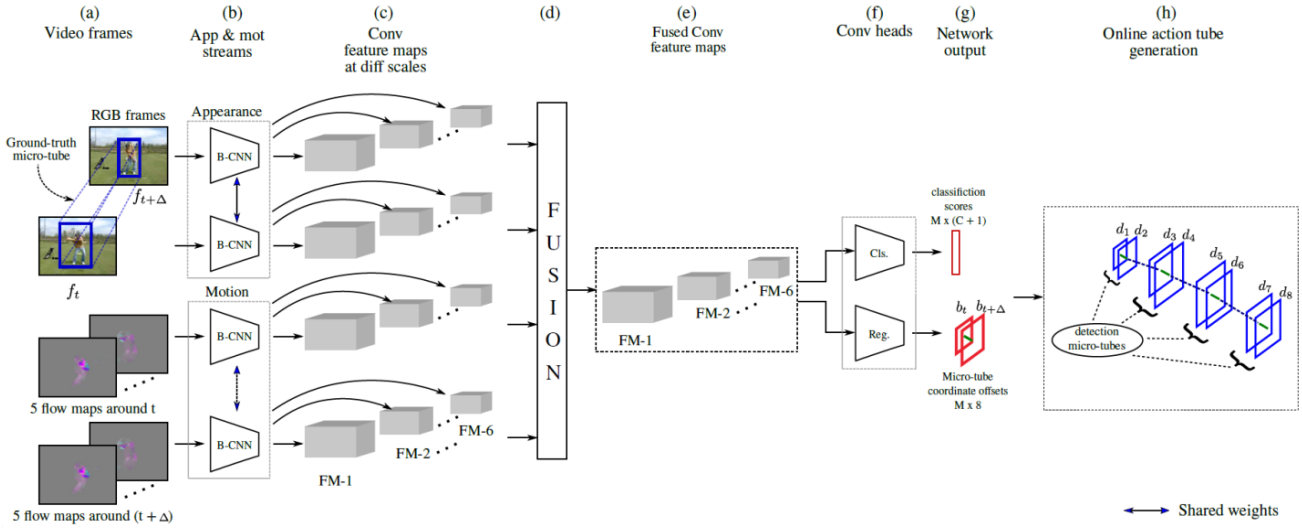


*Figure 16: Two-stream AMTnet architecture [18]. Details can be found in the text.*

The input to the network is a pair of successive (but not necessarily consecutive) RGB video frames $f_t$ and $f_{t+\Delta}$, and the corresponding stacked optical flow maps (a). These RGB and flow frames are propagated through their respective appearance and motion streams (b). The latter output convolutional feature maps at 6 different spatial scales (c), which are passed through to a fusion layer (d) where they are merged. The resulting fused conv feature maps (e) are then passed as inputs to a classification (cls) layer and a regression (reg) layer (f). The cls layer outputs $M \times (C + 1)$ softmax scores for $M$ micro-tubes and $C$ action classes, whereas the reg layer outputs $M \times 8$ bounding box coordinate offsets corresponding to M micro-tubes (g). At test time, the outputs of the cls and reg layers are passed to an online action tube generation algorithm (h), which incrementally builds action tubes by linking the detected micro-tubes in time.

Crucially, at test time, once the network has outputted a micro-tube spanning detections separated by Δ frames, in this approach the intermediate detections are obtained by linear interpolation (see Figure 17).

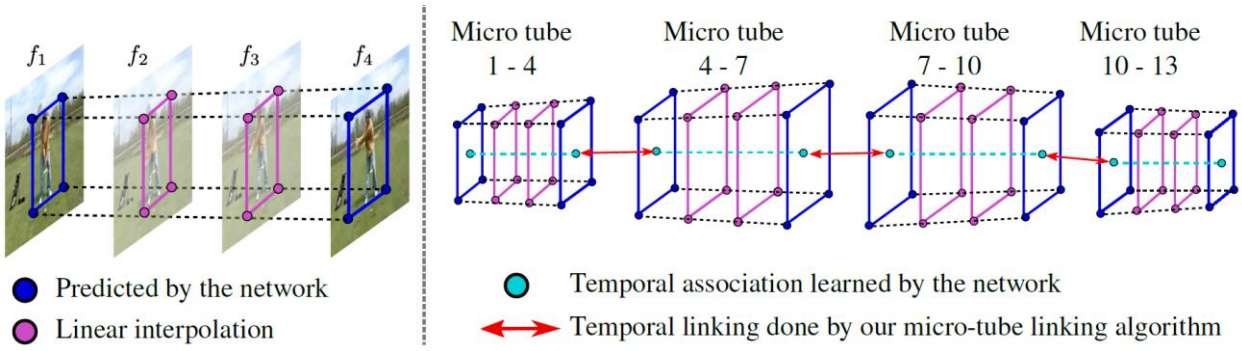


*Figure 17: Linear interpolation of intermediate detections in a micro-tube.*

The experimental results, reported in Tables 3 and 4, show out Two-Stream AMTnet consistently outperforms all existing online competitors, and offline ones as well on UCF-101, on the two most commonly accepted action detection benchmarks, JHMDB-21 and UCF-101-24.

| Methods | IoU=0.2 | IoU=0.5 | IoU=0.75 | IoU=0.5 : 0.95 | Acc % |
|---|---|---|---|---|---|
| MR-TS [21] | 74.1 | 73.1 | — | — | — |
| Saha et al, BMVC'16 [1] | 72.2 | 71.5 | 43.5 | 40.0 | — |
| OJLA [12] | — | 67.3 | — | 36.1 | — |
| Singh et al, ICCV'17 [3] | 73.8 | 72.0 | 44.5 | 41.6 | — |
| AMTnet [6], RBG only | 57.7 | 55.3 | — | — | — |
| ACT [15] | 74.2 | 73.7 | 52.1 | 44.8 | 61.7 |
| T-CNN [16] (offline) | 78.4 | 76.9 | — | — | |
| RTPR (offline) | **82.3** | **80.5** | — | — | — |
| **Two-Stream AMTnet [18]** | 73.5 | 72.8 | **59.7** | **48.3** | **69.6** |

*Table 3: Action localisation performance of Two-Stream AMTnet on JHMDB-21. Top performances in bold.*

| Methods | IoU=0.2 | IoU=0.5 | IoU=0.75 | IoU=0.5 : 0.95 | Acc % |
|---|---|---|---|---|---|
| MR-TS [21] | 73.7 | 32.1 | 0.9 | 7.3 | — |
| Saha et al, BMVC'16 [1] | 66.6 | 36.4 | 7.9 | 14.4 | — |
| OJLA [12] | 68.3 | 40.5 | 14.3 | 18.6 | — |
| Singh et al, ICCV'17 [3] | 76.4 | 45.2 | 14.4 | 20.1 | 92.2 |
| AMTnet [6], RBG only | 63.1 | 33.1 | — | 10.4 | — |
| ACT [15] | 76.5 | 49.2 | 19.7 | 23.4 | — |
| T-CNN [16] (offline) | 47.1 | — | — | — | — |
| RTPR (offline) | 76.3 | — | — | — | — |
| **Two-Stream AMTnet [18]** | **79.7** | **49.7** | **22.2** | **24.1** | **92.3** |

*Table 4: Action localisation performance of Two-Stream AMTnet on UCF101-24. Top performances in bold.*

# The SARAS surgical action detection dataset

An MSc dissertation conducted by OBU student Francis Kaping'A allowed us to produce very promising preliminary results on the application of OBU's action detection technologies, illustrated above, to surgical action detection from laparoscopic videos. Joint effort with Ospedale San Raffaele (OSR), another SARAS partner, has been directed towards generating a new benchmark dataset for the detection and recognition of surgical actions, by annotating four real-life endoscopic videos captured during RARP (prostatectomy) laparoscopic procedures provided by OSR.

Sensitive data used by San Raffaele Hospital (images and footage of radical prostatectomies) in the SARAS project was lawfully collected through explicit consent of the data subjects (legal bases: Article 6(1)(a), Article 9(2)(a)) for a previous observational study. The latter was approved, through a specific research protocol, by the Research Ethics Committee of San Raffaele Hospital, composed of more than forty members and currently evaluating more than 30 protocols/month (see Appendix, "**SARAS PROJECT – ETHICS AND DATA COMPLIANCE DOCUMENT**").

Moreover, all surgical videos were anonymized before starting the necessary annotation work.

## 4.1 *The data*

The four videos depict a Robotic Assisted Radical Prostatectomy (RARP), which is the resection of the whole prostate gland in patients with prostate cancer, with a secondary aim of preserving urinary continence and erectile function. This intervention is the gold standard for Robotic-Assisted surgeries (for more surgical information, please see Deliverable D1.1).

The surgical team present in the Operating Room (OR) is composed by: the main surgeon, operating at the da Vinci console; a surgical assistant (usually a trained urology resident), operating at the surgical table with laparoscopic tools. The duration of RARP surgery is about 3 to 4 hours.

The surgical area is accessed through small incisions in the abdomen and the use of trocars. However, in this case the first surgeon controls an advanced robotic system capable of moving surgical tools from outside the body. A high-tech interface lets the surgeon use natural wrist movements and a 3D screen during the entire operation. One of these trocars is placed over the umbilicus for camera port. This involves inserting a fibre-optical instrument and some other operating instruments into the patient's abdomen. The camera streams video data to the operator's console, where it is used to have a view of the patient's abdomen.

Two possible approaches exist as to how to access the surgical area within RARP: (i) the *transperitoneal* approach, with access to the abdomen, and the (ii) *extraperitoneal* one, with pelvic access. Most RARPs are executed through the transperitoneal approach, which is indeed the situation described in the SARAS procedural workflow.

Within the transperitoneal approach itself, two different modalities for reaching the target organs during RARP exist. (i) In the *anterior* modality, after transperitoneal access and insufflation, the space of Retzius is immediately entered and the prostate gland, seminal vesicle, and vasa are reached and dissected from the front. (ii) In the *posterior* modality, the seminal vesicles and vasa are initially reached and completely dissected behind the bladder.

The videos selected for the SARAS surgical action datasets concern the RARP *posterior* approach, for this procedure is routinely performed in the clinical practice by the expert urological surgeons of Ospedale San Raffaele.

It has to be noted, however, that for the SARAS simplified RARP to be executed through the project demonstrator it was necessary to select the transperitoneal *anterior* approach (see D1.1 Paragraph 2.1.3.4). This change was dictated by pre-testing evidence on the robotic platform and phantoms and, in accordance with OSR surgeons, is aimed to enlarge the surgical working space and to optimize the anatomical reconstruction of the phantom. As described below, transfer learning techniques will be employed to re-use the models learned from videos of RARP following the posterior approach on demonstrator data portraying the anterior approach.

## 4.2 *Annotation process*

To allow the training of the neural network architectures illustrated above, the videos need to be annotated by manually providing bounding boxes around the actions of interest in each video frame, and by inputting the class label associated with each bounding box. For this task, after comparing various options including the array of available Matlab Toolboxes, we elected the Microsoft virtual Object Tagging tool [22].

### 4.2.1 The Microsoft Virtual Object Tagging Tool

The video annotation procedure is illustrated in Figure 18, using a screenshot of the graphical user interface of the Virtual object Tagging Tool (VoTT). VoTT is a Microsoft open source tool used for drawing bounding boxes around regions of interest in visual data.

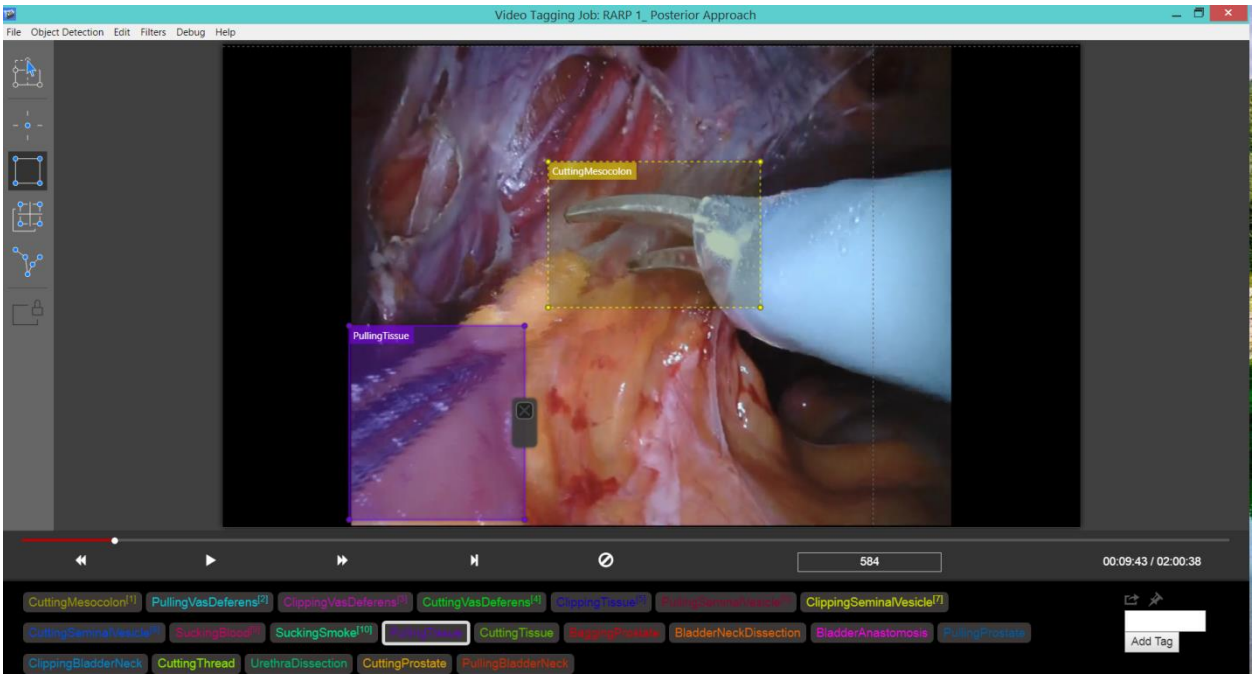


*Figure 18: Annotation process via the Microsoft Virtual object Tagging Tool (VoTT). As many bounding boxes as required are drawn using the graphical user interface – each bounding box is attached one or more labels from a pre-defined list (bottom). Each label is shown in a different colour to help with the annotation process.*

Annotation, for an action detection dataset, consists in locating action instances in each selected video frame via bounding boxes, and attaching one or more ground truth labels to each bounding box (see Figure 18). Initially, only one video of the four provided by OSR was annotated. The original video was captured at a rate of 50 frames per second. Annotation, on the other hand, was done at rate of 4 frames per second. The reason is that using a higher rate (say, 10 frames/second) would have unnecessarily prolonged the annotation work, as surgical actions do not typically change at such a quick pace, as we could observe from the videos.

### 4.2.2 Issues with the annotation process

A number of issues arise during the annotation process. To begin with, what constitutes an action or an event of interest is somewhat unclear. Some researchers have considered for this purpose surgical tools tracking methodologies [23], but for action detection and classification this would evidently cause the model to focus too much on the tools, rather than on what happens in the surgical cavity. As a result, the model would detect many false actions whenever a tool appears in the field of view. The same could happen when focussing on tissue strands or organs. We therefore decided to explore a combination of both organs and tools when setting the list of actions of interest and their descriptions. As a result, bounding boxes were drawn only when tools were close to the appropriate organs in order to deliver the identified actions of interest.

The question of what is the ideal size of a bounding box also arises. To balance the presence of tools and organs or tissue in a bounding box, bounding boxes were restricted to containing 30%-70% of either tools or organs.

Finally, to address the issue of determining the temporal extent of each action, a decision was made to only begin an action when a tool was close enough to the appropriate organ. Indicating the presence of an action when a tool is still far from the organ it is intervening on can potentially be very misleading, if the purpose is for SARAS' assistive arms to react appropriately.

Overall, the annotation task is subject to the inherent ambiguity of discriminating visually similar classes. For instance, it is hard to tell whether the aspirator is sucking blood, pushing some organs to make way, or sucking smoke. This was mitigated by seeking expert knowledge, which was provided by OSR's Dr Armando Stabile.

### 4.2.3 List of clinical actions analysed

The list of relevant clinical actions, decided in consultation Dr Stabile, contemplates 35 classes of surgeon actions. These are listed below, together with the number of instances present in the first video, originally analysed by student Francis Kaping'A.

| Class ID | Class name | Examples per class | Class ID | Class name | Examples per class |
|---|---|---|---|---|---|
| **0** | InsertingTrocar | 109 | **18** | PickingOutLymphNode | 200 |
| **1** | CuttingMesocolon | 315 | **19** | BaggingProstate | 30 |
| **2** | PullingVasDeferens | 366 | **20** | BladderNeckDissection | 1483 |
| **3** | CliipingVasDeferens | 30 | **21** | BladderAnastomosis | 3094 |
| **4** | CuttingVasDeferens | 68 | **22** | MHCreamApp | 63 |
| **5** | ClippingTissue | 70 | **23** | InsertingDrain | 545 |
| **6** | Bleeding | 61 | **24** | PullingProstate | 833 |

| | | | | | |
|---|---|---|---|---|---|
| **7** | PullingSeminalVesicle | 2,537 | **25** | CuttingDVC | 270 |
| **8** | ClippingSeminalVesicle | 118 | **26** | StitchingDVC | 439 |
| **9** | CuttingSeminalVesicle | 2,578 | **27** | ClippingBladderNeck | 128 |
| **10** | PickingOutCutTissue | 37 | **28** | CuttingThread | 93 |
| **11** | SuckingBlood | 2,979 | **29** | HMMaterialApp | 154 |
| **12** | SuckingSmoke | 144 | **30** | PullingLymphNode | 5,146 |
| **13** | CuttingLymphNodes | 6,701 | **31** | UrachusDissection | 264 |
| **14** | ClippingLymphNodes | 191 | **32** | CuttingProstate | 1,861 |
| **15** | InsertingCamera | 172 | **33** | PullingBladderNeck | 11 |
| **16** | PullingTissue | 2,182 | **34** | RemovingClip | 114 |
| **17** | CuttingTissue | 1,817 | | **Total no of examples:** | **35,203** |

*Table 5: Number of instances of each of the 35 clinically relevant actions identified in video #1 of the SARAS dataset.*

The video length was about 3hrs, with the portion actually containing relevant data spanning about 2hrs. As mentioned, annotation was performed at a rate of 4 frames per second. The portion annotated contained 24,229 frames, with 23,558 frames containing positive examples and the rest negative examples.

## 4.3 *Experimental results*

Encouraging preliminary results were obtained. The analysed video was split into a training and a testing set of video frames. Eventually, 75% of the video was used for training and 25% for testing. More precisely, the video was split into 38 portions of 800 frames each. Within each region, 600 frames were used for testing and 200 for testing. The following diagram (Figure 19) gives a graphical illustration of the split over time.

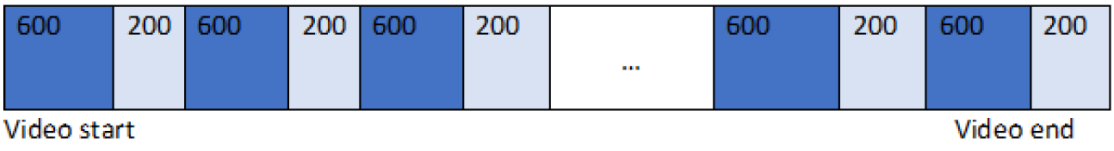


*Figure19: Training/testing split in video #1 of the SARAS surgical action dataset.*

Detections were assumed to be successful when the overlap between predicted and actual bounding box was above a certain threshold. Overlap was measured, as standard in action detection, using the Intersection over Union (IoU) ratio (see Figure 20):

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}.$$

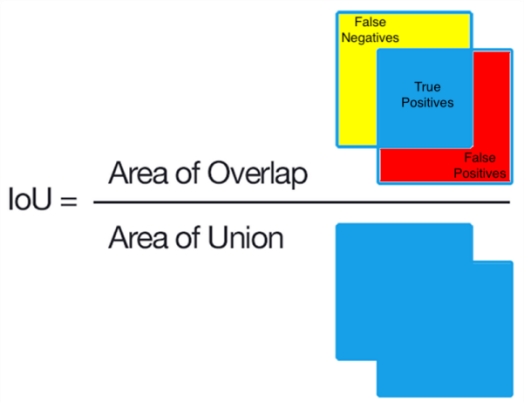


*Figure 20: Intersection over Union (IoU) index.*

### 4.3.1 Quantitative results in terms of Frame mean Average Precision

The selected threshold was 0.50 (i.e., predicted and actual bounding box would have to overlap by at least 50%). Results in terms of Frame m-AP (mean Average Precision, see above for the definition) are illustrated in Table 5. The training set amounted to 17,400 frames in total, while the test data contained 6,158 frames. The total number of boxes was 35,003, as many frames do contain more than one action. The performance reported is that of the online, real time multiple action detection model by Singh et al. [3], trained for 150,000 iterations.

| ID | Class Name | Training examples | Testing examples | Frame mean-AP | ID | Class Name | Training examples | Testing examples | Frame mean-AP |
|---|---|---|---|---|---|---|---|---|---|
| **0** | InsertingTrocar | 90 | 19 | 93.4 | **18** | PickingOutLymphNode | 127 | 73 | 81.8 |
| **1** | CuttingMesocolon | 236 | 79 | 99.8 | **19** | BaggingProstate | 30 | 0 | 0 |
| **2** | PullingVasDeferens | 262 | 104 | 98.8 | **20** | BladderNeckDissection | 1288 | 195 | 99.5 |
| **3** | CliipingVasDeferens | 18 | 12 | 100 | **21** | BladderAnastomosis | 2242 | 852 | 99.4 |
| **4** | CuttingVasDeferens | 47 | 21 | 100 | **22** | MHCreamApp | 63 | 0 | 0 |
| **5** | ClippingTissue | 48 | 22 | 98.9 | **23** | InsertingDrain | 529 | 16 | 100 |
| **6** | Bleeding | 48 | 13 | 42.2 | **24** | PullingProstate | 578 | 255 | 96.3 |
| **7** | PullingSeminalVesicle | 1973 | 564 | 95.0 | **25** | CuttingDVC | 82 | 188 | 99.9 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **8** | ClippingSeminalVesicle | 95 | 23 | 0.02 | **26** | StitchingDVC | 322 | 117 | 100 |
| **9** | CuttingSeminalVesicle | 1743 | 835 | 93.4 | **27** | ClippingBladderNeck | 123 | 5 | 100 |
| **10** | PickingOutCutTissue | 37 | 0 | 0 | **28** | CuttingThread | 35 | 58 | 100 |
| **11** | SuckingBlood | 2197 | 782 | 89.1 | **29** | HMMaterialApp | 154 | 0 | 0.00 |
| **12** | SuckingSmoke | 140 | 4 | 1.19 | **30** | PullingLymphNode | 4150 | 996 | 95.9 |
| **13** | CuttingLymphNodes | 5082 | 1619 | 94.3 | **31** | UrachusDissection | 164 | 100 | 100 |
| **14** | ClippingLymphNodes | 103 | 88 | 100 | **32** | CuttingProstate | 1152 | 509 | 91.3 |
| **15** | InsertingCamera | 160 | 12 | 99.4 | **33** | PullingBladderNeck | 8 | 3 | 100 |
| **16** | PullingTissue | 1544 | 502 | 90.8 | **34** | RemovingClip | 114 | 0 | 0.00 |
| **17** | CuttingTissue | 1333 | 484 | 87.1 | | | | | |

*Table 6. Frame m-AP for each of the individual action classes (IDs), as tested on video #1 of the SARAS dataset.*

The results are graphically illustrated in the histogram in Figure 21.

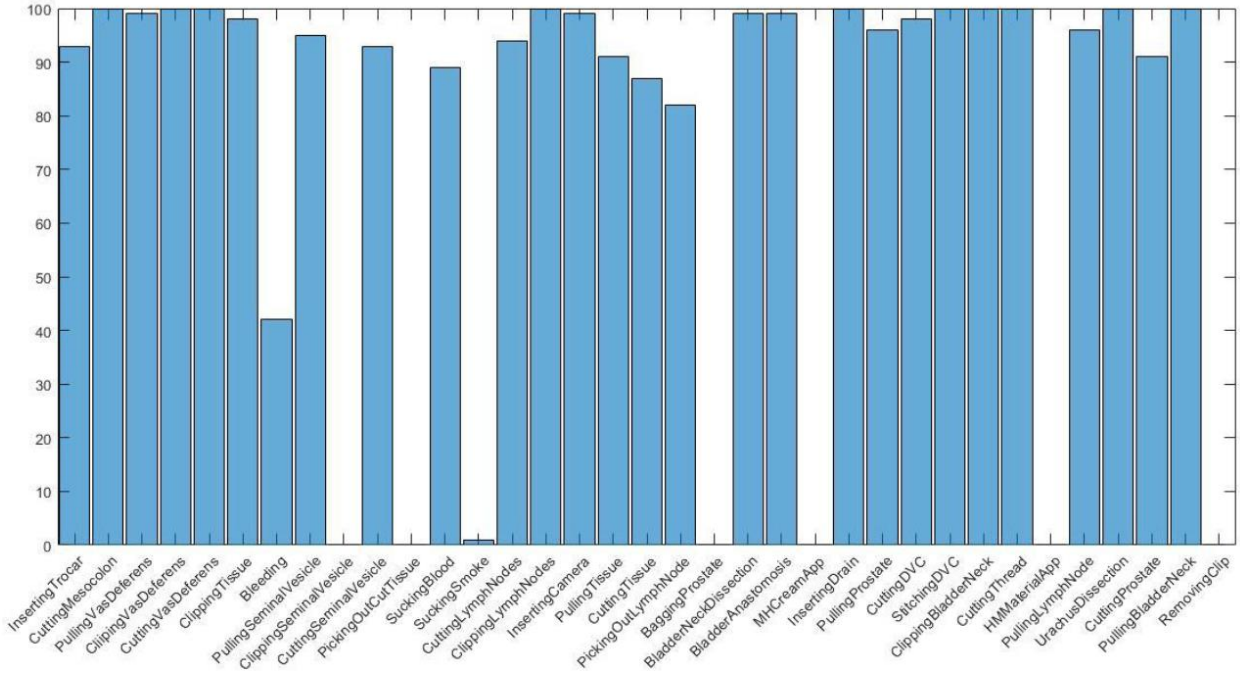


*Figure 21: Frame m-AP for each of the individual action classes, as tested on video #1 of the SARAS dataset.*

As it can be observed in Table 6, some classes do not appear in either the training data or the testing data. This is because the video frames were split into training and testing in time, without ensuring that each class had sufficient representation in both training and testing sets. As expected, the test results for classes with missing data in either training or testing set is 0. To appreciate the average performance of the model it is thus important to look at the overall mean-AP index, calculated while excluding 'anomalous' classes. For an IoU threshold of 0.5 (50% overlap), the overall such value was around 88% (after excluding 'bad' classes), 75.5% when including all classes. As said, this was obtained after training the model for 150,000 iterations.

### 4.3.2 Visual comparison with the ground truth

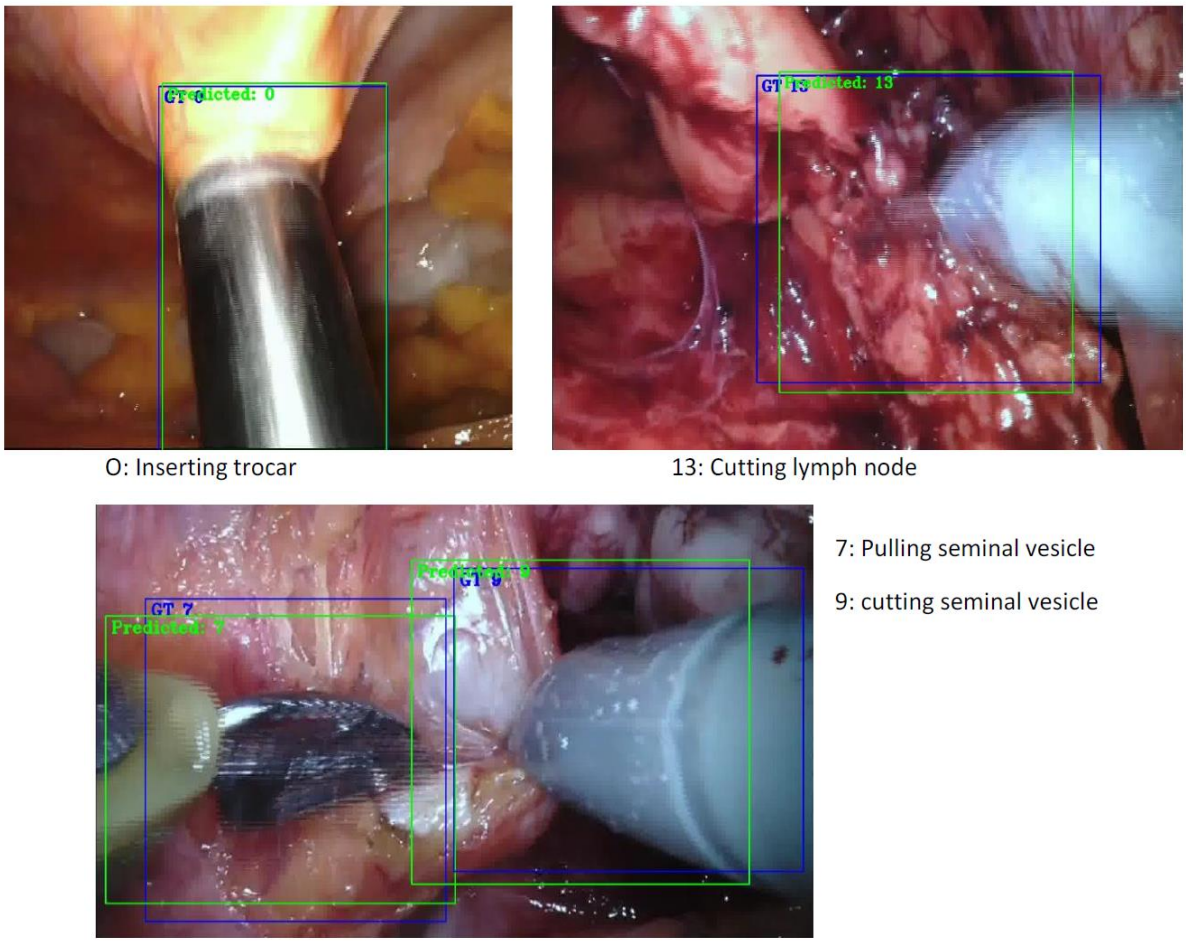


*Figure 22: Example detections for some of the actions considered: 0 – inserting trocar and 13 – cutting lymph node (top); 7 – pulling seminal vesicle and 9 – cutting seminal vesicle (bottom).*

Figure 22 visually illustrates some example detections produced by the above experiment. Various action classes such as 'inserting trocar', 'cutting lymph node' or 'cutting seminal vesicle' are considered. As one can appreciate, the overlap between predicted bounding boxes (in green) and the ground truth ones (in blue) is remarkable.

## 4.4 *Completion of the SARAS surgical action dataset*

In the first half of 2019 we are able, thanks to OSR's assistance, to complete the annotation of all four videos. The annotation was done using Microsoft VoTT, as previously described, at a rate of 1

frame per second. The choice of the relevant clinical actions was made on the basis of the reconstruction of the prostate's phantom conducted by our SARAS partners Dundee, and of the RARP simplified procedure described in Deliverable D1.1. Compared to the initial version of the dataset composed of a single video, the complete SARAS dataset contemplates 21 classes of surgeon actions. More details will be released soon.

As of June 2019 we are then ready to release the complete SARAS surgical action dataset, in order to make it publicly available to all researchers in the field. We believe this will be the first action detection dataset on surgical data. A suitable venue is the MICCAI series of challenges, cfr. for instance [24]. The annual MICCAI conference is the premiere venue in medical imaging. It attracts world leading biomedical scientists, engineers, and clinicians from a wide range of disciplines associated with medical imaging and computer assisted intervention. Both a conference and journal paper will be submitted to appropriate venues to mark the release of the dataset.

## 4.5 *Transfer learning and integration with demonstrator data*

As SARAS' SOLO-SURGERY platform is set up, new videos are being acquired from the demonstrator, capturing laparoscopic interventions conducted on the SARAS phantoms (synthetic anatomies). Obviously, the appearance of phantom organs, albeit as realistic as possible, is quite different from that of real tissue. Nevertheless, models learned on the above real RARP videos do not need to be discarded, but can be exploited as 'pre-training' information, i.e., to set sensible initial values for the weights of the various connections in the neural network.

As the fresh videos from the demonstrator become available, and are annotated, the networks learned from the real RARP videos can be 'fine-tuned' on the new data – in other words, the weights of the connections are adjusted to reflect the new data, starting from the initial values previously learned. This technique, known within machine learning as *transfer learning* [25], has been demonstrated to be extremely effective in all tasks it has been applied to, including medical imaging [26].

# Current work and future developments

Action detection technology does not only concern Task 6.1 (Online surgeon action recognition), but it affects as well Tasks 6.2 (Current procedure stage recognition) and 6.3 (Predicting future surgeon actions). Indeed, Task 6.2 envisages modelling an entire surgical procedure as a graph, whose nodes are formed by individual actions and events, represented as action tubes.

## 5.1 *Recurrent convolutional networks*

As we saw above, current action detection methodologies rely on object detectors that can locate actions of interest in the image plane, which are then encoded using traditional 2D Convolutional Neural Networks.

3D Convolutional Neural Networks (C3D, I3D, (2+1)D) [27,28,29] have recently risen to the forefront of video classification, as they can encode both the spatial appearance and the temporal dynamics of the action or event portrayed by a video in a joint fashion. However, to date 3D CNN architectures only work on entire videos taken as a whole, in a batch manner (i.e., they cannot process videos in real time, frame by frame, as they stream in).

In response, OBU has recently proposed a novel Recurrent Convolutional Network (RCN) designed to be the first *causal* 3D CNN architecture, i.e., able to generate features by analysing all past video frames. RCN's fundamental notion is to replace the temporal convolution component in the recent, efficient (2+1)D architecture [27], with a recurrent model inspired by Recurrent Neural Networks (RNNs) in the temporal dimension (see Figure 23).

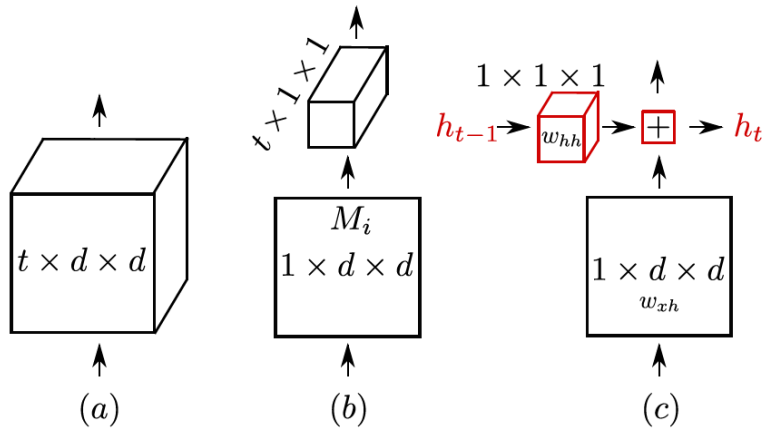


*Figure 23: Illustration of 3D architectures used on sequences of input frames. (a) Standard 3D convolution, as in I3D [30] or C3D [31]. (b) 3D convolution decomposed into a 2D spatial convolution followed by a 1D temporal one, as in S3D [29]. In R(2+1)D [27] the number $M_i$ of middle planes is increased to match the number of parameters in standard 3D convolution. (c) Our proposed decomposition of 3D convolution into 2D spatial convolution and recurrence (in red) in the temporal direction, with a $1 \times 1 \times 1$ convolution $w_{hh}$ as hidden state transformation.*

The unrolled diagram of an RCN network is illustrated in Figure 24. The network is composed of a single *Recurrent Convolutional Unit* (RCU) layer, modelled by the following recurrent equation:

$$h(t) = h_{t-1} * w_{hh} + x_t * w_{xh},$$

followed by a batch normalisation (BN) layer, a ReLU (Rectified Linear Unit) activation layer, and a final convolutional layer used for classification.

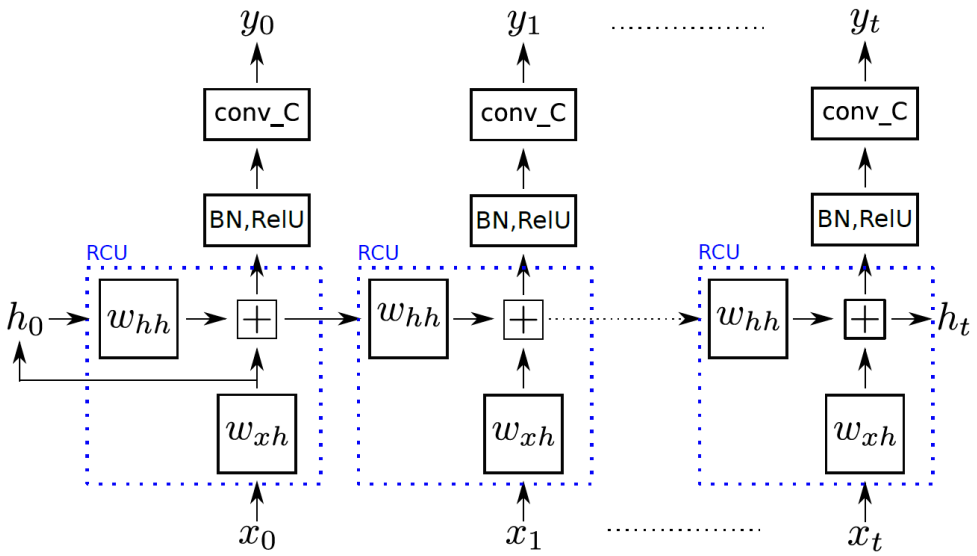


*Figure 24: An unrolled Recurrent Convolutional Network (RCN) composed, at each stage, by a single RCU layer followed by a batch normalisation (BN) layer, a ReLU activation layer, and a final convolutional layer (used for classification).*

As a result, RCN as presented here is not only causal, but poses no constraints on the modelling of temporal dependencies (as opposed to an upper bound of $n$ in the case of temporal convolutions). Temporal dependencies are only limited by the input sequence length at training time. In addition, RCN is designed to directly benefit from model initialisation via ImageNet pre-trained weights, as opposed to state of the art approaches, and in line with clear emerging trends in the field.

### 5.1.1 Datasets

Our experiments show that RCN outperforms baseline I3D and (2+1)D models, while displaying all the above desirable properties. In our initial tests [32] RCN was evaluated on the Kinetics [33] and MultiThumos [34] datasets.

The *Kinetics* dataset comprises 400 classes and 260K videos; each video contains a single atomic action. Kinetics has become a de facto benchmark for recent action recognition works. The average duration of a video clip in Kinetics is 10 seconds. The *MultiThumos* dataset is a multilabel extension of THUMOS. It features 65 classes and 400 videos, with a total duration of 30 hours. On average, it provides 1.5 labels per frame, 10.5 action classes per video. Each video can be up to 30 minutes long, in contrast with the Charades [35] dataset. Videos are densely labeled, as opposed to those in THUMOS [36] or ActivityNet [37]. MultiThumos allows us to show the dense prediction capabilities of RCN on long, real-world videos.

### 5.1.2 Performance of RCN

Table 7 compares the performance of RCN with that of other 3D CNN architectures on the Kinetics dataset.

| Model | Clip length | Initialisation | Accuracy (%) |
|---|---|---|---|
| ResNet34-(2+1)D [27] | 16 | random | 67.8 |

| | | | |
|---|---|---|---|
| ResNet34-I3D [30] | 16 | ImageNet | 68.2 |
| ResNet34-RCN | 16 | ImageNet | **70.3** |
| ResNet50-I3D | 8 | ImageNet | 68.8 |
| ResNet50-RCN | 8 | ImageNet | 71.2 |
| ResNet50-RCN-unrolled | 8 | ImageNet | **72.2** |

*Table 7. Video-level action classification accuracy of different models on the validation set of the Kinetics dataset. Top performers in bold.*

All 3D CNN architectures examined build on a backbone network derived from 2D CNNs. In our tests we considered, in particular, the state of the art ResNet34 and ResNet50 backbone nets.

It is clear from these figures that RCN significantly outperforms state-of-the-art 3D networks – e.g. our network outperforms the equivalent I3D network by more than 2% across the board. The ability to model long-term temporal reasoning of RCN is attested by the performance of the unrolled version (last row of Table 7).

Results on temporal action detection on MultiThumos are shown in Table 8.

| Model | Input | mAP@1% | mAP@8% |
|---|---|---|---|
| Two-stream+LSTM [34] | RGB+FLOW | 28.1 | — |
| MultiLSTM [34] | RGB+FLOW | 29.7 | — |
| Inception-I3D by [38] | RGB+FLOW | — | 30.1 |
| Inception-I3D + SE [38] | RGB | — | 36.2 |
| ResNet50-I3D | RGB | 34.8 | 36.9 |
| ResNet50-RCN | RGB | 35.3 | 37.3 |
| ResNet50-RCN-unrolled | RGB | **36.2** | **38.3** |

*Table 8. Action detection/segmentation results on the MultiThumos dataset. mAP is computed both from dense prediction at every frame (mAP@1) and every 8th frame (mAP@8)..*

ResNet50 was employed as a backbone for both our RCN and the baseline I3D. To capture the longer duration, we used 16-frame clips as input. Two LSTM-based causal models presented by [34] are shown in rows 1 and 2. Piergiovanni et al. [38] use pre-trained I3D to compute features, but do

not train I3D end-to-end, hence their performance is lower than in our version of I3D. Our RCN outperforms all other methods, including non-causal I3D+Super-Events (SE) and the I3D baseline.

## 5.2 *Whole action tube regression*

Our final objective is to be able, at each time instant, to predict whole action instances (tubes) on the fly. Both AMTnet and TraMNet are steps towards this goal. Our recent Recurrent Convolutional Network (RCN) proposal, described above, can then be plugged in to both described the temporal dynamics of an action tube, and to replace 2D feature encoding with state of the art 3D CNN representations.

OBU is currently working on a latent-variable formulation of whole action tubes, in order to provide a truly optimal solution to the action detection problem.

The concept is shown in Figure 25. Rather than solving for

$$T^* = arg \max_{T \subset V} score(T),$$

where $T$ is a subset of the input video associated with the instance of a known action class, current methods seek partial solutions (as bounding boxes) $R \subset I(t)$ for each video frame $I(t)$:

$$R^*(t) = arg \max_{R \subset I(t)} score(R),$$

to later compose all partial frame-level detections into an action tube.

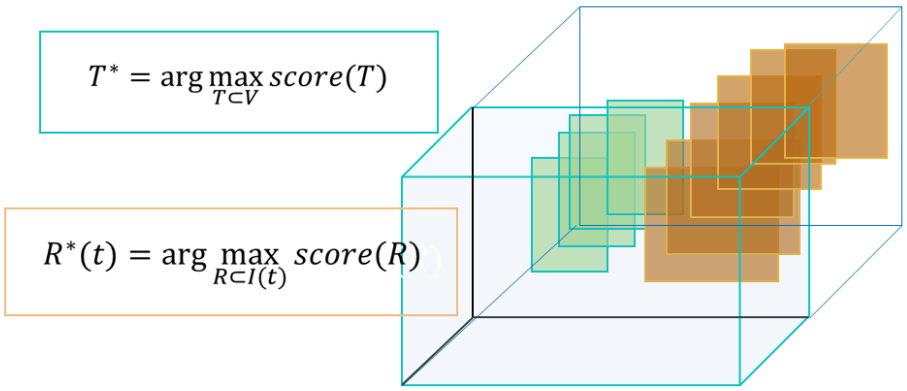


*Figure 25: A truly optimal solution to the (surgeon) action detection problem amounts to detecting whole action tubes T (top, in green) in an online fashion, rather than assembling video frame-level detections R, as in traditional approaches (bottom, in yellow).*

We intend to model whole action tubes by a *latent variable* model, describing the probability of a sequence of detections to occur. We will consider various alternatives: a Long-Short Term Memory (LSTM) network formulation, the RNN-based recurrent model at the core of RCN, and a hidden Markov model formulation, in deep learning form (https://github.com/clinicalml/dmm).

This ongoing work builds on our recent TraMNet design which exploits transition probabilities for frame-level detections.

## 5.3 *Modelling complex activities*

To conclude, recall that a laparoscopic procedure is a complex activity, composed by a number of coordinated events and actions, arranged over a space of time of three or four hours. Whereas simple surgeon actions and events can be effectively modelled as action tubes and detected as explained in this Deliverable, complex activities like a surgical procedure involve multiple, coordinated actions taking place at different times, in different parts of the scene.

Complex activities, we argue, can be effectively modelled as graphs of 'atomic' actions or events, as illustrated in Figure 26.

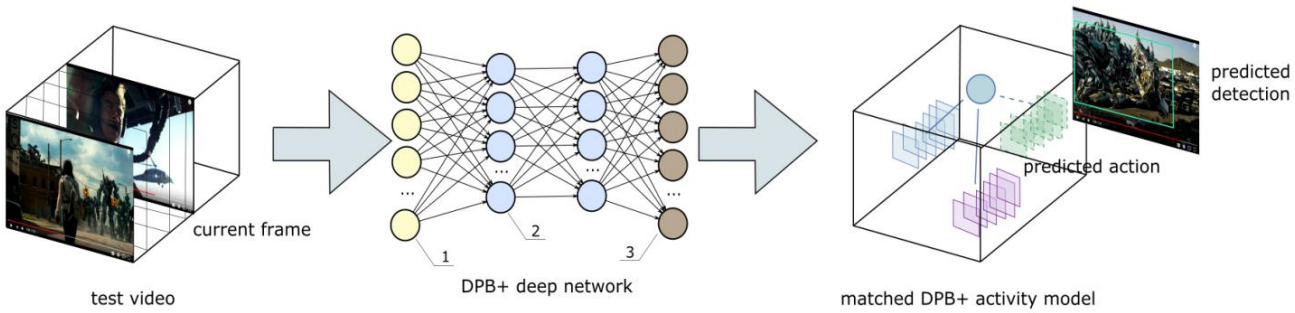


*Figure 26: Complex activities can be modelled, via appropriate deep neural networks, as graphs arranging multiple actions and events into a graph, spanning space and time. Once a complex activity is partly detected, the structure of the graph can be exploited to predict what action/event is going to take place next, when and where (in the image plane or in 3D).*

OBU is currently working on a deep network architecture designed to detect those graphs of actions, so that when one activity graph is partially observed (e.g., after we observe the initial stages of a RARP procedure) the network is in a position to guess what events may take place in the future and where. The block diagram of the architecture is depicted in Figure 27.

More technically, an 'activity' (e.g., a laparoscopic procedure) can be represented by a graph whose nodes and edges represent individual events (e.g., the surgeon cutting a line of tissue) and their relative spatial and temporal location (first the surgeon pulls up an anatomical structure, then he/she cuts), respectively. These graphs are called deformable part-based models (DPMs) [39].

The link between part-based models and deep neural networks remains relatively unexplored [40]. Evidence, though, suggests that DPMs are just a special form of convolutional neural networks [41]. Indeed, deformable part-based models can be implemented as CNNs by simply unrolling the inference steps of their training algorithm. The real time detection of complex video activities composed by co-occurring events, however, requires us to overcome classical DPM limitations by: (a) allowing flexibility in the number $P$ of events present, and (b) modelling the spatiotemporal shape of activity parts more flexibly than by cuboidal video sub-volumes, as in the 'action tube' formulation. Once networks regressing action tubes are made to be end-to-end trainable (i.e., their optimal weight values can be learned by optimising an appropriate cost function), the resulting architecture can also be.

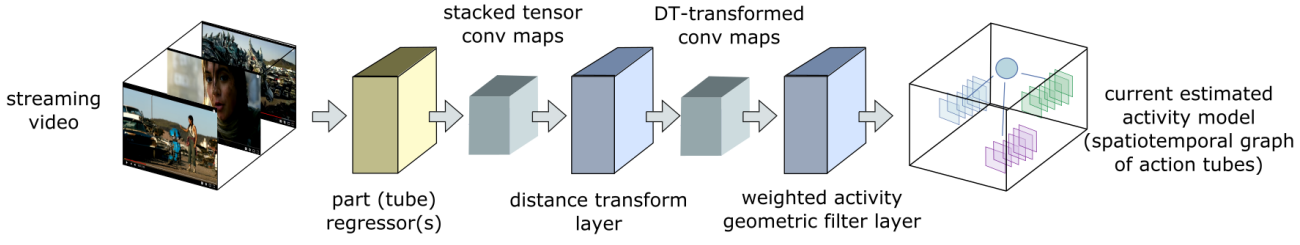


*Figure 27: Proposed deep learning architecture for complex visual activity modelling and detection.*

As in classical DPM inference [42], the (tensorial) feature maps produced by tube regressors for each part $p = 1, \dots, P$ (see Figure 27) are fed to a distance-transform (DT) and a geometric filter layer, checking that the detected parts are close to the relative spatiotemporal locations learned at training time. A weight for each activity part allows us to correctly detect a relevant activity class no matter how many 'atomic' events are involved. Sparsity (to minimise the number of necessary parts) can be enforced, e.g., by appropriately regularising the network's objective function [43]. Finally, detection is performed by updating a matching score frame-by-frame.

The results of this analysis will be compiled under Task 6.2 and 6.3, and will appear in Deliverable 6.2.

# References


[1] S. Saha, G. Singh, M. Sapienza, P. H. S. Torr, and F. Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference (BMVC)*, 2016.

[2] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521.7553, 436-444

[3] G. Singh, S. Saha, M. Sapienza, P. Torr e F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, pp. 25-36, 2004.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. arXiv preprint arXiv:1512.02325, 2015

[6] S. Saha, G. Singh and F. Cuzzolin. AMTnet: Action-Micro-Tube regression by end-to-end trainable deep architecture. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.

[7] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems* 27, pages 568–576, 2014.

[8] Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human action classes from videos in the wild. Technical report, CRCV-TR-12-01 (2012).

[9] Idrees, H., Zamir, A.R., Jiang, Y.G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The thumos challenge on action recognition for videos in the wild. *Computer Vision and Image Understanding* 155 (2017) 1–23.

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*, pages 2556–2563, 2011.

[11] P. Weinzaepfel, X. Martin, and C. Schmid. Human action localization with sparse spatial supervision. arXiv preprint arXiv:1605.05197, 2016.

[12] H. S. Behl, M. Sapienza, G. Singh, S. Saha, F. Cuzzolin and P. H. S. Torr. Incremental Tube Construction for Human Action Detection. In *British Machine Vision Conference (BMVC)*, 2018.

[13] Christian Wolf et al. The LIRIS Human activities dataset and the ICPR 2012 human activities recognition and localization competition. Technical Report RR-LIRIS-2012-004, March 2012.

[14] G. Singh, S. Saha and F. Cuzzolin. TraMNet - Transition Matrix Network for Efficient Action Tube Proposals. In *Asian Computer Vision Conference (ACCV)*, Perth, Australia, Dec 2018

[15] Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C. Action tubelet detector for spatiotemporal action localization. In *International Conference on Computer Vision (ICCV)*, 2017.

[16] Hou, R., Chen, C., Shah, M. Tube convolutional neural network (T-CNN) for action detection in videos. In *International Conference on Computer Vision (ICCV)*, 2017.

[17] Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D.A., Toderici, G., Li, Y., Ricco, S., Sukthankar, R., Schmid, C., et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. arXiv preprint arXiv:1705.08421 (2017).

[18] S. Saha, G. Singh and F. Cuzzolin. Two-Stream AMTnet for Action Detection. Submitted to the *IEEE Transations on Pattern Analysis and Machine Intelligence*, 2019.

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.

[21] X. Peng and C. Schmid. Multi-region two-stream R-CNN for action detection. In *European Conference on Computer Vision*, pp. 744–759, 2016.

[22] https://github.com/microsoft/VoTT

[23] Z. Chen, Z. Zhao and X. Cheng. Surgical instruments tracking based on deep learning with lines detection and spatio-temporal context. In *Chinese Automation Congress (CAC)*, pp. 2711 - 2714, 2017.

[24] https://www.miccai2019.org/calls/call-for-challenges/

[25] Pan, S.J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pp. 1345-1359, 2009.

[26] Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5), pp. 1285-1298, 2016.

[27] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pages 6450–6459, 2018.

[28] X.Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[30] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[32] G. Singh and F. Cuzzolin. Causal spatiotemporal representations. Submitted to the *International Conference on Computer Vision (ICCV)*, 2019.

[33] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[34] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. arXiv preprint arXiv:1507.05738, 2015.

[35] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, 2016.

[36] Y. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14, 2014.

[37] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[38] A. Piergiovanni and M. S. Ryoo. Learning latent superevents to detect multiple activities in videos. In

*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5304–5313, 2018

[39] Y. Tian, R. Sukthankar and M. Shah. Spatiotemporal Deformable Part Models for Action Detection. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 2642-2649, 2013.

[40] N. Zhang et al. Part-based R-CNNs for Fine-grained Category Detection. In *European Conference on Computer Vision (ECCV)*, 2014.

[41] R. Girshick et al. Deformable Part Models are Convolutional Neural Networks. arXiv:1409.5403v2, 2014.

[42] P. Felzenszwalb et al. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (2010).

[43] W. Wen et al. Learning Structured Sparsity in Deep Neural Networks. In *Advances in Neural Information Processing Systems*, 2016.

# Appendix

## *GitHub code repositories*

Code associated with the paper: “Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos”, by Saha et al.

https://bitbucket.org/sahasuman/bmvc2016_code

Code associated with the paper: “Online Real-time Multiple Spatiotemporal Action Localisation and Prediction”, by Singh et al.

https://github.com/gurkirt/realtime-action-detection

Code associated with the paper: “Incremental Tube Construction for Human Action Detection”, by Behl et al.

https://github.com/harkiratbehl/OJLA

## *Videos and media*

YouTube video related to the paper: “Deep Learning for Detecting Multiple Space-Time Action Tubes in Videos”, by Saha et al.

https://www.youtube.com/watch?v=vBZsTgjhWaQ

YouTube video related to the paper: “Online Real-time Multiple Spatiotemporal Action Localisation and Prediction”, by Singh et al.

https://www.youtube.com/watch?v=e6r_39ETe-g

YouTube video related to the paper: “Incremental Tube Construction for Human Action Detection”, by Behl et al.

https://www.youtube.com/watch?v=vGtokmcozYo

# SARAS PROJECT – ETHICS AND DATA COMPLIANCE DOCUMENT

The advent of robotics has profoundly impacted biomedical practice, especially surgery. The ambition of the "Smart Autonomous Robotic Assistant Surgeon" project (SARAS, www.saras-project.eu) is to develop a highly efficient and next generation surgery system – that the SARAS consortium defined "solo-surgery system" – able to support surgeons during Robotic Minimally Invasive Surgeries (R-MIS). In particular, the solo-surgery system consists of a pair of cooperating and autonomous robotic arms holding the surgical instruments. Its purpose is to help the first surgeon to perform R-MIS, without the need of an expert assistant surgeon. This will hopefully increase the social and economic efficiency of a hospital while guaranteeing the same level of safety for patients.

In addition to its primary purpose reported upon, the SARAS project also aims to promote the Open Access Policy for the Horizon 2020 programme, according to the "Guidelines to the Rules on Open Access to Scientific Publications and Open Access to Research Data in Horizon 2020" (see: http://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-pilot-guide_en.pdf). Open access refers to the practice of providing online access to scientific information that is free of charge to the end-user and reusable. The expression 'Scientific information' refers to peer-reviewed scientific research articles and/or research data (data underlying publications, curated data and/or raw data). The term 'access' includes not only the right to read, download and print the paper, but also the right to copy, distribute, search, link, crawl and mine. The practice of Open Access is nonetheless subjected to the terms and conditions set out in the Grant Agreement, and has to be compliant with the aforementioned Guidelines. This means that data that will be then made open within the scientific community should have been collected and stored in compliance with regulatory and ethics norms.

As for data compliance, the SARAS project has received funding from the European Union's Horizon 2020, which is regulated by European Union and State laws. Therefore, any partners, contractors or service providers that use or reuse personal data on behalf of the consortium must comply with EU Regulation 2016/679 (also known as EU General Data Protection Regulation, from here after: GDPR) which came into force in May 2018. The GDPR is relevant to all organizations inside the European Union (EU), the European Economic Area (EEA) and to organizations from other countries, if they process data of European citizens. The GDPR regulates the collection, storage, and processing of personal data, which are defined in Article 4 as "any information relating to an identified or identifiable natural person ('data subject') "*any data that can be linked to a specific natural person*". Data that do not include such identifiers are commonly regarded as anonymous and are outside the scope of GDPR (Recital 26).

To lawfully process personal data for scientific research purposes (within a H2020 project, which is regulated by European Union and State laws) several legal elements must be considered in advance. In particular, the data controller, which in this phase of the SARAS project is San Raffaele Hospital, must:

a) respect the principles set by Article 5 (general principles) and 25 (data protection by design and by default) of the GDPR;

b) guarantee the presence of legal grounds for data processing, according to Articles 6-9-10-89 of the GDPR;
c) inform data subjects (or its exceptional exclusion) according to Articles 13-14 of the GDPR.

For the processing of special categories of data (like genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health, also known as sensitive data), the legal grounds identified under Article 6 shall be applied only if Article 9 GDPR provides for a specific derogation from the general prohibition to process special categories of data.
Patient consent provides both a lawful basis and a permitted condition for processing sensitive data (with the only difference that for sensitive data consent must be also "explicit").

Sensitive data used by San Raffaele Hospital (images and footage from the surgical endoscope of radical prostatectomies) in the SARAS project, have been lawfully collected through explicit consent of the data subject (legal bases: Article 6(1)(a), Article 9(2)(a)) for a previous observational study, that took place in 2018 at the Urological Research Institute of San Raffaele Hospital. This study has been approved by the Research Ethics Committee of San Raffaele Hospital, composed of more than forty members, currently evaluating more than 30 protocols/month, for a total of roughly 400 per year.
In line with Articles 13 and 14 of the GDPR – that impose to adequately inform data subjects about their personal data processing – before their prostatectomy, patients involved in the project received a privacy notice where they were asked to give consent to the investigator for the use of images and footages for further educational purposes. According to Article 5(1)(e), which allows the controller to store data for a longer period for scientific research purposes, Hospital San Raffaele kept the sensitive data collected during the prostatectomy adequately protected for future research purposes. According to the privacy notice given to patients at data collection time, prior to using sensitive data for educational purposes data has to be anonymized by San Raffaele Hospital, and so we did.
In line with recital 21 and 26 of the GDPR, European data protection law applies to "personal data," which is defined, in part, as "any information relating to an identified or identifiable natural person." Therefore, data that has been anonymized is no longer "personal data" and - as a consequence - is not subject to the requirements of data protection law.
In addition to GDPR, as the data used within the SARAS project have been collected in the context of a previously approved observational study (see upon), SARAS is also compliant with the following international ethics and regulatory standards regulating human subject research:

- Convention for the protection of Human Rights and dignity of the human being with regard to the application of biology and medicine: Convention on Human Rights and Biomedicine, Oviedo 1997.
- The Belmont Report. Ethical Principles and guidelines for the protection of Human Subjects of Biomedical and Behavioral Research. DHEW Publication N. 78-0012, Washington, 1978.
- Declaration of Helsinki, ethical principles for medical research involving human subjects, revised October 2013.
- Council for International Organizations of Medical Sciences (CIOMS) in collaboration with the World Health Organization (WHO), International Ethical Guidelines for Biomedical Research Involving Human Subjects, revised in 2016.

- Commission Directive 2005/28/EC laying down principles and detailed guidelines for good clinical practice as regards investigational medicinal products for human use, as well as the requirements for authorization of the manufacturing or importation of such products.
- Regulation (EU) 536/2014 of the European Parliament and of the Council of 16 April 2014 on clinical trials on medicinal products for human use, and repealing Directive 2001/20/EC

According to this analysis, we are able to maintain that San Raffaele Hospital lawfully collected and stored sensitive data for further educational purposes. Furthermore, in the course of the SARAS study, before further processing for educational purposes (a definition which includes scientific research purposes) the data previously collected during the aforementioned observational study, San Raffaele Hospital anonymised the data, meaning that only "data rendered anonymous in such a manner that the data subject is not or no longer identifiable" have been used. According to Recital 26, anonymized data falls outside the scope of personal data protection regulation, which does not apply to the processing of "anonymous information, including for statistical or research purpose". As a conclusion the data controller, in this case San Raffaele Hospital, is thus not obliged to comply with the rules on further processing of personal data imposed by the GDPR and Italian State Law, since no personal data are involved.