Estimation of Interaction Forces in Robotic Surgery using a Semi-Supervised Deep Neural Network Model

Arturo Marban\textsuperscript{1,2}, Vignesh Srinivasan\textsuperscript{2}, Wojciech Samek\textsuperscript{2}, Member, IEEE, Josep Fernández\textsuperscript{1}, Alicia Casals\textsuperscript{1}, Senior Member, IEEE

Abstract—Providing force feedback as a feature in current Robot-Assisted Minimally Invasive Surgery systems still remains a challenge. In recent years, Vision-Based Force Sensing (VBFS) has emerged as a promising approach to address this problem. Existing methods have been developed in a Supervised Learning (SL) setting. Nonetheless, most of the video sequences related to robotic surgery are not provided with ground-truth force data, which can be easily acquired in a controlled environment. A powerful approach to process unlabeled video sequences and find a compact representation for each video frame relies on using an Unsupervised Learning (UL) method. Afterward, a model trained in an SL setting can take advantage of the available ground-truth force data. In the present work, UL and SL techniques are used to investigate a model in a Semi-Supervised Learning (SSL) framework, consisting of an encoder network and a Long-Short Term Memory (LSTM) network. First, a Convolutional Auto-Encoder (CAE) is trained to learn a compact representation for each RGB frame in a video sequence. To facilitate the reconstruction of high and low frequencies found in images, this CAE is optimized using an adversarial framework and a $\text{L1}$-loss, respectively. Thereafter, the encoder network of the CAE is serially connected with an LSTM network and trained jointly to minimize the difference between ground-truth and estimated force data. Datasets addressing the force estimation task are scarce. Therefore, the experiments have been validated in a custom dataset. The results suggest that the proposed approach is promising.

Index Terms—Vision Based Force Sensing, Robotic Surgery, Deep Neural Networks, Semi-Supervised Learning.

I. INTRODUCTION

Force feedback is a desired feature in Robot-Assisted Minimally Invasive Surgery systems. It allows the integration of the “sense of touch”, resulting in potential benefits. For instance, improved manipulation of human soft-tissues avoiding damage due to excessive applied forces. However, the integration of force feedback in surgical robotic systems still remains an open problem [1][2]. Advances in the fields of computer vision and artificial intelligence have resulted in new approaches to address VBFS. VBFS enables the estimation of interaction forces between surgical instruments and soft-tissue by processing video sequences. Such data is easily provided by surgical robotic systems, nonetheless, its interpretation is challenging.

Different methods have been proposed to address VBFS in robotic-assisted surgery scenarios. They estimate forces from (monocular/stereo) video sequences relying on an accurate modeling of soft-tissues’ deformation (in 3D space) caused by the interaction with surgical instruments. Moreover, in VBFS, the processing of the surgical tool motion is beneficial (i.e. the tool-tip trajectory). A VBFS approach was investigated in [3] using a simplified scenario consisting of a rubber membrane. Its deformation was recovered by tracking nodal displacements and a finite element method was used to model the mechanical relationship between deformation and force. A more realistic scenario was studied in [4], which addresses monocural force estimation using a real lamb liver as experimental material. The authors proposed a virtual template to model soft-tissue surface deformation. However, it is assumed that the soft-tissue surface behaves as a smooth function with local deformation. The relationship between force and penetration depth caused by the surgical tool was modeled based on a stress-strain bio-mechanical model. VBFS applied to neurosurgery was investigated in [5] and [6]. In [5], soft-tissue surface deformation is computed using a depth map extracted from stereo-endoscopic images. Then, a surface mesh based on spring-damper models processes this information to render force as output. In contrast, the authors in [6] developed a method based on quasi-dense stereo correspondence to recover surface deformation from stereo video sequences. Afterward, force is estimated from the surgical tool displacement (which is extracted from the deformation data), using a 2nd order polynomial model. Models based on neural networks have been investigated in recent years. For instance, [7] proposed a 3D lattice in a minimization framework for modeling the complex deformation of soft-tissues. Furthermore, a recurrent neural network was designed to estimate force by processing the information provided by this lattice in addition to the surgical tool motion. Subsequent notable works by the same author include [8] and [9], in which the recurrent neural network described in [7] is improved by designing a model based on the Long-Short Term Memory (LSTM) network architecture [10], achieving high accuracy in the estimation of forces (in 3D space).

The literature review of VBFS in robotic-assisted surgery, reveals that the proposed neural network models have been designed in a Supervised Learning (SL) setting. However, the advantages of using a Semi-Supervised Learning (SSL)
Many approaches remain unexplored. SSL represents an interesting avenue of research, given that unlabeled data is easily available (i.e. video sequences) and labeled data is scarce (i.e. video sequences in addition to ground-truth force data). For this purpose, Unsupervised Learning (UL) techniques are essential. They allow learning meaningful representations from unlabeled raw data. In this context, an approach based on Generative Adversarial Networks (GAN) [11] has not yet been investigated for VBFS in robotic-assisted surgery. GANs are generative models that consist in two competing neural networks with different objectives: a generator $G$ and a discriminator $D$. The goal of $G$ is to “fool” $D$ by learning to generate samples that resemble the real data (ground-truth). In contrast, the goal of $D$ is to distinguish between real (ground-truth) and fake (samples generated by $G$) data. As the training process evolves, $G$ learns the probability distribution of the real data. Auto-encoders are neural networks with fully connected layers that encode high-dimensional data into a latent space and decode this information, reconstructing the input data in its output [12][13]. A better model for processing data with spatial correlations (i.e. images) is a Convolutional Auto-Encoder (CAE). In [14], this model is studied for feature vector extraction and pre-training of Convolutional Neural Networks (CNN). The authors concluded that this model can learn biologically plausible filters. In addition, it was found that optimizing a pre-trained CNN tends to outperform the same model with its parameters initialized from scratch. When few labeled data is available, pre-training CNNs with an UL approach can help in designing models in a SL setting. Regarding the reconstructed data, its quality is affected by the design of the loss function used to optimize the CAE model. To improve this quality and learn better representations in the latent space, the traditional CAE model can be extended to a GAN framework. For this purpose, a CAE model can be designed and optimized based on several design choices described in [15], [16], [17], [18] and [19]. An adversarial auto-encoder is proposed in [15] which shapes the distribution of the latent space using a GAN framework. The authors in [16] designed a generative model as a combination of a variational auto-encoder and a GAN framework. In particular, they pointed out that a loss function designed with an element-wise metric (i.e. squared error) is simple but not suitable for image data. Instead, they suggest a feature-wise metric to measure image similarity. Specifically, they designed a loss function that uses the hidden representations of the layers in the GAN discriminator, which improves the quality of the generated samples. The task of image-to-image translation is addressed in [17]. In that work, the authors suggest the design of a discriminator network that processes image patches and outputs a probability map instead of a single scalar value (as defined in the original GAN framework [11]). This approach improves the quality of the samples rendered by the investigated model. Furthermore, in this work, low and high frequencies of image data are modeled by a loss function designed based on the $L1$-loss and GAN framework, respectively. This loss function design avoids the blurring effect on images rendered by the model due to the $L1/L2$-loss. This approach has also been investigated in the task of video frame prediction [18]. In [19], a GAN framework is proposed for transferring the texture of real into simulated images, while preserving the annotations of simulated images. The discriminator network used in this work shares some similarities with the model described in [17], suggesting that a discriminator network that outputs a probability map represents a suitable design choice for some applications.

In the present work, a model in a SSL setting is proposed for the estimation of forces in the context of robotic surgery. This model is composed of an encoder network serially connected with an LSTM network. It addresses the estimation of forces related to pushing actions (i.e. pressing the surgical tool against soft-tissues), which are essential in the execution of tasks such as the palpation of soft-tissues. The model is optimized in two stages. First, in the UL stage, a CAE is optimized in an adversarial framework using a large dataset of unlabeled video sequences describing interactions between surgical instruments and artificial soft-tissues. The CAE design and optimization is based on the works described in [15]-[19]. The objective of this stage is to design an encoder network as a feature extractor. The feature vectors computed by this neural network represent a learned representation of high dimensional data, such as video sequences. Subsequently, this encoder network is serially connected with an LSTM network and trained in a SL setting using fewer data than in the UL stage. In this stage, video sequences in addition to ground-truth force and tool data (i.e. surgical tool trajectory and grasper status) are available. The main contributions of this work are:

- In the SL stage, the impact of applying image processing operations to video sequences, such as mean normalization and space-time transformation, in the estimated force signal quality is investigated. This study shows the importance of highlighting motion in video sequences due to tool-tissue interactions.
- The effectiveness of using a loss function with two terms is investigated in the optimization of the model in the SL stage. The first term measures the distance between the ground-truth and estimated force signals (i.e. measured by root mean squared error), while the second term measures the distance between their gradients (i.e. derivative of the force signal with respect to time). This loss function design eases the modeling of smooth and sharp details found in force and torque signals.

II. METHODS

A. Dataset

Datasets addressing the force estimation task are scarce, therefore, the experiments have been validated in a custom dataset. It consists of video sequences, tool data and ground-truth interaction forces. An experimental platform was used for this purpose. In this platform, a slave robot manipulator (Stäubli RX60B) with an attached (motorized) surgical tool interacts with a digestive apparatus made of artificial soft-tissue (Silicone-Smooth On ECOFLEX 0030). Forty-four
video sequences (480 × 640 @ 50 FPS), totaling 4.31 hours, were recorded using 4 digital cameras (DFK 72BUC02). The tool data is described by the surgical tool-tip trajectory in the 3D space and its grasping status (i.e. opened/closed grasper), at each time instant. The interaction forces and torques between the surgical tool and artificial soft-tissues were acquired by a 6D force sensor (ATI Gamma SI-32-2.5). The force sensor resolution is: 0.00625 N for $f_x$ and $f_y$, 0.0125 N for $f_z$, and 0.0005 Nm for all the torques ($\tau_x$, $\tau_y$, and $\tau_z$). This sensor was attached at the robot manipulator’s end-effector and its z-axis was aligned with the surgical instrument shaft (see Fig. 3c). The measured forces and torques lie in the range $+2.5/-10$ N and $+/-5$ Nm, respectively.

B. Preprocessing of Video Sequences

The recorded video sequences were processed by tracking and extracting a region of interest of size $200 \times 300$ pixels from every frame. These image regions improve the visibility of the interaction between the surgical tool and soft-tissues. With this aim, mean normalization and space-time transformations were used. The mean frame normalization consists in computing a mean frame for every video sequence by averaging all the raw frames (with equal contribution). Each computed mean frame is subtracted from every corresponding video sequence. The result of this operation is the elimination of the static background present in video sequences, since it does not contribute to the learning process in the force estimation task. In the application of human pose estimation from video sequences (with static background), this operation was found beneficial [20]. The space-time transformation consists in creating a 3-channel image by concatenating gray-scale versions of the past, current, and next color frames along the channel dimension. This image representation encodes temporal information and was found useful in the estimation of sound from silent video sequences [21].

The three types of video frames investigated in the experiments are illustrated in Fig. 1. These are raw, space-time and full-processed frames. Raw video frames processed with the space-time transformation, referred to as space-time frames, emphasize motion from two sources: (i) camera motion and (ii) motion due to the interaction between surgical instruments and soft-tissues. In this representation, image regions where motion is present are rendered as colored pixels. In contrast, static image regions are shown as pixels in grayscale. Full-processed video frames, with mean normalization and space-time transformation, only emphasize the motion caused by the interaction of the surgical tool with soft-tissues. In this representation, only this type of motion is rendered as colored pixels, while the rest of the image is suppressed (i.e. constant gray color).

C. Semi-Supervised Model

The SSL model, composed of an encoder network and a LSTM network, is designed in two stages.

In the first stage, the encoder network $Enc$ maps an input image $X$ to a latent space vector $Z$. This encoding process is expressed as $Z = Enc(X)$. The reverse operation is carried out by the decoder network $Dec$, whose objective is to reconstruct the original image $X$ from the latent space $Z$. Thus, the reconstructed image is $\hat{X} = Dec(Z)$. The CAE is optimized in an adversarial framework using two discriminators. The first discriminator, represented by $Disc_x$, is a fully convolutional neural network that distinguishes between real ($X$) and reconstructed images ($\hat{X}$). The output of this discriminator is a 3 channel probability map, $P_x \in \mathbb{R}^{7 \times 10 \times 3}$. The second discriminator, $Disc_z$, is a fully connected neural network that outputs a single scalar probability, $P_z \in \mathbb{R}$. It classifies latent space vectors $Z$ as belonging to a standard normal distribution, $Z_{\text{real}} \sim \mathcal{N}(0, 1)$, or as generated by the encoder network, $Z = Enc(X)$. In Fig. 2a, the number of output feature maps corresponding to each convolutional layer are shown for $Enc$ (CE1-CE5), $Disc$ (CD1-CD5) and $Disc_z$ (C1-C6). For instance, CE1-64 indicates that layer CE1 outputs 64 feature maps. Analogously, the size of each fully connected layer is shown for $Enc$ (FCE), $Disc$ (FCD) and $Disc_z$ (FC1-FC3). Thus, FC1-4096 describes layer FC1 with a dimension of 4096. Further details of the architecture depicted in Fig. 2a are provided in Table I. In the second stage, the model shown in Fig. 2b referred as Encoder-LSTM network, is trained end-to-end in a SL setting. This model consists in three neural networks: $Enc_{\text{video}}$, $Enc_{\text{tool}}$ and $\Phi_{\text{LSTM}}$. $Enc_{\text{video}}$ and $Enc_{\text{tool}}$ process video frames $X_t$ and tool data $X_{t, \text{tool}}$, at each time instant $t$, respectively. $X_{t, \text{tool}}^{\text{tool}} = [x_t, y_t, z_t, s_t]$ describes the tool trajectory in 3D space $(x_t, y_t, z_t)$ and its grasping status $s_t$ ($s_t = 0$ if the grasper is closed, otherwise $s_t = 1$). $Enc_{\text{video}}$ has the same topology as $Enc$ in the UL model depicted in Fig. 2a. Nonetheless, some changes are introduced in the layers of $Enc_{\text{video}}$ while preserving the learned parameters from the UL stage. $Enc_{\text{tool}}$ is a fully connected neural network that maps tool data from a lower ($X_{t, \text{tool}} \in \mathbb{R}^4$) to a higher dimensional space ($Z_{t, \text{tool}} \in \mathbb{R}^{404}$). It avoids the use of a very small representation for the tool data vector (i.e. $Z_{t, \text{tool}} = X_{t, \text{tool}} \in \mathbb{R}^3$) with respect to that computed from video sequences ($Z_{t, \text{video}} \in \mathbb{R}^{4096}$). The size of each network’s layer, FC1-FC3, is indicated in Fig. 2b (i.e. FC1-16 describes layer FC1 with a dimension of 16). Table I details the design of each layer in $Enc_{\text{video}}$ and $Enc_{\text{tool}}$. The neural networks $Enc_{\text{video}}$ and $Enc_{\text{tool}}$ output the feature
Fig. 2: (a) In the UL stage, an encoder network $Enc$ is designed by optimizing a CAE in an adversarial framework. An input image $X$ is transformed into a latent space $Z$ by the encoder network $Enc$. The decoder network $Dec$ reconstructs the input image $X$ from $Z$, rendering $\tilde{X}$ as output. The discriminator networks $Disc_x$ and $Disc_z$ are applied on image data ($X$ and $\tilde{X}$) and latent space ($Z$), respectively. (b) Encoder-LSTM model used in the SL stage. The neural networks $Enc_{video}$ and $Enc_{tool}$ process video frames and tool data, respectively. Their outputs are used to create a feature vector $Z_t$ at every time instant $t$, which is modeled over $T$ time steps by an LSTM-CIFG network, $\Phi_{LSTM}$. (c) Design of the two-layer LSTM-CIFG network. Each layer has 256 blocks and processes the feature vectors $Z_t$ over $T = 64$ time steps. 50% of the outputs from the last cell are averaged. A fully connected layer (of dimension 6) with linear activation is used as the output layer. To prevent over-fitting, dropout is applied with probability $P$ during training at the output of each LSTM-CIFG cell.

D. Model Optimization

The optimization of the SSL model starts with the CAE model and ends with the Encoder-LSTM network.

1) CAE Optimization: In this stage, the CAE parameters are updated by processing unlabeled samples from the created dataset. In the following equations, the discriminators $Disc_x$ and $Disc_z$ shown in Fig. 2a are represented by $D_\phi$ and $D_\alpha$, respectively. Likewise, the CAE model, $X = Dec(Enc(X))$, and the encoding model for the latent space, $Z = Enc(X)$, are represented by the generator networks $G_\theta$ and $G_\alpha$, respectively. Given $M_v$ samples from a dataset (i.e. images), the discriminator network $D_\phi$ updates its parameters $\phi$ with the loss function defined in (1).

A reconstructed image $\tilde{X}$ is computed by the generator network $G_\theta$ as an encoding-decoding process. Therefore, $\tilde{X} = G_\theta(X) = Enc(Dec(X))$. $G_\theta$ updates its parameters $\theta$ with the loss function defined in (2).

The distribution of the latent space $Z$ is shaped to follow standard normal distribution $\mathcal{N}(0, 1)$ by using an adversarial framework. The discriminator $D_\alpha$ and generator $G_\alpha$ networks have parameters $\alpha$ and $\beta$, respectively. $D_\alpha$ and $G_\alpha$ are optimized with to the loss functions defined in (3) and (4), respectively. The $L1$-loss expressed in (5) was selected to
penalize the difference between the ground-truth $X$ and the reconstructed images $G_\theta(X)$. This loss function produces a lower blurring effect with respect to the $L2$-loss. Equation (6) is applied to the layers of the discriminator $D_\phi$. It measures the distance between the hidden representations produced in $D_\phi^{(l)}$ at a layer $l$, given as input ground-truth ($D_\phi(X)^{(l)}$) and reconstructed ($D_\phi(G_\theta(X))^{(l)}$) images.

$$
\mathcal{L}_{D_\phi}(\alpha) = -\frac{1}{M_c} \sum_i^{M_c} \left( \log(D_\phi(x_i)) + \log(1 - D_\phi(G_\theta(x_i))) \right) 
$$

$$
\mathcal{L}_{D_\phi}(\beta) = -\frac{1}{M_c} \sum_i^{M_c} \log(D_\phi(G_\theta(x_i))) 
$$

$$
\mathcal{L}_{L1}(X) = \|X - G_\theta(X)\|_1 
$$

$$
\mathcal{L}_{ACT}(X;l) = \|D_\phi(G_\theta(X))^{(l)} - D_\phi(G_\theta(X))^{(l)}\|_1 
$$

The total image reconstruction loss in (7), represents a linear combination of the loss functions (2), (4), (5) and (6), weighted by the scalars $\lambda_G$, $\lambda_G^2$, $\lambda_{L1}$ and $\lambda_{ACT}$, respectively.

$$
\mathcal{L}_R(X, \theta, \phi) = \lambda_G \mathcal{L}_G(\phi) + \lambda_{G^2} \mathcal{L}_{G^2}(\beta) + \lambda_{L1} \mathcal{L}_{L1}(X) + \lambda_{ACT} \mathcal{L}_{ACT}(X;l) 
$$

2) Encoder-LSTM Optimization: In this stage, the parameters of $Enc_{video}$ are initialized from the pre-trained encoder network $Enc$ of the CAE model. Therefore, during the optimization, the parameters of $Enc_{video}$ are fine-tuned while those of $Enc_{tool}$ and $\Phi_{LSTM}$ are optimized from scratch. Equation (8) describes the loss function used for the joint training of the $Enc_{video}$, $Enc_{tool}$ and $\Phi_{LSTM}$ networks, which define the Encoder-LSTM model (see Fig. 2b). It is a linear combination of the Root Mean Squared Error (RMSE) and Gradient Difference Loss (GDL) weighted by $\lambda_{RMSE}$ and $\lambda_{GDL}$, respectively. The RMSE defined in (9), penalizes the distance between ground-truth $F_t^{(i)}$ and estimated $\hat{F}_t^{(i)}$ force components at time $t$, indexed by $i = 0, ..., N - 1$, where $N$ is the total number of force components. On the other hand, the GDL defined in (10) measures the distance between the gradients of ground-truth and estimated force components, referred as $\partial F_t^{(i)}/\partial t$ and $\partial \hat{F}_t^{(i)}/\partial t$, respectively. These gradients are approximated by convolving the $i$-th force component $F_t^{(i)}$ with the kernel $h = [-1 0 1]$. Therefore, $\partial F_t^{(i)}/\partial t = F_t^{(i)} \ast h$ and $\partial \hat{F}_t^{(i)}/\partial t = \hat{F}_t^{(i)} \ast h$. Finally, in (9) and (10), samples are summed over a temporal window from $t = 0, ..., T_S$, being $T_S$ the total number of samples.

$$
\mathcal{L}(F, \hat{F}) = \lambda_{RMSE} \mathcal{L}_{RMSE}(F, \hat{F}) + \lambda_{GDL} \mathcal{L}_{GDL}(F, \hat{F}) 
$$

$$
\mathcal{L}_{RMSE}(F, \hat{F}) = \frac{T_S}{N} \sum_t \sum_i \left( F_t^{(i)} - \hat{F}_t^{(i)} \right)^2 
$$

$$
\mathcal{L}_{GDL}(F, \hat{F}) = \frac{T_S}{N} \sum_t \sum_i \left\| \frac{\partial F_t^{(i)}}{\partial t} - \frac{\partial \hat{F}_t^{(i)}}{\partial t} \right\|_1 
$$

III. EXPERIMENTS & RESULTS

The dataset consists of $\sim$780K samples split in 77% as the training set and 23% as the test set. Each sample is represented by an RGB frame, downsized from $200 \times 300$ to $100 \times 150$ pixels. In addition, every frame is provided with a 6D vector of ground-truth interaction forces and a 4D vector of tool data. This proportion of samples (unlabeled video sequences) are used in the UL stage. Afterward, in the SL stage, a subset of samples (video sequences, force and tool data) of size $\sim$320K and $\sim$40K are taken from the training and test sets, respectively. Relatively, these samples represent a proportion of 89% as the training set and 11% as the test set. However, with respect to the total size of the dataset ($\sim$780K samples), they represent a percentage of 41% and 5% as the training and test sets, respectively. The neural network models were implemented in Tensorflow [23] and the experiments were carried out using a single NVIDIA Titan X Graphic Processing Unit.

A. Convolutional Auto-Encoder: Image Reconstruction

The CAE was optimized over 241K iterations ($\sim$257 hours) with the Adam [24] solver, starting with a learning rate of $1 \times 10^{-4}$. In every iteration, two gradient descent updates were applied on the parameters of the generator.
Fig. 3: (a) A sample of reconstructed images by the CAE at test time. (b) Visualization of the feature maps computed by the encoder network at each layer (CE1, ..., CE5) during the supervised learning stage. The feature maps are shown for different input video frames (corresponding to cases A, B, and C) while the network is being optimized. Setup of the 6D force sensor and surgical tool used in the experiments. During the training (inference) stage, the Encoder-LSTM network processes (estimates) force and torque data measured with respect to $O_s = \{X_s, Y_s, Z_s\}$, which is the reference frame of the force sensor with respect to the world. The force sensor z-axis, $Z_s$, is aligned with the tool shaft.

<table>
<thead>
<tr>
<th>CASE†</th>
<th>ESTIMATED FORCE COMPONENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_x$</td>
</tr>
<tr>
<td>Pearson Correlation Coefficient (PCC)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0.1598</td>
</tr>
<tr>
<td>B</td>
<td>0.1978</td>
</tr>
<tr>
<td>C</td>
<td>0.2487</td>
</tr>
<tr>
<td>D</td>
<td>0.2294</td>
</tr>
<tr>
<td>Root Mean Squared Error (RMSE)</td>
<td></td>
</tr>
<tr>
<td>Units</td>
<td>N</td>
</tr>
<tr>
<td>A</td>
<td>0.0615</td>
</tr>
<tr>
<td>B</td>
<td>0.0553</td>
</tr>
<tr>
<td>C</td>
<td>0.0562</td>
</tr>
<tr>
<td>D</td>
<td>0.0630</td>
</tr>
</tbody>
</table>

†The models studied in cases A, B and C process raw, space-time and full-processed frames, respectively. The RMSE and GDL are considered in the loss function. In case D, the model takes as input the full-processed frames and only considers the RMSE in the loss function.

The estimated force signal quality (test set) measured by the Pearson Correlation Coefficient (PCC) and Root Mean Squared Error (RMSE), for each case (A-D), is presented in Table II. These results suggest that the Encoder-LSTM model has difficulties in learning from raw (case A) and space-time (case B) frames. This problem is alleviated by explicitly providing the Encoder-LSTM model information about the interaction between the surgical instrument and soft-tissues. Such information is emphasized in the model investigated in case C by using raw frames processed with mean normalization and space-time transformation. Nonetheless, other techniques with similar objective can potentially
work. For instance, the use of optical flow or an attention model. By inspecting some of the feature map activations in the encoder network during the training stage, as depicted in Fig. 3b, it is possible to understand how difficult it is to process raw and space-time frames. This illustration reveals that meaningful information, specifically motion due to tool-tissue interactions, is only propagated through the encoder network layers when this model is fed with full-processed frames (see bottom row in Fig. 3b). Regarding the loss function design, by comparing cases C and D, it is clear that using the GDL in the loss function provides advantages in the learning process. This result indicates that the RMSE and GDL ease the modeling of smooth and sharp details found in force signals, respectively.

It is important to notice that the interaction forces reported in the experiments are mainly transmitted along the surgical instrument shaft while performing pushing actions. During data acquisition, the z-axis of the force sensor was aligned with the surgical tool shaft (see Fig. 3c for reference). Therefore, in Table II, the PCC and RMSE values attributed to the estimated force $f_z$ are the most representative. The quality of this force component, which corresponds to case C, is shown in Fig. 4b by plotting the estimated vs the ground-truth data samples. For the same case, a sample of force signals computed over time by the Encoder-LSTM model is presented in Fig. 4a. In this illustration, the amplitude of the estimated force $f_z$ differs (at some points over time) with respect to ground-truth data. However, its shape is almost completely recovered. The rest of forces and torques have smaller values as a result of the pushing actions.

An insight from the proposed approach is that the LSTM-CIFG network is performing a time series estimation from a latent ($Z_t \in \mathbb{R}^{4160}$) to a force data space ($\hat{F}_t \in \mathbb{R}^6$). From that perspective, the initial samples estimated by this network should have a large error. However, Fig. 4a shows that such samples are close to the ground-truth data (i.e. see the force component $f_z$). This result can be explained by the initial state of the tool in the recorded dataset. That is, the tool is not in contact with soft-tissues, and therefore, the force close to zero.

The force sensing accuracy, usually measured with the RMSE, is reported to fall below 0.1 N, both in prototyped sensors [25] and in those developed under a vision-based approach [9]. In terms of this metric, the proposed model needs to be improved for real operational purposes (see the RMSE reported for $f_z$ corresponding to case C in Table II).

![Fig. 4: (a) Estimated force signals over time for a pushing action. The amplitude of these signals is shown in force (N) and torque units (Nm). (b) Estimated vs ground-truth force data (in Newtons) related to the $f_z$ component. The ideal line fitting the data samples (circles colored in blue) is shown in dotted style and black color. The best fitting line with a correlation coefficient of $R = 0.8084$, is depicted in solid style and magenta color.](image)

IV. CONCLUSIONS

In this work, a VBFS model designed in a SSL setting has been investigated. The results from the UL stage, suggest (qualitatively) that the CAE model designed in an adversarial framework, provides reconstructed images with sharp details (Fig. 3a). However, some difficulties appear in the learning process during the SL stage, when the encoder network of the CAE is serially connected with the LSTM-CIFG network. The encoder network, used as a feature vector extractor, has difficulties in finding a good representation of raw and space-time frames, useful for its processing by the LSTM-CIFG network. The best results are obtained when raw frames are processed with mean normalization and space-time transformation. Therefore, this suggests the importance of providing the neural network with information about the motion that results from the interaction between the surgical tool and soft-tissues. Additionally, in the SL stage, a loss function that considers the distance between ground-truth and estimated force (i.e. using the RMSE) is not enough to provide force estimates with good quality. By taking into account the distance between the gradients of ground-truth and estimated force (i.e using the GDL), the quality of force estimates is improved. As future work, three research
directions can be explored to improve the accuracy of the proposed approach. First, the use of depth information could help to reduce the gap between the amplitude of ground-truth and estimated force signals, i.e. using a method such as [26]. Second, an attention model [27] would allow to automatically process those image regions that contribute to the force. Finally, the proposed model is to be improved by interpreting its predictions with methods such as layer-wise relevance propagation [28], [29].

ACKNOWLEDGMENT

The first author of this work acknowledges the Mexican National Council for Science and Technology (CONACYT) and the Mexican Secretariat of Public Education (SEP) for their support in doctoral studies. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 779813 (SARAS). We thank the NVIDIA Corporation for supporting this research by donating a Titan X GPU card.

REFERENCES