A shape-constraint adversarial framework with instance-normalized spatio-temporal features for inter-fetal membrane segmentation

Alessandro Casella[a,b,*], Sara Moccia[c,d], Dario Paladini[f], Emanuele Frontoni[e], Elena De Momi[b], Leonardo S. Mattos[a]

[a] *Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy*
[b] *Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy*
[c] *The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy*
[d] *Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, Pisa, Italy*
[e] *Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy*
[f] *Department of Fetal and Perinatal Medicine, Istituto "Giannina Gaslini", Genoa, Italy*

Abstract

Background and Objectives During Twin-to-Twin Transfusion Syndrome (TTTS), abnormal vascular anastomoses in the monochorionic placenta can produce uneven blood flow between the fetuses. In the current practice, this syndrome is surgically treated by closing the abnormal connections using laser ablation. Surgeons commonly use the inter-fetal membrane as a reference. Limited field of view, low fetoscopic image quality and high inter-subject variability make the membrane identification a challenging task. However, currently available tools are not optimal for automatic membrane segmentation in fetoscopic videos, due to membrane texture homogeneity and high illumination variability. **Methods** To tackle these challenges, we present a new deep-learning framework for inter-fetal membrane segmentation on in-vivo fetoscopic videos. The framework enhances existing architectures by (i) encoding a novel (instance-normalized) dense block, invariant to illumination changes, that extracts spatio-temporal features to enforce pixel connectivity in time, and (ii) relying on an adversarial training, which constrains macro appearance. **Results** We performed

*Corresponding author
Email address: alessandro.casella@iit.it (Alessandro Casella)

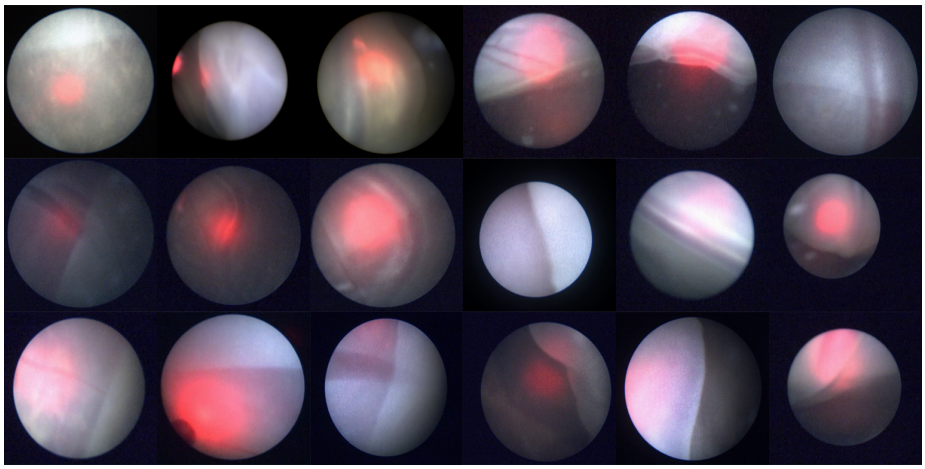


Figure 1: Sample frames from our dataset. The frames are extracted from intra-operative videos acquired in the actual surgical practice for Twin-to-Twin Transfusion Syndrome (TTTS). Each frame refers to a different video. Although video acquisition was performed with the same equipment, the frames present high variability, in terms of: (i) different membrane position, shape, tissue area in the field of view, contrast and texture, (ii) noise and blur, (iii) presence of amniotic fluid particles, (iv) vessels along the membrane equator, (v) different levels of illumination, (vi) presence of laser-guide light.

a comprehensive validation using 20 different videos (2000 frames) from 20 different surgeries, achieving a mean Dice Similarity Coefficient of 0.8780 ± 0.1383. **Conclusions** The proposed framework has great potential to positively impact the actual surgical practice for TTTS treatment, allowing the implementation of surgical guidance systems that can enhance context awareness and potentially lower the duration of the surgeries.

Keywords: Inter-Fetal Membrane, Twin-to-Twin Transfusion Syndrome (TTTS), Deep Learning, Fetoscopy

1. Introduction

Twin-to-twin transfusion syndrome (TTTS) may occur, during identical twin pregnancies, when abnormal vascular anastomoses in the monochorionic placenta result in uneven blood flow between the fetuses. If not treated, the risk of

perinatal mortality of one or both fetuses can exceed the 90% (Baschat et al., 2011). To recover the blood flow balance, the most effective treatment is minimally invasive laser surgery in fetoscopy (Quintero, 2003; Roberts et al., 2014). At the beginning of the surgical treatment, the surgeon identifies the interfetal membrane, which is used as a reference to explore the placenta vascular network and identify vessels to be treated. Limited field of view (FoV), poor visibility, fetuses' movements, high illumination variability (as shown in Fig. 1) and limited maneuverability of the fetoscope makes the membrane identification a challenging task. This results in increased surgery duration, as well as increased risks of complications from the patients' side, such as premature rupture of the membranes Beck et al. (2012), and mental workload, from the surgeons' side.

The Surgical Data Science (SDS) (Maier-Hein et al., 2017) community is working towards developing computer-assisted algorithms to perform intra-operative tissue segmentation (Moccia et al., 2020). However, SDS approaches for membrane segmentation have only been marginally explored.

Work relevant to TTTS video analysis focuses on surgical planning, surgical-phase detection, intrauterine cavity segmentation, placental vessel segmentation and mosaicking reconstruction. Examples of surgical-phase detection in TTTS include the work of Vasconcelos et al. (2018), where a ResNet encoder is used to detect the ablation phase, and Bano et al. (2020), which extends Vasconcelos et al. (2018) by adding an LSTM layer to integrate temporal information and detect different surgical phases. In Torrents-Barrena et al. (2020), a reinforcement learning approach that relies on capsule networks has been proposed to perform automatic intrauterine cavity segmentation from multi-planar placenta magnetic-resonance imaging recordings, for surgical planning purposes. As for placental vessel segmentation, the work in Almoussa et al. (2011) proposes a neural network trained on manually handcrafted features from E4-vivo placenta images. In Sadda et al. (2019), a UNet architecture is proposed to perform patch-based vessel segmentation from intra-operative fetoscopic frames. Large efforts have also been put in mosaicking strategies to provide the surgeons with navigation maps of the placenta. In Daga et al. (2016), SIFT is used as fea-

ture extractor for frame registration, while in Gaisser et al. (2018); Peter et al. (2018); Bano et al. (2019); Tella-Amo et al. (2019) deep-learning strategies are presented.

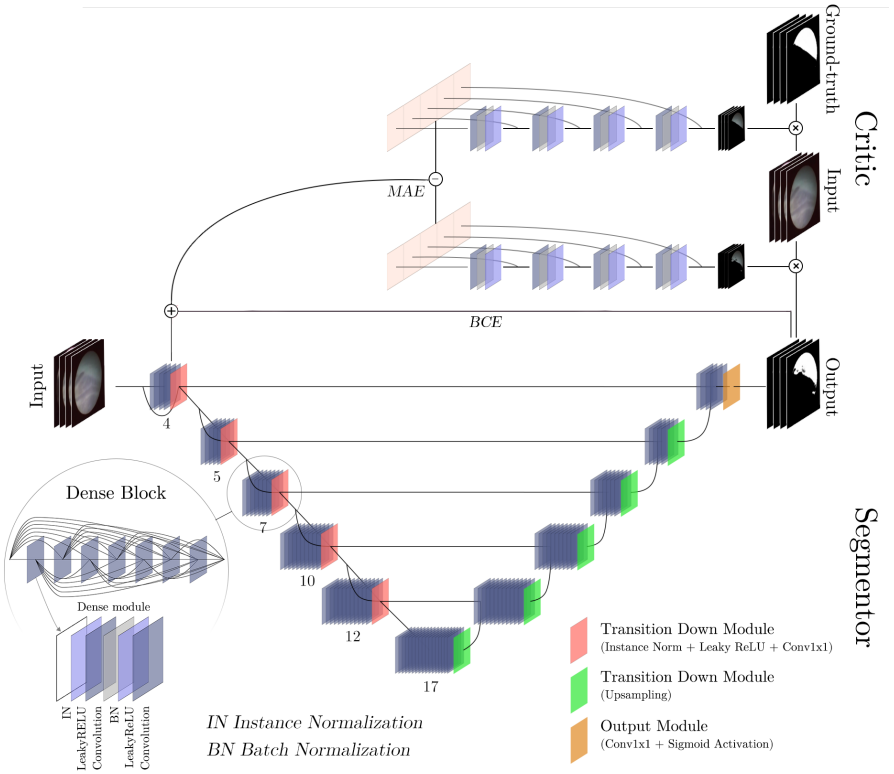


Figure 2: Proposed framework to inter-fetal membrane segmentation in fetoscopic videos. The *segmentor* is a U-shaped network with long-skip connections, consisting of dense blocks, each of which is composed by multiple (number below each block) dense modules. Each module is composed of two pre-activated 3D convolutions, where the normalization is performed at instance (1st convolution) and batch (2nd convolution) level. The transition down and transition up modules perform downsampling and upsamplig, respectively. The *critic*, inspired by Casella et al. (2020), consists of a 3D version of the encoder branch of UNet. During training, as explained in Sec. 2.3, the *critic* extracts the feature vectors from the input masked by the *segmentor* output and the gold standard. The Mean Absolute Error (MAE) computed between the two vectors, contribute, along with the per-pixel binary cross entropy (BCE), to the loss that is minimized during training.

Previous work (Casella et al., 2020), implemented a residual network along with an adversarial training strategy to enforce placenta-shape constraining. Despite achieving promising results, the work does not address the problem of high illumination variability in fetoscopic frames. Furthermore, the temporal information naturally encoded in the fetoscopic videos is not processed. 3D architectures such as V-Net Milletari et al. (2016) have been widely used for volumetric segmentation in medical images. More recently, 3D architectures have been used for processing endoscopic videos. Hence, temporal feature proccessing showed to be effective in segmentation tasks in close fields (e.g., instrument joint detection (Colleoni et al., 2019) and pose estimation (Moccia et al., 2019) to enhance the temporal continuity in feature processing.

Following such considerations, in this work we implement an adversarial strategy to train a novel densely connected 3D fully convolutional neural network (FCNN), which we call the *segmentor*, for inter-fetal membrane segmentation. The third dimension refers to the time for spatio-temporal feature extraction. The dense topology of the *segmentor* is here built with an adaptive mechanism for instance illumination normalization. With a comprehensive study with 20 videos (2000 frames) acquired from 20 women during actual surgery, we investigated the following research hypotheses:

- Hypothesis 1 (H1): The instance-normalized topology can tackle the illumination variability typical of fetoscopic videos acquired during TTTS surgery.
- Hypothesis 2 (H2): The spatio-temporal features can boost segmentation performance enforcing the consistency of segmentation masks across sequential frames.

Here, the gold standard annotation was obtained manually under the supervision of an expert surgeons.

1.1. Contribution of the work

In this work, we address the problem of automatic inter-fetal membrane segmentation to enhance surgeon context awareness during TTTS surgery. Specifically, we extend the adversarial framework presented in Casella et al. (2020) to process, via spatio-temporal convolution, surgical video clips. This allows us to exploit the temporal information naturally encoded in videos. We further design a dense block that encodes instance normalization, to account for illumination changes in the video clips. Recent work explored the potential of adversarial training with spatio-temporal features in other domains than fetal surgery. In Xu et al. (2018), an adversarial architecture was proposed to segment and quantify myocardial infarctions by combining 3D convolutions and Long Short-Term Memory (LSTM) architecture to process spatio-temporal features in cine magnetic-resonance images. To the best of our knowledge, this work is the first to investigate the joint potential of adversarial training, spatio-temporal features and instance normalization in a densely connected segmentation architecture for inter-membrane segmentation in fetoscopic images. We perform extensive experiments on the frame-sampling strategy to build the temporal clips, as well as ablation studies to identify the best configuration of our framework. We will make the dataset collected for this work publicly available, to foster further research in the field.

2. Methods

The proposed framework consists of the *segmentor*, described in Sec. 2.1, and a discriminator network (*critic*), described in Sec. 2.2. The overall framework is shown in Fig. 2. The *segmentor* and *critic* are trained in an adversarial fashion, following the strategy proposed in Casella et al. (2020) and described in Sec. 2.3.

2.1. Segmentor

The *segmentor* has a dense UNet-like architecture consisting of downsampling and upsampling path, linked via long-skip connections. It consists of 11

Table 1: Architecture details for the (top) *segmentor* and (bottom) *critic*. The *IN Conv3D* and *BN Conv3D* refer to *Instance Normalization - leaky ReLu - 3D Convolution* and *Batch Normalization - leaky ReLu - 3D Convolution*, respectively.

Layers	Output Size	Segmentor
Dense Block (1)	$W \times H \times w_{length} \times 192$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 4$
Transition Module (1)	$\frac{W}{2} \times \frac{H}{2} \times w_{length} \times 195$	1×1 conv 2×2 average pool, stride 2, 2, 1
Dense Block (2)	$\frac{W}{2} \times \frac{H}{2} \times w_{length} \times 240$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 5$
Transition Module (2)	$\frac{W}{4} \times \frac{H}{4} \times w_{length} \times 435$	1×1 conv 2×2 average pool, stride 2, 2, 1
Dense Block (3)	$\frac{W}{4} \times \frac{H}{4} \times w_{length} \times 336$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 7$
Transition Module (3)	$\frac{W}{8} \times \frac{H}{8} \times w_{length} \times 771$	1×1 conv 2×2 average pool, stride 2, 2, 1
Dense Block (4)	$\frac{W}{8} \times \frac{H}{8} \times w_{length} \times 480$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 10$
Transition Module (4)	$\frac{W}{16} \times \frac{H}{16} \times w_{length} \times 1251$	1×1 conv 2×2 average pool, stride 2, 2, 1
Dense Block (5)	$\frac{W}{16} \times \frac{H}{16} \times w_{length} \times 576$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 12$
Transition Module (5)	$\frac{W}{32} \times \frac{H}{32} \times w_{length} \times 1827$	1×1 conv 2×2 average pool, stride 2, 2, 1
Dense Block (6)	$\frac{W}{32} \times \frac{H}{32} \times w_{length} \times 816$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 17$
Transition Up Module (1)	$\frac{W}{16} \times \frac{H}{16} \times w_{length} \times 2643$	3×3 Conv3D transpose
Dense Block (7)	$\frac{W}{16} \times \frac{H}{16} \times w_{length} \times 816$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 12$
Transition Up Module (2)	$\frac{W}{8} \times \frac{H}{8} \times w_{length} \times 2067$	3×3 Conv3D transpose
Dense Block (8)	$\frac{W}{8} \times \frac{H}{8} \times w_{length} \times 576$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 10$
Transition Up Module (3)	$\frac{W}{4} \times \frac{H}{4} \times w_{length} \times 1347$	3×3 Conv3D transpose
Dense Block (9)	$\frac{W}{4} \times \frac{H}{4} \times w_{length} \times 480$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 7$
Transition Up Module (4)	$\frac{W}{2} \times \frac{H}{2} \times w_{length} \times 915$	3×3 Conv3D transpose
Dense Block (10)	$\frac{W}{2} \times \frac{H}{2} \times w_{length} \times 336$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 5$
Transition Up Module (5)	$W \times H \times w_{length} \times 531$	3×3 Conv3D transpose
Dense Block (11)	$W \times H \times w_{length} \times 240$	$\begin{bmatrix} 1 \times 1 \text{ IN Conv3D} \\ 3 \times 3 \text{ BN Conv3D} \end{bmatrix} \times 4$
Output Module	$W \times H \times w_{length} \times 1$	1×1 Conv3D, sigmoid

Layers	Output Size	Critic	
		Segmentor output branch	*Gold standard* branch
Feature Extraction (1)	$\frac{W}{2} \times \frac{H}{2} \times w_{length} \times 64$	1×1 IN Conv3D	1×1 IN Conv3D
		2×2 average pool, stride 2, 2, 1	2×2 average pool, stride 2, 2, 1
Feature Extraction (2)	$\frac{W}{4} \times \frac{H}{4} \times w_{length} \times 64$	3×3 IN Conv3D	3×3 IN Conv3D
Feature Extraction (3)	$\frac{W}{8} \times \frac{H}{8} \times w_{length} \times 64$	3×3 IN Conv3D	3×3 IN Conv3D
Feature Extraction (4)	$\frac{W}{16} \times \frac{H}{16} \times w_{length} \times 64$	3×3 IN Conv3D	3×3 IN Conv3D
Feature Extraction (5)	$\frac{W}{32} \times \frac{H}{32} \times w_{length} \times 64$	3×3 IN Conv3D	3×3 IN Conv3D
Output Layer		Concatenate	

dense blocks, 10 transition (up and down) modules and an output module, as reported in Table 1. Inspired by DenseNet (Huang et al., 2017; Jegou et al., 2017), we use dense blocks to foster feature connectivity. Each block is made of multiple dense modules.

To take the temporal information into account, we use 3D convolution to build the modules. Hence, the input of our *segmentor* is a temporal clip (i.e., set of w_{length} temporally consecutive video frames) obtained with a sliding window algorithm (Fig. 3). The sliding window algorithm is inspired by Hou et al. (2017): starting from the first video frame, the first w_{length} frames contribute to the temporal clip (x) of dimensions $W \times H \times N_{channels} \times w_{length}$, where W and H are the frame width and height, respectively, and $N_{channels}$ is the number of image channels. The window then slides of Δ_f frames along the temporal direction, skipping Δ_w frames. This process is repeated until there are available frames, and results in a collection of temporal clips.

The downsampling path of our *segmentor* is designed with an increasingly higher number of modules (from 4 to 17) in each dense block. Each dense block is followed by a *transition down* module for downscaling. Such module is composed of a 1x1x1 convolution and average pooling layer with stride $2, 2, 1$. This stride allows to compensate for the growth in the number of feature channels that occurs in each dense block.

The upsampling path, symmetric to the downsampling one, is designed with an increasingly lower number of modules (from 17 to 4) in each dense block. The *transition up* module performs upscaling, at the end of each dense block, to recover the spatial resolution lost with the stride. Following the standard

UNet (Ronneberger et al., 2015) implementation, the upsampled feature maps are concatenated to those of the downsampling path via long-skip connections. The output module, at the end of the *segmentor*, consists of a 1x1 convolution layer activated with the sigmoid function. The *segmentor* produces as output (y) consisting of w_{length} segmentation masks, hence preserving the temporal size of the input clip.

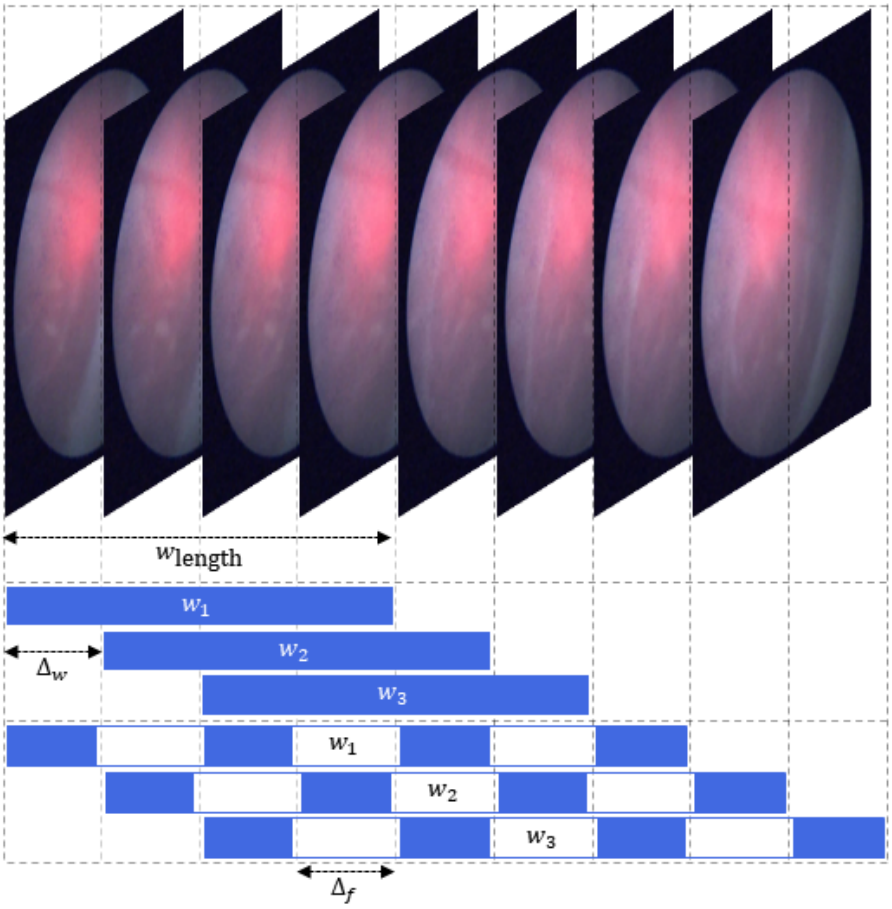


Figure 3: Sliding window algorithm used for building temporal clips (w_1, w_2, w_3). The window, consisting of w_{length} frames, slides of Δ_w frames. For $w_{length} = 4$ and $\Delta_w = 1$, the generated clips overlap of 3 frames. While building a temporal clip, the sliding window can possibly skip Δ_f frames. For $\Delta_f = 0$, clips consist of consecutive frames.

Implementing a dense framework can potentially tackle the processing of images with uniform illumination (Zhou et al., 2020), while poorly illuminated images may still represent a challenge. A possible solution to this problem may be to use instance normalization, which allows attenuating the dependency from instance-specific contrast information (Ulyanov et al., 2016). By building upon the dense module proposed in (Huang et al., 2017), we propose a new dense module that uses two (leaky ReLu) pre-activated convolutions, instead of a single one. In our framework, the first convolution is normalized at instance level, to account for fetoscopic video illumination variability. The process of instance normalization is performed as follows:

$$\hat{z}_{w,h,i,t,n} = \frac{z_{w,h,i,t,n} - \mu_{i,t,n}}{\sqrt{\sigma^2_{i,t,n} + \epsilon}} \tag{1}$$

$$\mu_{i,t,n} = \frac{1}{WH} \sum_{w=1}^{W} \sum_{h=1}^{H} z_{w,h,i,t,n} \tag{2}$$

$$\sigma^2_{i,t,n} = \frac{1}{WH} \sum_{w=1}^{W} \sum_{h=1}^{H} (z_{w,h,i,t,n} - \mu_{i,t,n}) \tag{3}$$

where z is the block input, which has dimensions $W \times H \times N_{channels} \times T \times N_{batch}$, being W and H the spatial dimensions, T the temporal dimension, $N_{channels}$ the number of channels and N_{batch} the batch size. The $\hat{z}$ is the instance-normalized output and ϵ is a trainable parameters accounting for the bias. The t refers to the t-th frame of the temporal clip and n is the n-th temporal clip in the batch.

The second convolution in our dense block keeps normalization at batch level, to preserve *segmentor* generalization capability (Pan et al., 2018; Nam and Kim, 2018).

The *segmentor* is trained by comparing its output against the gold standard segmentation at temporal-clip level using the per-pixel binary cross entropy

(BCE), defined as:

$$BCE(y, S(x)) = \\ -\frac{1}{IHW}\sum_{i}^{w_{length}}\sum_{h}^{H}\sum_{w}^{W}(y_{i,w,h})\cdot log(S(x)_{i,w,h}) \\ +(1-y_{i,w,h})\cdot log(1-S(x)_{i,w,h}) \quad (4)$$

where $y_{i,w,h}$ and $S(x)_{i,w,h}$ denote the gold standard value and the corresponding prediction of the *segmentor* at pixel location (w, h) in the i-th frame of the temporal clip. The BCE is minimized in an adversarial fashion during training, as explained in Sec. 2.3.

2.2. Critic

The *critic* is inspired by that proposed in Casella et al. (2020). It is composed by two branches, as described in Table 1 and shown in Fig. 2, for extracting features from both the gold-standard segmentation and the *segmentor* output. Each branch consists of a standard UNet encoding path (i.e., without dense blocks) with batch-normalized convolution layers. Each branch is used for feature extraction at different scales from each input.

In the proposed architecture, the *critic* implements 3D convolutions for processing the temporal information while extracting features. We decided to keep the *critic* architecture similar to its original implementation because the role of the *critic* is to provide a shape constraining mechanism for the *segmentor* output. We decided to keep the *critic* architecture similar to its original implementation in Casella et al. (2020) because the role of the *critic* is to provide provide a shape-constraining mechanism on the *segmentor* to enforce segmentation consistency across sequential frames, and thus preserve the membrane macro-appearance. The use of dense blocks would have introduced unnecessary complexity with an increase in memory requirements.

The gold standard branch of the *Critic* takes as input the time clip masked (i.e., pixel-wise multiplication) by the gold standard (y). The *segmentor* branch takes as input x masked by the output of the *segmentor* ($S(x)$). The two masked

inputs are processed by the network in order to get two feature vectors. These are compared using the mean absolute error (MAE), defined as:

$$MAE(C(x \cdot y), C(x \cdot S(x))) = \frac{\sum_{i}^{w_{length}} \sum_{j}^{M} |C_j(x_i \cdot y_i) - C_j(x_i \cdot S(x_i))|}{w_{length} \cdot M} \quad (5)$$

where C is the output of the critic (i.e., the feature vector) and M is the length of the feature vector. The MAE is minimized in an adversarial fashion during training, as explained in Sec. 2.3.

2.3. Adversarial training strategy

We train our framework from scratch in an adversarial fashion. The *segmentor* and *critic* layers are initialised using He normal initialization (He et al., 2015). The stochastic gradient descent is used as optimizer, to minimize an adversarial loss (L), which sums up the contribution of two loss functions, i.e., the BCE loss from the *segmentor*, and the MAE from the *critic*:

$$L = BCE[y, S(x)] + MAE[C(x \cdot y), C(x \cdot S(x))] \quad (6)$$

The two terms of the loss function span in two different ranges. While there is a possible risk of divergence of the loss during training, the introduction of hyper parameters may allow to balance the action of the two terms in the loss function avoiding possible divergences, However, this never occurred in our experiments.

3. Experimental design

3.1. Dataset

To experimentally evaluate our two research hypotheses, we collected a dataset of 20 fetoscopic videos acquired during 20 different surgical procedures for treating TTTS in 20 women. The videos had a frame size of 720×576 pixels, with an acquisition frame rate of 25 frames per second. From each video, we extracted 100 consecutive frames among those in which the inter-fetal membrane

Table 2: Summary of the ablation study described in Sec. 3.4: *E1: 2D vanilla segmentor*, *E2: 3D vanilla segmentor*, *E3: 2D vanilla adversarial framework*, *E4: 2D adversarial framework*. The work in Casella et al. (2020), which is the closest to ours, is shown, too.

	Dense blocks	Instance normalization	3D convolution	Adversarial training
Casella et al. (2020)				X
E1	X			
E2	X		X	
E3	X			X
E4	X	X		X
Proposed	X	X	X	X

Table 3: Results of 3-fold cross-validation for $E1$, $E2$, $E3$, $E4$ and [Casella et al., 2020] in the ablation study. Segmentation Accuracy (*Acc*), Dice Similarity Coefficient (*DSC*) and Sensitivity (*Sens*) on the test set are reported in terms of mean ± standard deviation. The best results are highlighted in bold.

Framework	Metric	Fold 1	Fold 2	Fold 3	Overall
[Casella et al., 2020]	*Acc*	.8320 ± .1787	.8182 ± .1480	.9454 ± .0893	.8652 ± .1435
	DSC	.6414 ± .3616	.6483 ± .3225	.8734 ± .2550	.7210 ± .3161
	Sens	.6264 ± .3611	.6612 ± .3441	.8781 ± .2700	.7219 ± .3275
2D vanilla segmentor (*E1*)	*Acc*	.8597 ± .1548	.8302 ± .1350	.9218 ± .0893	.8706 ± .1402
	DSC	.6308 ± .3700	.6787 ± .2507	.9003 ± .1388	.7366 ± .2702
	Sens	.5762 ± .3668	.6758 ± .2687	.9525 ± .0453	.7348 ± .2638
3D vanilla segmentor (*E2*)	*Acc*	.8380 ± .1252	.8516 ± .0933	.9331 ± .1080	.8742 ± .1096
	DSC	.6123 ± .2921	.7208 ± .2494	.8681 ± .1524	.7338 ± .2386
	Sens	.5077 ± .2800	.7849 ± .2480	.9082 ± .1484	.7340 ± .2323
2D vanilla adversarial segmentor (*E3*)	*Acc*	.8701 ± .1515	.8090 ± .1614	.9371 ± .1013	.8720 ± .1406
	DSC	.6502 ± .3767	.6483 ± .3025	.9107 ± .1330	.7364 ± .2893
	Sens	.6150 ± .3898	.6509 ± .3297	.9406 ± .0810	.7355 ± .2985
2D adversarial framework (*E4*)	*Acc*	.9524 ± .0723	.8291 ± .1153	.9345 ± .0997	.9053 ± .0974
	DSC	.8964 ± .1715	.7457 ± .1947	.8697 ± .1821	.8373 ± .1830
	Sens	.8957 ± .1505	**.8910 ± .0743**	.9518 ± .1132	**.9128 ± .1169**
Proposed	*Acc*	**.9636 ± .0346**	**.8604 ± .0976**	**.9709 ± .0465**	**.9316 ± .0655**
	DSC	**.9111 ± .1179**	**.7698 ± .1993**	**.9530 ± .0614**	**.8780 ± .1383**
	Sens	**.9004 ± .1328**	.8495 ± .1804	**.9697 ± .0762**	.9065 ± .1366

Table 4: Results of the sliding window configuration tested in $E5$, $E6$ in the ablation study. Segmentation Accuracy (Acc), Dice Similarity Coefficient (DSC) and Sensitivity ($Sens$) on the test set are reported in terms of mean ± standard deviation. The best results are highlighted in bold.

Configuration	Δ_f	Δ_w	Acc	DSC	$Sens$
E5	1	1	$.9510 \pm .0377$	$.8985 \pm .0823$	$.9060 \pm .0776$
	2	1	$.9276 \pm .0806$	$.8425 \pm .1822$	$.8284 \pm .2074$
	3	1	$.9162 \pm .1079$	$.8389 \pm .1993$	$.8335 \pm .1869$
E6	0	2	$.9361 \pm .0841$	$.8827 \pm .1246$	$.8644 \pm .1368$
	0	3	$.9374 \pm .0695$	$.8804 \pm .1254$	$.8922 \pm .0919$
	0	4	$.9173 \pm .0908$	$.8431 \pm .1593$	$.8356 \pm .1295$
Proposed	0	1	$\mathbf{.9636 \pm .0346}$	$\mathbf{.9115 \pm .1179}$	$\mathbf{.9004 \pm .1328}$

was detected by an expert surgeon. The membrane was manually annotated in each frame under the supervision of the surgeon.

The videos, despite being acquired with the same equipment, showed high variability in terms of image noise, blur, field of view size, camera view, illumination, appearance, TTTS stage and placenta position, as shown in Fig. 1. This dataset, to the best of our knowledge, is the biggest dataset currently available for inter-fetal membrane segmentation. We will make it publicly available upon publication of the paper.

We used videos from 17 subjects for training (1700 frames), 3 of which were used as validation set (300 frames). The remaining 3 videos (300 frames), from 3 subjects that did not contribute to the training and validation set, were used for testing. We performed 3-fold cross-validation to evaluate the performance of the proposed segmentation framework.

Each frame was cropped to contain only the FoV of the fetoscope and, resized to 128x128 pixels both for smoothing noise and limiting memory usage.

3.2. Parameter setting

Temporal clips for training were built using the sliding window algorithm with $w_{length} = 4$, $\Delta_f = 0$ and $\Delta_w = 1$. We kept $\Delta_w = 1$ and $\Delta_f = 0$ as in Colleoni et al. (2019); Moccia et al. (2019); Hou et al. (2017) to have overlapping clips, and increasing the amount of clips available for training. Although in Colleoni et al. (2019); Hou et al. (2017) a $w_{length} = 8$ was used, we used $w_{length} = 4$ due to the higher complexity of our framework, which required higher memory usage and computational power. Validation and testing temporal clips were built using the same parameters but with $\Delta_w = 4$ (i.e., without overlap). This resulted in 1649 temporal clips (1358 for training, and 291 for validation) and 75 temporal clip for testing.

With our frame size of 128x128 pixels, the *critic* produced the two vectors with a dimension of 1397760 features.

During training, at each iteration step, each (unitary) batch was augmented with random rotation in range $(-25°, +25°)$, horizontal and vertical flip, and scaling with a scaling factor in range $(0.5, 1.5)$. The best model among epochs was chosen as the one that provided the best segmentation performance in terms of Dice Similarity Coefficient (DSC) on the validation set:

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{7}$$

We trained the proposed framework using TensorFlow on a GeForce RTX2080 TI (11GB) for 300 epochs for each fold, with an initial learning rate of 10^{-2}. Due to memory constraints, we fed the network with unitary batches. Each epoch lasted approximately 800s, for a total of about 70 hours to complete the training of one fold.

3.3. Performance metrics

For evaluating the segmentation performance on the test set, we computed, for each frame, the average DSC, Accuracy (Acc) and Sensitivity ($Sens$) be-

tween the prediction and gold standard masks:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Sens = \frac{TN}{TN + FP} \tag{9}$$

where TP and TN are the number background and membrane pixels correctly identified, whereas FP and FN are the background and membrane pixels that are misclassified. The Mann–Whitney–Wilcoxon test on Acc and DSC, both imposing a significance level (p) equal to 0.05, were used to assess whether or not remarkable differences existed between the tested architectures.

3.4. Ablation studies

We compared the results of the proposed framework against those of the adversarial network presented in Casella et al. (2020), which is the closest work with respect to ours. Considering that a comprehensive comparison with standard state of the art approaches (e.g., UNet (Ronneberger et al., 2015) and ResNet (He et al., 2016)) is already provided in Casella et al. (2020), we here focused on the ablation studies.

To assess the impact of each component of the overall framework, we performed the following experiments (Table 2):

Experiment 1 (E1): We implemented a 2D vanilla segmentor, which is the segmentor described in Sec. 2.1 with the standard dense blocks proposed in Huang et al. (2017) and 2D kernel convolutions, as baseline for our comparisons. The training loss function was the one reported in Eq. 6 but without the MAE term.

Experiment 2 (E2): To see if the adversarial training had an impact on segmentation performance, we compared the proposed framework with the 3D segmentation network without the critic, hence trained in a non adversarial fashion (3D vanilla segmentor)..

Experiment 3 (E3): To asses the impact of dense blocks combined with the adversarial training, we trained the 2D vanilla segmentor using the adversarial training strategy described in Sec. 2.3 (2D vanilla adversarial framework), thus including the 2D version of our critic network.

Experiment 4 (E4): To understand if the temporal information and instance normalization affected the segmentation performance, we compared the proposed framework with its version with 2D convolution 2D adversarial framework.

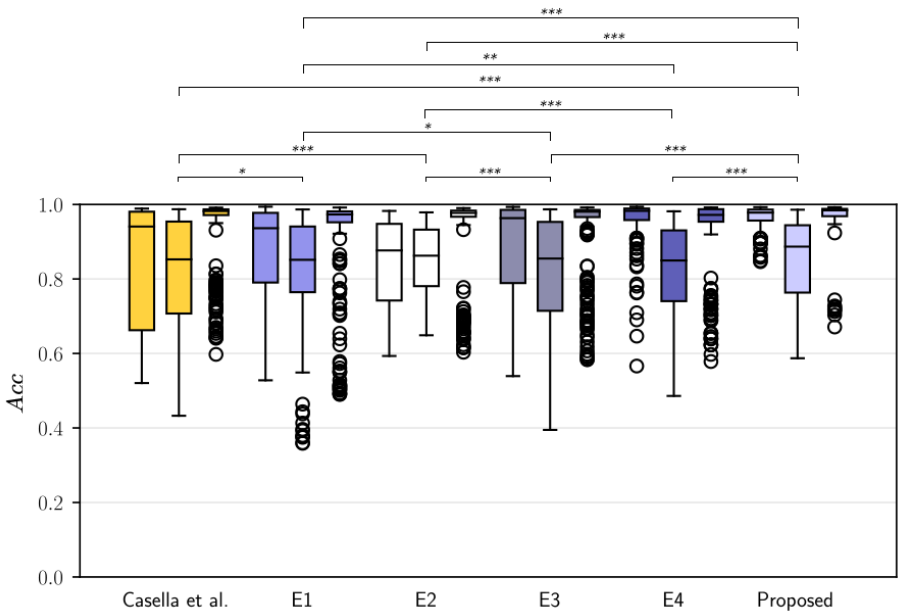


Figure 4: Boxplot of performance comparison between *E1*, *E2*, *E3*, *E4* in the ablation study and Casella et al. (2020). The comparison is shown in terms of accuracy (*Acc*) for each fold. Black asterisks highlight significant differences between the different architectures (Mann–Whitney–Wilcoxon) ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$).

We performed ablation studies on the parameters of the sliding window algorithm:

Experiment 5 (E5): To assess how Δ_f affected the training process, we trained our framework using clips generated with Δ_f equal to 1, 2 and 3.

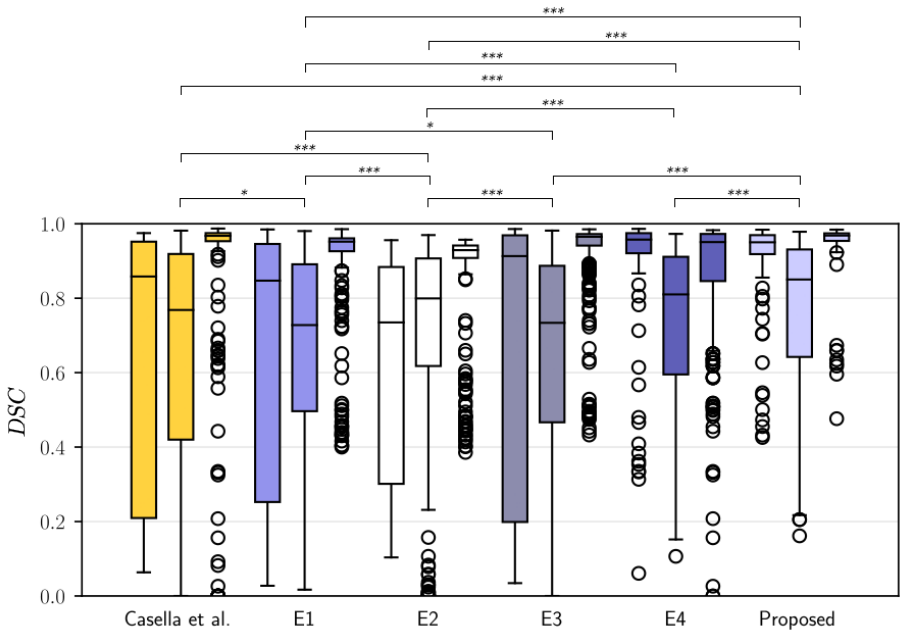
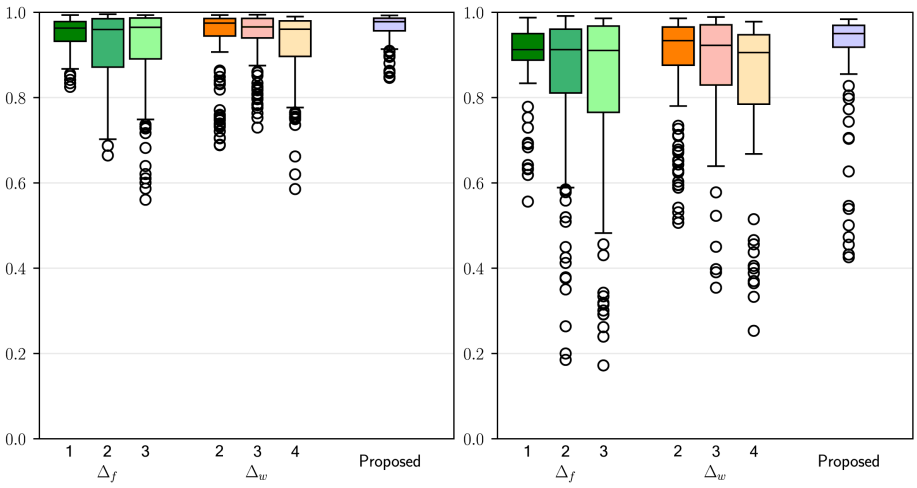


Figure 5: Boxplot of performance comparison between *E1*, *E2*, *E3*, *E4* in the ablation study and Casella et al. (2020). The comparison is shown in terms of Dice similarity coefficient (*DSC*) for each fold. Black asterisks highlight significant differences between the different architectures (Mann–Whitney–Wilcoxon) ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$).

Experiment 6 (E6): To assess how Δ_w influenced the training process, we trained our framework using temporal clips with Δ_w equal to 2, 3 and 4 (i.e., no overlap). Finally, we tested the behaviour of our architecture when processing frames where the membrane was not present. To this goal, we extracted short sequences of 40 frames without membrane, from 4 videos in our dataset.

4. Results

The proposed framework processed ≈ 80 temporal clips per second during inference. The best segmentation result among all folds was achieved by the proposed framework with $Acc = 0.9316 \pm 0.0655$, $DSC = 0.8780 \pm 0.1383$ and $Sens = 0.9065 \pm 0.1366$. Mean metric values are reported, with standard deviation in brackets. Figure 4 and Fig. 5 show the boxplots of the performance



(a) *Acc* (b) *DSC*

Figure 6: (a-b) Boxplot of performance for the sliding window ablation study for each value of the parameters (i.e. Δ_f (*E5*) and Δ_w (*E6*)) shown in terms of (c) accuracy (*Acc*) and (d) Dice Similarity Coefficient (*DSC*). The proposed method refers to $\Delta_f = 0$ and $\Delta_w = 1$.

comparison between our results and those achieved with Casella et al. (2020) (i.e. the closest work with respect to ours) and the ablation models presented in *E1*, *E2*, *E3* and *E4*. For robustness and performance evaluation, we performed a 3-fold cross-validation of the architectures in the ablation study, as described in Sec. 3.4. Quantitative results of the 3-fold cross-validation are shown in Table 3. Detailed results for each video are presented in the supplementary materials.

The work in Casella et al. (2020) achieved the worst results among all folds with a *DSC* of 0.7210 ± 0.3161. The *2D vanilla segmentor* (*E1*) showed comparable results with respect to the *3D vanilla segmentor* (*E2*) and the *2D vanilla adversarial segmentor* (*E3*), with a *DSC* of 0.7366 ± 0.2702, 0.7338 ± 0.2386 and 0.7364 ± 0.2893, respectively. The *2D adversarial framework* (*E4*) showed an improvement in results achieving the closest performance with respect to ours, with an average $DSC = 0.8373 \pm 0.1830$. The *2D adversarial framework* and the proposed framework outperformed the mean *DSC* obtained by *E1*, *E2* and *E3* by, at least, 0.1007 and 0.1414, respectively.

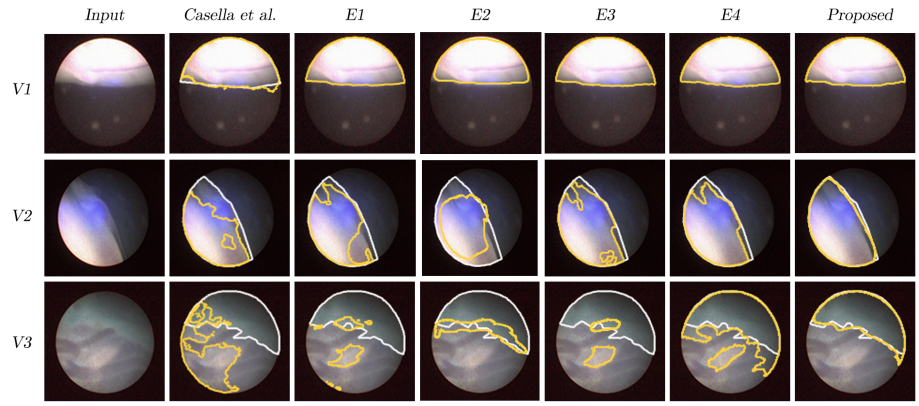


Figure 7: Sample segmentation results obtained on the test set using the architectures described in the ablation study (Sec. 3) and in Casella et al. (2020). The *gold standard* and predicted segmentation are highlighted in white and yellow, respectively. Each row refers to a different video, while each column refers to a different experiment: input: original input frame, Casella et al. (2020): our previous work, $E1$: *2D vanilla segmentor*, $E2$: *3D vanilla segmentor*, $E3$: *2D vanilla adversarial framework*, $E4$: *2D adversarial framework*, and proposed: *3D adversarial framework*.

Table 4 and Fig. 6 show the results achieved by the proposed framework when using different parameters of the sliding window algorithm to generate the training clips, as explained in Sec. 3.4. The best results in both *E5* and *E6* was achieved by the proposed set of sliding window parameters (i.e., $\Delta_f = 0$ and $\Delta_w = 1$). The lowest performance, when testing *E5*, was achieved with $\Delta_f = 3$. For *E6*, the lowest performance was the one with $\Delta_w = 4$ (no overlap between temporal clips).

Visual samples for the tested models are shown in Fig. 7. Each row shows the segmentation results of a sample frame extracted from the testing videos ($V1$, $V2$ and $V3$) from *Fold 1*.

In Fig. 8, qualitative results for three consecutive frames in a clip are shown both for the *2D adversarial framework* and the proposed framework. The white and yellow borders highlight the *gold standard* and the segmentation results.

5. Discussion and conclusions

This paper introduced a shape-constrained adversarial framework with instance-normalized spatio-temporal features to perform automatic inter-fetal membrane segmentation in fetoscopic video clips, while tackling the high illumination variability in fetoscopic videos.

The proposed *3D adversarial framework* provided accurate and robust segmentation, reaching the highest mean DSC value (0.8780) among the 3 folds, as well as the lowest DSC standard deviation when compared with the performed ablation studies. This confirmed our research hypotheses that instance normalization and spatio-temporal features can tackle the peculiar challenges of fetoscopic videos, listed in Sec. 1 and shown in Fig. 1.

This was not the case for the other tested approaches in the ablation study. The *2D vanilla segmentor* (*E1*), which has been reported to perform well with homogeneous illumination in closer fields (Zhou et al., 2020), showed one of the lowest performance in the ablation study, due to the high illumination variability of fetoscopic videos.

Its mean DSC (0.7366) was similar to that of the framework presented in Casella et al. (2020) (0.7210). Hence, the dense non-adversarial *segmentor* achieved similar performance with respect to the residual (i.e., non-dense) *segmentor* trained in adversarial fashion. This may be explained considering that the instance normalization and adversarial training address different aspects (i.e., feature connectivity and membrane-shape consistency among consecutive frames).

By exploiting both the adversarial training and dense blocks, the *2D vanilla adversarial segmentor* achieved a mean DSC value of (0.7364).

The *3D vanilla segmentor* (*E2*) configuration achieved a mean DSC among all folds of 0.7338, comparable with *E1* and Casella et al. (2020). We noticed that 3D convolution alone was not able to boost segmentation consistency, as the results are comparable with the *2D vanilla adversarial framework* (*E3*). This can be explained because 3D convolutions and adversarial training emphasize

two different aspects, temporal consistency and shape constraint. Both aspects led to improvements in membrane segmentation performance.

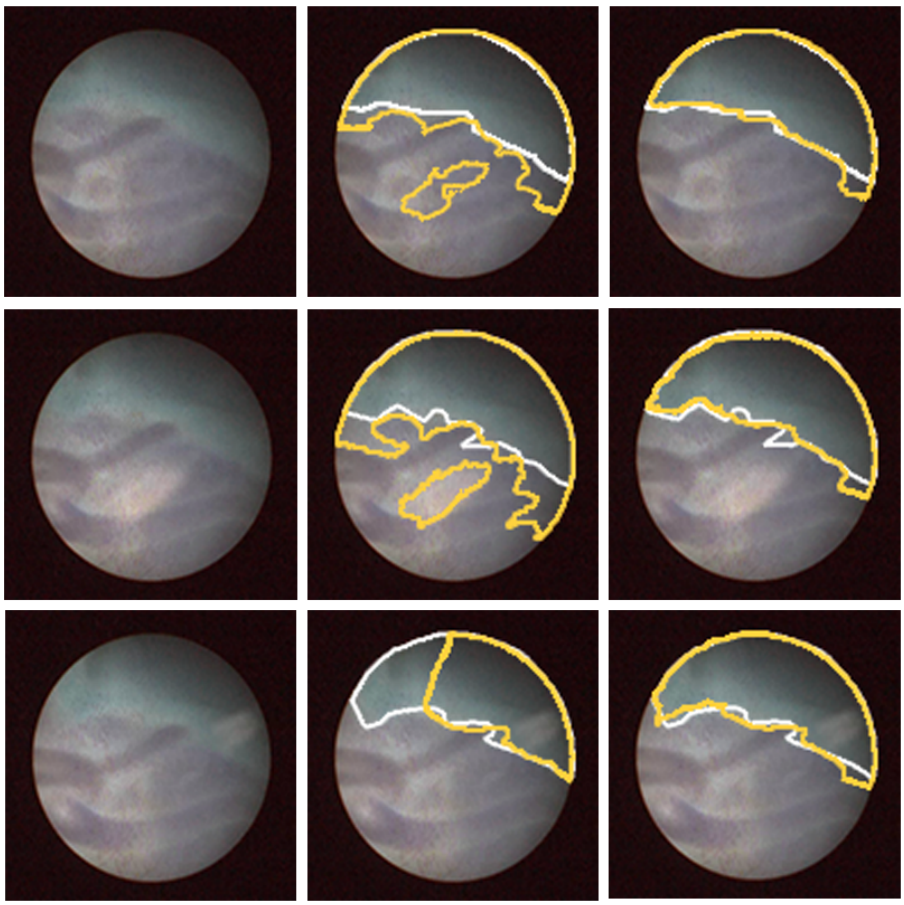


Figure 8: Sample results of inter-fetal membrane segmentation for three consecutive frames in a clip. Results are shown for the (second column) *2D adversarial framework* and (third column) the proposed framework. The *gold standard* and segmentation prediction are highlighted in white and yellow, respectively.

However, the strong illumination variability in fetoscopic videos still represented an issue from some clips. This issue was tackled with the *2D adversarial framework*, which included the dense blocks with instance normalization, that achieved an average $DSC = 0.8373$.

The *3D adversarial framework* further improved the overall segmentation performance (Fig. 4 and Fig. 5). This may be explained considering that the introduction of 3D kernels allow the processing the temporal information naturally encoded in endoscopic videos, improving segmentation consistency across sequential frames and thus reducing the inter-quartile ranges and the number of outliers. The performance obtained for *Fold 2* is the lowest, as shown in Table 3.

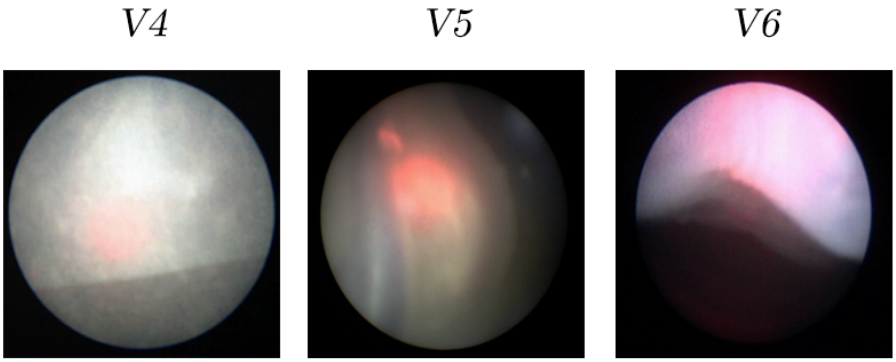


Figure 9: Sample frame from patients in the test set of *Fold 2*. *V4* is a patient with posterior placenta and, *V5* and *V6* are patients with anterior placenta. Performance results for this fold are shown in Table 3, detailed metrics for each video are shown in the supplementary materials.

This may be explained considering that several patients in *Fold 2* have anterior placentas, some examples are shown in Fig. 9. In anterior placentas, the interfetal membrane can be folded several times, bringing additional complexity. In the training of *Fold 2*, there were few patients with anterior placentas, posing issues to segmentation generalizability.

The dataset presented in this work included a larger number of TTTS patients (20 patients) than that of Casella et al. (2020) (7 patients), granting higher variability in terms of membrane shape, texture, color and illumination, as described in Sec. 3.1. The method proposed in our previous work Casella et al. (2020) was not able to fully tackle such variability, highlighting the need for more advanced solutions.

From the visual-analysis perspective of *Fold 1*, for the first testing video

(Fig. 7, *V1*) all the models performed well. This may be due to the fact that the inter-fetal membrane is highly contrasted with respect to the background. In the second video (*V2*), the presence of the laser-diode light (blue spot) strongly hampered the detection of the membrane for the models in Casella et al. (2020), *E1* and *E2*. The inclusion of the adversarial training (*E3*), and then of the instance normalization (*E4*) increased the segmentation accuracy. By combining the adversarial training with the processing of the temporal information encoded in the video clips, the proposed framework further improved the segmentation performance. The video frames shown in Fig 7 (*V3*), presented a quite different illumination level with respect to the others in the dataset (Fig. 1). In this case, combining instance normalization with the processing of the temporal information was crucial to provide accurate segmentation. This may be also seen in Fig. 8, where the *2D adversarial framework* was not able to preserve the shape of the membrane across sequential frames.

The results achieved by the models trained with different sliding window parameters showed the importance of preserving temporal connectivity in the temporal clips used for training. As showed in Fig. 6, as the sliding windows parameters Δ_f (*E5*) and Δ_w (*E6*) increased, the segmentation performance decreased. This may be explained considering that the proposed framework enforce pixel connectivity in the temporal dimension.

To verify that the trained architecture was not biased to produce segmentation masks for frames where the membrane was not present, we extracted 4 short video sequences (40 frames each), from 4 original videos in the dataset, in which the membrane was not visible. These videos were not present in the training set. A sample from each video is shown in Fig. 10. We noted that in the 84.38% of the cases (135 frames) the network did not produce false-positive segmentation. However, for 29 frames (15.62%), the colour and the texture of the background were similar to those of the membrane, yielding to the false-positive segmentation. This limitation could be solved by a preliminary frame-selection step (Bano et al., 2020), where only frames in which the membrane is visible are further processed by the proposed framework.

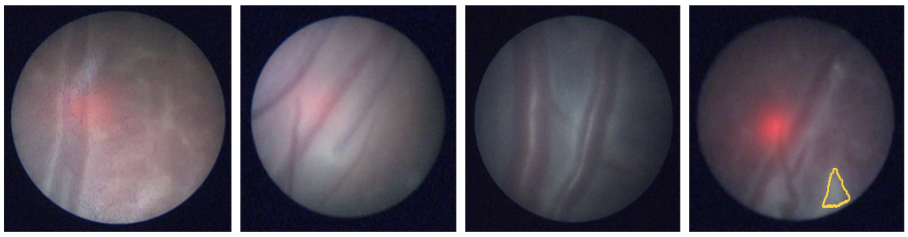


Figure 10: Sample frames in which the membrane is not present.Each frame was extracted from a video not used to train the network, as described in Sec. 3.4. The predicted segmentation is highlighted in yellow.

A possible limitation of the proposed framework can be seen in segmentation inaccuracies in clips with rapid and wide movements of the fetoscope, where the membrane changes in appearance very quickly. However, this rarely happens during these very delicate operations as rapid motions pose a risk to the patients. In such cases, the temporal connectivity introduced to guarantee consistency across consecutive frames can affect the accuracy of segmentation negatively. A possible solution to attenuate this limitation could be to exploit long short-term memory networks to take into account a wider time horizon (Wang et al., 2017).

Another possible limitation may be seen when processing frames with membrane occlusions due to fetal movements, umbilical cord and glare from the photocoagulation laser. While we did not address this specific aspect in this paper, a possible solution to tackle it would be to rely on the automatic selection of occlusion-free frames (Bano et al., 2020).

A limitation of the experimental protocol may be seen in the dataset size, which could be increased to further validate the proposed framework. However, the dataset is already the largest available for inter-fetal membrane segmentation from fetoscopic videos, and we will make it publicly available to further research in this field. The proposed framework also has large potential to be translated to other anatomical districts, where complex environment and high variability still affect the learning capability of state-of-the-art FCNNs.

To conclude, the achieved results suggest that the proposed approach may be

effective in supporting surgeons in the identification of the inter-fetal membrane in fetoscopic videos. This may have a positive impact on TTTS surgery, by lowering the surgery duration and, as a consequence, by reducing surgeons' mental workload and patients' risks.

Ethical standards

The proposed study is a retrospective study. Data used for the analysis were acquired during actual surgery procedures and then were anonymized to allow researchers to conduct the study. All the patients gave their consent on data processing for research purpose. The study fully respects and promotes the values of freedom, autonomy, integrity and dignity of the person, social solidarity and justice, including fairness of access. The study was carried out in compliance with the principles laid down in the Declaration of Helsinki, in accordance with the Guidelines for Good Clinical Practice.

Conflict of Interest

No benefits in any form have been or will be received from a commercial party related directly or indirectly to the subjects of this manuscript.

References

Almoussa, N., Dutra, B., Lampe, B., Getreuer, P., Wittman, T., Salafia, C., Vese, L., 2011. Automated vasculature extraction from placenta images, in: Medical Imaging: Image Processing, International Society for Optics and Photonics. p. 79621L.

Bano, S., Vasconcelos, F., Amo, M.T., Dwyer, G., Gruijthuijsen, C., Deprest, J., Ourselin, S., Vander Poorten, E., Vercauteren, T., Stoyanov, D., 2019. Deep sequential mosaicking of fetoscopic videos, in: International Conference

on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 311–319.

Bano, S., Vasconcelos, F., Vander Poorten, E., Vercauteren, T., Ourselin, S., Deprest, J., Stoyanov, D., 2020. FetNet: a recurrent convolutional network for occlusion identification in fetoscopic videos. International Journal of Computer Assisted Radiology and Surgery 15, 791–801.

Baschat, A., Chmait, R.H., Deprest, J., Gratacós, E., Hecher, K., Kontopoulos, E., Quintero, R., Skupski, D.W., Valsky, D.V., Ville, Y., 2011. Twin-to-twin transfusion syndrome (TTTS). Journal of Perinatal Medicine 39.

Beck, V., Lewi, P., Gucciardo, L., Devlieger, R., 2012. Preterm prelabor rupture of membranes and fetal survival after minimally invasive fetal surgery: a systematic review of the literature. Fetal Diagnosis and Therapy 31, 1–9.

Casella, A., Moccia, S., Frontoni, E., Paladini, D., De Momi, E., Mattos, L.S., 2020. Inter-foetus Membrane Segmentation for TTTS Using Adversarial Networks. Annals of Biomedical Engineering 48, 848–859.

Colleoni, E., Moccia, S., Du, X., De Momi, E., Stoyanov, D., 2019. Deep Learning Based Robotic Tool Detection and Articulation Estimation with Spatio-Temporal Layers. IEEE Robotics and Automation Letters 4, 2714–2721.

Daga, P., Chadebecq, F., Shakir, D.I., Herrera, L.C.G., Tella, M., Dwyer, G., David, A.L., Deprest, J., Stoyanov, D., Vercauteren, T., Ourselin, S., 2016. Real-time mosaicing of fetoscopic videos using {SIFT}, in: Webster, R.J., Yaniv, Z.R. (Eds.), Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling, SPIE.

Gaisser, F., Peeters, S.H.P., Lenseigne, B.A.J., Jonker, P.P., Oepkes, D., 2018. Stable image registration for in-vivo fetoscopic panorama reconstruction. Journal of Imaging 4.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: IEEE International Conference on Computer Vision, pp. 1026–1034.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Hou, R., Chen, C., Shah, M., 2017. An End-to-end 3D Convolutional Neural Network for Action Detection and Segmentation in Videos. arXiv preprint arXiv:1712.01111 .

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 2261–2269.

Jegou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1175–1183.

Maier-Hein, L., Vedula, S.S., Speidel, S., Navab, N., Kikinis, R., Park, A., Eisenmann, M., Feussner, H., Forestier, G., Giannarou, S., Hashizume, M., Katic, D., Kenngott, H., Kranzfelder, M., Malpani, A., März, K., Neumuth, T., Padoy, N., Pugh, C., Schoch, N., Stoyanov, D., Taylor, R., Wagner, M., Hager, G.D., Jannin, P., 2017. Surgical data science for next-generation interventions. Nature Biomedical Engineering 1, 691–696.

Milletari, F., Navab, N., Ahmadi, S.A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: International Conference on 3D Vision, IEEE. pp. 565–571.

Moccia, S., Migliorelli, L., Carnielli, V., Frontoni, E., 2019. Preterm infants' pose estimation with spatio-temporal features. IEEE Transactions on Biomedical Engineering .

Moccia, S., Romeo, L., Migliorelli, L., Frontoni, E., Zingaretti, P., 2020. Supervised {CNN} Strategies for Optical Image Segmentation and Classification in Interventional Medicine, in: Deep Learners and Deep Learner Descriptors for Medical Applications. Springer, pp. 213–236.

Nam, H., Kim, H.E., 2018. Batch-instance normalization for adaptively style-invariant neural networks, in: Advances in Neural Information Processing Systems, pp. 2558–2567.

Pan, X., Luo, P., Shi, J., Tang, X., 2018. Two at once: Enhancing learning and generalization capacities via ibn-net, in: European Conference on Computer Vision, pp. 464–479.

Peter, L., Tella-Amo, M., Shakir, D.I., Attilakos, G., Wimalasundera, R., Deprest, J., Ourselin, S., Vercauteren, T., 2018. Retrieval and registration of long-range overlapping frames for scalable mosaicking of in vivo fetoscopy. International Journal of Computer Assisted Radiology and Surgery 13, 713–720.

Quintero, R.A., 2003. Twin{\textendash}twin transfusion syndrome. Clinics in Perinatology 30, 591–600.

Roberts, D., Neilson, J.P., Kilby, M.D., Gates, S., 2014. Interventions for the treatment of twin-twin transfusion syndrome. Cochrane Database of Systematic Reviews 2014.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention , 234–241.

Sadda, P., Imamoglu, M., Dombrowski, M., Papademetris, X., Bahtiyar, M.O., Onofrey, J., 2019. Deep-learned placental vessel segmentation for intraoperative video enhancement in fetoscopic surgery. International Journal of Computer Assisted Radiology and Surgery 14, 227–235.

Tella-Amo, M., Peter, L., Shakir, D.I., Deprest, J., Stoyanov, D., Vercauteren, T., Ourselin, S., 2019. Pruning strategies for efficient online globally consistent mosaicking in fetoscopy. Journal of Medical Imaging 6, 1.

Torrents-Barrena, J., Piella, G., Gratacos, E., Eixarch, E., Ceresa, M., Ballester, M.A.G., 2020. Deep Q-CapsNet Reinforcement Learning Framework for Intrauterine Cavity Segmentation in TTTS Fetal Surgery Planning. IEEE Transactions on Medical Imaging , 1–1.

Ulyanov, D., Vedaldi, A., Lempitsky, V., 2016. Instance Normalization: The Missing Ingredient for Fast Stylization.

Vasconcelos, F., Brandão, P., Vercauteren, T., Ourselin, S., Deprest, J., Peebles, D., Stoyanov, D., 2018. Towards computer-assisted TTTS: Laser ablation detection for workflow segmentation from fetoscopic video. International Journal of Computer Assisted Radiology and Surgery 13, 1661–1670.

Wang, X., Gao, L., Song, J., Shen, H., 2017. Beyond Frame-level CNN: Saliency-Aware 3-D CNN with LSTM for Video Action Recognition. IEEE Signal Processing Letters 24, 510–514.

Xu, C., Xu, L., Brahm, G., Zhang, H., Li, S., 2018. Mutgan: Simultaneous segmentation and quantification of myocardial infarction without contrast agents via joint adversarial learning, in: Medical Image Computing and Computer Assisted Intervention, Springer International Publishing, Cham. pp. 525–534.

Zhou, S., Nie, D., Adeli, E., Yin, J., Lian, J., Shen, D., 2020. High-Resolution Encoder-Decoder Networks for Low-Contrast Medical Image Segmentation. IEEE Transactions on Image Processing 29, 461–475.