Real-time Target Tracking in Fluoroscopy Imaging using Unet with Convolutional LSTM

by

Tengya Peng

Medical Physics Program
Duke University & Duke Kunshan University

Lei Ren, Advisor

Fang-Fang Yin

James Bowsher

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Medical Physics Program

2020
ABSTRACT

Real-time Target Tracking in Fluoroscopy Imaging using Unet with Convolutional LSTM

by

Tengya Peng

Medical Physics Program
Duke University & Duke Kunshan University

Lei Ren, Advisor

Fang-Fang Yin

James Bowsher

An abstract of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Medical Physics Program at Duke University and Duke Kunshan University

2020
Abstract

Target localization precision is crucial for the treatment outcome of radiation therapy. In lung stereotatic body radiation therapy (SBRT), verifying target motion in the real time 2D fluoro images is often used as a vital tool to ensure adequate coverage of the target volume before the treatment delivery starts. However, accurate target localization in 2D fluoroscopy images is very challenging due to the overlapping anatomical structures in the projection images. The localization is often visually performed by physicians and physicists, which is a subjective process that depends on the experience of the clinician. In this paper, we have developed a deep learning network for automatic target localization to improve the efficiency and robustness of the process. Specifically, the deep learning network adopts a Unet architecture with a coarse-to-fine structure. In addition, we innovatively incorporate convolutional Long Short-Term Memory (LSTM) layer into the network to utilize the time correlation between the fluoro images. A Generative Adversarial method was used to train the network to further improve its localization accuracy. A hybrid loss was used to improve the feature learning during the training. The model was tested on a large amount of data generated by the digital X-CAT phantom. Various patient sizes, respiratory amplitudes, and tumor sizes and locations were simulated in the X-CAT phantoms to test the accuracy and robustness of the method. Our model has been proved with great accuracy
not only on massive samples but also on specific set of samples. On massive samples, our model achieves IOU 0.92 and centroid of mass difference 0.16 and 0.07 cm in vertical and horizontal direction. On unique set of samples, the IOU is even higher to be 0.98. The centroid of mass difference could be amazingly 0.03 and 0.007 cm. In summary, our results demonstrated the feasibility of using this deep learning network for real target tracking in fluoro images, which will be crucial for target verification before or during lung SBRT treatments.
6. Discussion ................................................................. 28
7. Conclusion ................................................................. 30
References ................................................................. 31
1. Introduction

Lung cancer has been the most common cause of cancer death worldwide. Radiation therapy, especially stereotactic body radiation therapy (SBRT) is becoming a primary modality to treat early stage non-small cell lung cancer (NSCLC). In SBRT, target localization precision is crucial for the treatment outcome due to the high fractional dose, tight PTV margin, and sharp dose fall off outside PTV. Poor localization accuracy can lead to poor target coverage and unnecessarily high dose to the surrounding normal tissues.\[1\] Currently, the most commonly used imaging technique for target localization is cone-beam CT (CBCT), which can be acquired by an on-board imaging system that’s widely available on most radiotherapy machines. However, a major limitation of CBCT is that it doesn’t provide real time verification of the target motion, which is crucial for ensuring the adequate coverage of the tumor. To address this, real time 2D fluoro images are typically acquired after CBCT to verify the tumor motion. Physicists and physicians usually review the fluoro images to verify the target motion with reference to the PTV volume, which is a very challenging task due to the overlapping of anatomical structures in the 2D projection images. As a result, the verification process is subjective and very much dependent on the experience of the clinicians. In this study, we aim to address this bottleneck by developing a deep learning based method to automate the target localization in fluoro images to improve its accuracy and robustness.
Some related works have been explored to localize the lung tumor based on one single x-ray projection images. Zeng et al. used the deformation method to estimate the respiration motion from cone-beam projections based on a generic B-spline model.\[^3\] However, it involves large parameters and takes several hours to complete the computation. Xu et al. tried to use optical flow algorithm to find the average motion of tumors based on the contoured reference image.\[^4\] Their method estimates a rough region for the tumor respiration motion and then fine-tunes the localization by hand, which is subjective and slow. Some others try to use Principal component analysis (PCA) based motion modeling as a prior knowledge to predict the 3D tumor trajectory motion first and then track the tumors on 2D projection images. Q. Zhang et al. first developed a lung motion model based on principal component analysis (PCA) method.\[^5\] Based on this model, Li et al. acquired few eigenvectors and coefficients by applying PCA on the deformation vector fields (DVF) between a reference phase and other phases.\[^6\] Through varying the PCA coefficients, new DVFs could be generated and lead to new 3D volumetric images and 2D digitally reconstructed radiographs (DRRs). The DVFs could be constantly adjusted in the matching process between the acquired 2D DRRs and the x-ray projection images that we want to deal with. The 3D localization of the tumor could be derived by applying the corrected DVFs on the reference images. Y. Zhang further utilized a free-deformation registration process to improve the localization accuracy.\[^7\] These methods involve huge iterative optimization computing, which is not
ideal. Moreover, the results of these methods are poor because there are certainly intensity differences between the digitally reconstructed radiographs and acquired x-ray projections, which may lead to mismatching. Xu et al. introduced a linear model to approximate the nonlinear mapping to partly decrease the intensity mismatching. [8] A major limitation of this matching method based on PCA is that it can only localize the same patient in another set of projection images with prior knowledge to learn. Also, patient breathing changes from prior data to on-board images causing errors in the motion modeling, and consequently errors in the target localization.

Artificial intelligence or deep learning techniques have gained significant advances in the recent years partially due to the fast development of the computing power. Therefore, there have been some related works trying to adopt deep learning techniques for target localization. Wei et al. used a PCA method to build the lung motion model, and then used the neural network to learn and predict the PCA coefficients to derive the DVF. [9] As stated above, their method is also limited by the inaccuracy in the PCA modeling. Wang et al. used a seven layer Bi-LSTM network to predict the centroid coordinate of the tumor with a latency of 400 ms. [10] However, their method still needs labeling data of several initial cycles. Our aim is to localize tumors without any aid of labeling data. So far, no work has been published to continuously localize tumors with neural network directly using 2D fluoro projection images without any prior labels.
As we know, lung, ribs and hearts have different respiration motion rhythm. Their motion paths conform to periodic curves with different phases. Therefore, it is feasible to use neural network to distinguish tumor movement from other movements.

In the localization task, we aim to locate the tumors by distinguishing the tumor respiration motion from the motion of other structures in the projection images. A key problem is how to connect each frame of image together and capture the temporal information and spatial information. To achieve this, we propose a novel network whose backbone is residue Unet architecture with convolutional LSTM layers. To improve consistency performance, we adopted the hybrid objective and GAN method to train our network.

The contributions of this work are two-fold.

- We are the first to use purely a deep learning method to track tumor motions in the 2D fluoro projection images. We integrated many state of the art deep learning techniques to build and train the network. For example, we introduced residue refinement Unet block. We added convolutional LSTM layers in skip connection layers in Unet. We adopted the GAN method along with the hybrid loss we proposed to further improve the training of the network.

- A vast amount of digital phantoms were simulated using XCAT to evaluate the performance of our model for different patient sizes, breathing patterns, and tumor
sizes, locations, and motion patterns. Results demonstrated the feasibility of the proposed network for localizing tumors in the fluoro images.
2. Related Work and Inspiration Source

The task of localizing tumors in time-correlated fluoro images is comparable to the Video Object Segmentation (VOS) problem in the computer vision (CV) area. There are mainly four mainstream methods to do VOS task.

**One-shot method.** One-shot model segments frame by frame, treating the video segmentation problem as a series of 2D segmentation. One-shot model is usually used in face recognition problem since it’s fast with few parameters. However, there are temporal information in video besides spatial information. One-shot model misses the temporal information which sometimes could be essential.

**Propagation method.** Propagation based model resorts to previous frame mask for better segmentation mask in the next frame. In other words, the model combines the next frame and previous frame results together when segmenting the next frame. Propagation based model improves segmentation consistency when the object translates or rotates smoothly. However, the model’s performance can be affected when the object is partly shadowed or sheltered. We resort to propagation method in our model.

**Feature matching method.** Feature matching method utilizes Siamese network. Feature matching method first computes the features of each frame. It then calculates pixel-level feature matching between other frames and the reference frame, and then segment the current frame from the matching result. SiamRPN and SiamRPN++ achieve a good result and speed on video segmentation. Feature matching
based model gains more robustness when the object is partly shadowed or sheltered than propagation based model. Since the two methods are complementary to each other, RANet\textsuperscript{[14]} utilizes both the two methods. We have tried to use SiamRPN to track tumors in our task but failed. Because the overlaps of tumors and other tissues in 2D projections changes consistently with the tumor position during respiration. As a result, the constantly changing features cannot match well with the reference frame.

**Optical flow method.** Optical flow is used to guide the propagation process by converting the previous segmentation result to next frame segmentation prediction. Optical flow method estimates the pixel displacement between two frames and generates the displacement vector, which is then used to warp the current frame to generate the prediction of segmentation in the next frame. Flownet\textsuperscript{[15]} is the first model trying to get the optical flow using end-to-end neural network. Flownet2.0\textsuperscript{[16]} and PWCNet\textsuperscript{[17]} have some evolutions and improve the optical flow results. TecoGAN\textsuperscript{[18]} uses optical flow to warp previous result to give a rough prediction of current frame and feed together as input. However, optical flow fails to distinguish motionless rigid object. It is also ineffective if the pixel in the first frame disappeared in the next frame. This situation occurs when the object is partly obscured or overlapped. Our task is using the projection images where there are plenty of overlaps and occlusions, which makes it challenging to use optical flow to localize the target. Our studies also demonstrated the failure of such methods for our task.
3. Proposed Method

In this section, we would like to first provide an overview of our proposed model, describing the structure of generator and discriminator. In generator structure, we would illustrate the coarse-to-fine architecture. Then we would put emphasis on the convolutional LSTM layer. In the next step, we would show how we train our model using Generative adversarial method and the proposed hybrid loss.

3.1 Network Overview

Overall the model is trained using the Generative Adversarial Network (GAN) architecture, which includes two parts: the generator and discriminator.

Figure 1. Illustration of our proposed model.
3.1.1 Generator

Inspired by the propagation method, we want to combine the nearby frame information together for the current frame segmentation. Because we deal with sequence of N image, the input vector should be 4D in shape [Batch_size,N,H,W,C]. [H,W,C] is the dimension for a single image. N is the frame number, which in our experiment is 10. The projection image is gray scale so C here is unity. So, we combine every three frames nearby together as new channel in 4D vector for the sequence as Figure 2 shows. Now that the input vector has dimension of [Batch_size,10,H,W,3].

![Figure 2. Input and output of the generator.](image)

The generator is responsible for predicting the probability density map for the segmentation. Our generator, as shown in Figure 1, is a coarse-to-fine structure which has two Unets connected to each other via residue relationship. The Unet architecture
consists of a contracting path to capture context and a symmetric expanding path that enables precise localization.\cite{39} The contracting path extracts and down samples the features while expanding path restores the feature dimension which is so called up sampling. Unet is an encoder-decoder structure. The down path serves as encoder while the up path serves as decoder. There is a skip connection layer between the down sampling path and up sampling path, which provides local information to the global information while up sampling. In first Unet, we add a convolutional LSTM layer in every skip connection layer. In second Unet, we only add convolutional LSTM layer at the bottom. Like DCGAN\cite{20}, we use convolutions with stride 2 instead of max pooling to do down sampling. We use deconvolution instead of bilinear method to do up sampling, because max pooling or bilinear up sampling will cause problem of feature continuity and make it hard to train the GAN. We use leaky relu as our activation layer in generator and discriminator. Different from DCGAN, we still use sigmoid layer for the output in generator.

### 3.1.2 Discriminator

The input into the discriminator is the output from the generator or the ground truth. The discriminator will output the probability between 0 and 1 to tell whether the input is the ground truth or not. The discriminator can be viewed as the down path of the generator to encode and down sample features. But the discriminator has fewer layers, since generally discriminator should be weaker than the generator in GAN. At
the bottom, there is a fully connection layer behind the convolutional LSTM layer, along
with sigmoid layer to output the probability.

3.1.3 The coarse-to-fine Architecture

We found that the boundary result of using one Unet is not ideal. So, we adopted
the residual refinement module from BASNet.\textsuperscript{[21]} Two Unets constitute a coarse-to-fine
architecture. The first Unet is in charge of a coarse prediction while the second one is in
charge of the refinement by learning the residue between the coarse prediction from the
first Unet and the ground truth as

\[ S_{\text{refined}} = S_{\text{coarse}} + S_{\text{residual}}, \]

(1)

The coarse prediction means a rough segmentation result with blurry and noisy
boundaries or a unevenly predicted regional probabilities. This is somewhat similar to
the concept in Resnet as

\[ y = F(x; \{W_i, b_i\}) + x \]

(2)

where \( x \) and \( y \) are input and output of the layer, \( F(x; \{W_i, b_i\}) \) represents the residual
mapping to be learned, and \( F \) symbolize the convolution layer.\textsuperscript{[22]}

The refinement Unet is much smaller than the first Unet. There are only 5 layers
compared to 7 layers in first Unet. Also, the kernel size is half of that in the first Unet to
save more GPU memories and make the model smaller.
### 3.2 Convolutional LSTM

Long Short-Term Memory (LSTM) network has been demonstrated powerful to deal with sequence-to-sequence learning problems in Natural Language Processing (NLP). Information in the previous frames can be used to perform the present task. LSTM has long-term memories and learns to remember the essential part of past information and ignore the unnecessary part. LSTM is also found to be easier to train than other Recurrent Network (RNN) networks such as GRU, and it solves the problem of gradient vanishing to some extent.

Fully-connected LSTM is used to deal with sentences in NLP problems. In order to model well the spatial temporal relationships, Shi et al. extend the idea of FC-LSTM to Convolutional LSTM, which replace fully connection layers with convolution layers. The inner structure of convolutional LSTM module is shown in Figure 3.

![Figure 3. Inner structure of convolutional LSTM module.](image)

The major innovation of LSTM is that it proposed a cell state $c_t$ and a hidden state $h_t$ while using three controlling gates: input gate $i_t$, output gate $o_t$, and forget gate $f_t$. The figure shows the inner structure of the convolutional LSTM module.
to process the data flow. The cell state $c_t$ acts as an accumulator of the state information while $h_t$ represents the current output. The past state $c_{t-1}$ could be partly “forgotten” by the forget gates $f_t$ if they are on. Also, the input information could be accumulated in current cell state $c_t$ only if the input gates $i_t$ are activated. The input $x_t$ will be first processed by the convolution layer before being accumulated. The current output $h_t$ is decided by output gates $o_t$ and current cell state $c_t$. The key equations are shown below.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$C_t = f_t \ast C_{t-1} + i_t \ast \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$h_t = o_t \ast \tanh(C_t)$$

In above equations, $\sigma$ represents sigmoid activation function while tanh represents tanh activation function. $W \cdot x + b$ represents the convolution operation, and $[h_{t-1}, x_t]$ means concatenate the two vectors together.

Several convolutional LSTM layers could be stacked together like fully-connection LSTM does, but we only use single layer in our network. Moreover, we use Bi-LSTM structure, which deals with the images sequence in two directions: forward and backward, to fully take the sequence order into consideration. We have two Convolutional LSTM modules in our convolutional LSTM layer, with one feed in
original images sequence and the other one feed in the reversed sequence of images. The output of the Bi-LSTM is the concatenation of the output from the two Convolutional LSTM modules. The Bi-LSTM structure is shown in Figure 4.

3.3 Model Training

Segmentation task could be viewed as kind of image to image style translation. GAN has been verified effective to perform segmentation tasks.\textsuperscript{[24,25,26]} GAN can learn the mapping between two distributions.\textsuperscript{[27,28]} We can understand as: the objects in images that we want to segment is one distribution, and the segmentation ground truth is the other distribution.

The GAN involved the game theory: The generator is trained to give the segmentation results that cannot be distinguished from the ground truth masks by the discriminator, which is trained to do as well as possible at distinguishing the generator’s result from the ground truth. Actually, we are using w-GAN\textsuperscript{[29,30]}, which is a improved version of GAN. The objective of our generative adversarial training can be expressed as

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{bi-lstm.png}
\caption{Bi-LSTM}
\end{figure}
\[
V(D, G) = E_{x \sim P_{data}} [D(x)] - E_{x \sim P_{\hat{x}}} [D(x)] - \lambda E_{\hat{x} \sim P_{\hat{x}}} \left[ \left( \|\nabla_x D(\hat{x})\|_2 - 1 \right)^2 \right]
\]

(4)

where \( x \) is the image we want to segment. \( G \) represents the generator while \( D \) represents the discriminator. \( \lambda E_{\hat{x} \sim P_{\hat{x}}} \left[ \left( \|\nabla_x D(\hat{x})\|_2 - 1 \right)^2 \right] \) is the gradient penalty term which warrants the training of discriminator to converge. \( \hat{x} \) is the sampled examples from data distribution \( P_{\hat{x}} \). The training of GAN is a two-player game. The training of generator is to maximize the probability of discriminator making a mistake. The training of discriminator is to maximize the accuracy of making correct judgments. The generator and discriminator are trained alternatively. Here, we train a generator two times and discriminator once in an epoch. The discriminator is fixed while the generator is being trained, and vice versa. When training the generator, we feed the output from the generator into the discriminator and maximize the output probability. When training the discriminator, we maximize the output probability when we feed the ground truth, and minimize the probability when we feed the output from the generator.

Like BASNet, we use a hybrid loss. The hybrid loss is only for the generator, which gives the segmentation result. The hybrid loss could be shown below:

\[
L_{\text{hybrid}} = \lambda L_1 + L_{\text{ssim}} + L_{\text{iou}}.
\]

(5)
\( L_i \) is the L1 loss which takes account the output and ground truth difference and result in fewer noises. It is defined as

\[
L_i = \| y - G(x) \|. 
\]

\( (6) \)

where \( y \) is the ground truth and \( G(x) \) is the output from the generator.

\( L_{ssim} \) is the SSIM loss\(^{31}\) which is originally proposed for image quality assessment. Even though being simple and easily optimized, MSE (Mean Square Error) fails to capture perceived visual quality of the image. SSIM loss can learn the structural similarity that compares local patterns of pixel intensities that have been normalized for luminance and contrast. Thus, we integrate SSIM loss to learn the structural information from the ground truth. SSIM loss is defined as

\[
L_{ssim} = 1 - \frac{(2\mu_x \mu_y + 0.01^2)(2\sigma_{xy} + 0.03^2)}{(\mu_x^2 + \mu_y^2 + 0.01^2)(\sigma_x^2 + \sigma_y^2 + 0.03^2)}
\]

\( (7) \)

where \( \mu_x, \mu_y, \sigma_x, \sigma_y \) are the mean and standard deviation of pixel values in image \( x \) and \( y \) separately, and \( \sigma_{xy} \) is their covariance.

\( L_{iou} \) is the IOU loss\(^{32}\). IOU is first used to measure the overlapped proportion of two sets. Then it is used widely as evaluation metric for segmentation and object detection. Here, we integrate it directly into our hybrid loss as
where \( y(r,c) \in \{0,1\} \) is the ground truth label and \( S(r,c) \) is the segmentation result from the generator.

L1 loss learns the background and prevents the noise. SSIM loss learns the similarity from a path-level, which consider he local neighborhood around each pixel. SSIM especially focus on the boundary similarity. IOU is a map-level measure which gives more focus on foreground. And, the most important, the GAN loss learns from a global view, not from the whole image but from the whole images sequence. So, our model learns tumors in single x-ray projection images sequence from the structure, how it looks like after projection and overlapped with ribs or heart, and the movement.

Summing up both the generative adversarial loss and hybrid loss, the total objective is

\[
L_{\text{total}} = \arg \min_G \max_D \left[ \max D \left( G, D \right) \right] + \lambda L_1 + L_{\text{ssim}} + L_{\text{iou}}
\]  

(9)

where \( \lambda \) balances the training speed of each part and here we give it value of 100. And we use Adam optimizer to optimize our total objective.
4. Experiments and Setup

The sizes and positions impact the tracking because of tumors in different sizes and locations have different overlaps thus having different features. It would be very challenging if the tumor is too small especially when the tumor overlaps with tibs in the projection image. The respiration amplitude also play an important part in tracking.

Our proposed method is validated and verified by X-CAT phantom. X-CAT phantom could be used to generate many samples to simulate different patient scenarios. We conduct our experiments from two aspects. The first is that we train and test our model on massive patient scenarios in X-CAT, which we call it group-based model. Each scenario represents a unique patient size, tumor position and size, and respiration amplitude. The other experiment is that we train and test our model to achieve the optimal performance on a specific patient, which we call it patient-specific model. The model is trained based on patient-specific data to optimize its accuracy for the patient.

Several metrics are used to evaluate the result. The IOU and dice coefficient tell the proportion of how much our results of tracking region match with the ground truth. The centroid difference tells the distance deviation of the centroid between the results and ground truth.

4.1 X-CAT Phantom

In XCAT phantoms, we can adjust the long and short axis scale to simulate the height and width of patient. We can place the sphere tumors anywhere we want in lung.
We could also adjust the diameters of tumors. We could also modify the extent of diaphragm and anterior-posterior (AP) extension of chest to simulate different respiration motion amplitudes in anterior-posterior (AP) and superior-inferior (SI) direction. The AP extension parameter controls the anteroposterior lung and rib motions. The diaphragm extension or SI extension is more crucial in 2D projection. It controls the vertical lung motion. Moreover, the tumor and body motion is separate in X-CAT phantom. We can adjust the tumor SI amplitude independently.

The X-CAT phantom provides a 3D body model. Digital reconstructed radiograph (DRR) algorithm is used to simulate X-ray projection. DRR programs can adjust the projection angle and the projection resolution. The X-CAT phantom separates the respiration process into 10 phases, each collected in 0.5s. Therefore, we have a set of 10 projection images representing one respiration cycle.

4.2 Group Based Model

As described above, we train and test our network on massive samples of different body sizes, tumor sizes, positions, and respiration amplitudes. We want to validate our model’s generation ability. There are 110 samples, created by XCAT phantom, as our training dataset to train the model. These samples are highly representative in tumor sizes and locations. The diameter of the tumors is in range of 20 mm to 40 mm. The tumors could be in the left or right lung, upper, middle, and lower part of lung. The tumors could have overlaps with ribs and the heart. The Ap (anterior-
posterior) and SI (superior-inferior) amplitudes are randomized in range of 0.8 cm to 2 cm and 1.6 cm to 4 cm, which represents common patients’ respiration motion amplitudes. We test the trained model on another 30 samples as our testing dataset. In these testing samples, the tumors could be smaller or bigger, whose sizes are not in the training samples. And the positions of tumors could be more challenging.

4.3 Sub-group Model

In this section, we want to testify if model trained by different patients could be adopted and finetuned (transferred) to specific patient. We have 15 sets of samples. Each set represent a specific patient. In each set, the tumor position, size and body size should be the same. Each set contains 20 cases of samples of different respiration amplitude. Theses samples represent data from the same patient acquired many times, which might have different respiration amplitudes. The tumor respiration amplitude is from 1,1.1,1.2,13…..2.9 times of the body amplitude. First, we train and test only on these 15 sets of samples. We use each set’s 10 former cases to train the model and test on the 10 latter cases. We want to simulate the situation that we use patients’ data of small motion amplitudes to train the model but test on the patients’ data of large motion amplitudes. Second, we use the pretrained model in experiment to finetune on these 15 sets of samples. 10 former cases are used to finetune and 10 latter cases are used as testing dataset.
There are various ways to finetune by choosing different layers in the generator to fix. We can fix the former fewer layers in the downpath and few final layers of the upgoing layers in Unet structure. Or we can fix the bottom of the Unet. By comparison, we choose to fix the bottom layers in Unet structure.

**4.4 Patient Specific Model**

As described above, in this section, our model is trained and tested only on a specific set of samples, which represents a unique patient. We expand the samples in the set to 100 cases whose tumor respiration amplitude is from 1, 1.1, 1.2… to 10.9 times of the body respiration amplitude. We only use the set’s 40 former and 40 latter cases to train the model, and test on the remained cases. In this situation, we train the model with cases of small amplitudes and large amplitudes but test on cases of medium amplitudes. We have done this experiment on different sets of samples having different tumor positions. This experiment aims to verify our model’s ability to be used on single specific patient. A patient could come to hospital and thus have several sets of CT or projection images. We can train our model on the prior and test our model on new data.
5. Results

5.1 Group Model

The performance is state of the art. The mean IOU for the testing dataset is 0.92 while the dice coefficient is 0.96. We also calculate the centroid of tumor mass difference. The mean centroid of mass difference is 0.16 cm and 0.07 cm in vertical and horizontal direction. Figure 5 presents three specific samples from our testing dataset. We can only present their first frames. The left side figure presents the fluoro image sequence that we want to track while the right side figure presents the tracking results. The tumors are tracked and painted in red. The upper case represents a very tricky situation when the tumor is overlapped and blurred by the heart. The middle case represents situation when tumor is small and might overlap with ribs. The lower case represents easy situation when tumor is big enough to distinguish from surrounding tissues. In the upper case, the IOU is 0.92 while dice coefficient is 0.96. The centroid of mass difference is 0.17 cm and 0.11 cm in vertical and horizontal direction. For the middle case, the IOU is 0.91 while the dice coefficient is 0.95. The centroid of mass difference is 0.15 cm and 0.06 cm in vertical and horizontal direction. For the lower case, the IOU is 0.95 while dice coefficient is 0.97. The centroid of mass difference is 0.09 cm and 0.05 cm in vertical and horizontal direction.

We can conclude that smaller sizes of tumors are harder to track with high accuracy. It is harder to find the boundaries when the tumors are totally overlapped by
the heart in the projection image. When tumors are partly overlapped by ribs, the accuracy also falls. But the result is still above 0.8 which is still a high level, because our model tracks the tumors by their motion and movements. It becomes easier when the tumors are large enough and parts that are not overlapped have larger proportion.

Figure 5. Specific cases: (a) tumors overlapped by heart (b) tumors partly overlapped by ribs. (c) tumors are large enough to identify
5.2 Sub-group Model

First, we train and test only on 15 sets of samples. Then we use the pretrained model by massive samples to finetune on these 15 sets of samples. The results are compared to verify if finetuning works. We choose an easy set of samples, which has relatively large tumors in middle part of the lung without many overlaps, and a relative hard set of samples in which tumors are partly overlapped with heart. They are shown in Figure 6. The comparison result is shown in Table 1.

![Figure 6. Two Sets of sample in finetuning experiment: (a) Easy set (b) Hard set](image)

<table>
<thead>
<tr>
<th>No.</th>
<th>Without finetuning</th>
<th>Finetuning on pretrained model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IOU</td>
<td>Dice coefficient</td>
</tr>
<tr>
<td>1</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.93</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 1: Comparison of finetuning or not
We also tried messing up the samples in each set, training on samples of random amplitude and test model on the remained samples. The results are much better, because the above situation is the most challenging one.

### 5.3 Patient Specific Model

Our model is trained and tested only on a set of samples, representing a specific patient. The set of samples we choose is shown in Figure 7. The tumor is relatively big with 30 mm diameter in right upper part of lung. The result is shown in Table 2.

![Figure 7. Specific set of samples](image)

<table>
<thead>
<tr>
<th>Set No.</th>
<th>Mean IoU</th>
<th>Mean Dice coefficient</th>
<th>Centroid difference\pixel</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>0.99</td>
<td>0.98</td>
<td>[0.04,0.01]</td>
</tr>
<tr>
<td>42</td>
<td>0.99</td>
<td>0.98</td>
<td>[0.04,0.01]</td>
</tr>
<tr>
<td>43</td>
<td>0.98</td>
<td>0.99</td>
<td>[0.03,0.01]</td>
</tr>
<tr>
<td>44</td>
<td>0.98</td>
<td>0.99</td>
<td>[0.03,0.01]</td>
</tr>
<tr>
<td>45</td>
<td>0.98</td>
<td>0.99</td>
<td>[0.02,0.01]</td>
</tr>
<tr>
<td>46</td>
<td>0.98</td>
<td>0.99</td>
<td>[0.02,0.01]</td>
</tr>
<tr>
<td>47</td>
<td>0.98</td>
<td>0.99</td>
<td>[0.02,0.01]</td>
</tr>
<tr>
<td>48</td>
<td>0.98</td>
<td>0.99</td>
<td>[0.02,0.01]</td>
</tr>
</tbody>
</table>
The mean IOU achieves 0.98. The centroid of mass difference achieves 0.03 cm and 0.007 cm in vertical and horizontal direction. So, our model can work very effectively on specific patient. But in this case, our model can only learn the selected patient’s case and fail to work on other patients. When tracking another patient, there appears overfitting problems as shown in Figure 8.

![Figure 8. Fails to predict on another new patient](image)

We have also plot the vertical coordinate of tumor centroid, as shown in Figure 9. It conforms to cosine wave. It is seen that the tumor movement is predicted with minor deviation in tiny centroid of mass difference.
Figure 9. Plot of vertical coordinate of tumor centroid
6. Discussion

Our model works well on massive samples and specific set of samples. Because we have adopted several latest deep learning tricks. We successfully utilize the temporal information in projection image sequence by adding the convolutional LSTM layers and combing nearby frames together. It has been verified that fully-connection LSTM have great performance on 1D sequence. It has been tested to predict the coordinate of centroid of tumors having a time lag. We have proved that the tumor respiration motion conforms nearly to cosine function in Figure 7. So, we expand this tracking problem to 2D sequence by using convolutional LSTM. The coarse-to-fine architecture greatly improves the boundary and centroid of mass difference accuracy. The generative adversarial training method along with hybrid loss enable our model easier to train, and improve the robustness. These factors make our model stand out from other precious works.

Finetuning from pretrained model gives worse outcome as shown in Table 1. Because there are many other samples of different tumor locations to give more distractions to the sets of samples we want to test on. Hence, we try to train and test on specific set of samples. The performance becomes nearly perfect because the tumor size is fixed and locations are in a rough region to some extent. From this point, we see that position of tumors really play an important role. Different tumor position might have different overlaps with other tissues, resulting in different features to be learned. For
example, if there are no samples whose tumor is totally overlapped with the heart in training dataset, the model can’t work on this sample because the features of tumors totally overlapped with the heart have not be learned. But we have tried to put samples whose tumor is of larger size and overlapped totally with the heart into the training dataset, and test the model on samples having much smaller sizes of tumors while other conditions remained unchanged. The model still works out. Similarly, training on samples of smaller sizes of tumors but testing on samples of larger sizes of tumors that are not in training set still works well. Our model has the ability to deduce from harder cases to easier cases. Our model is thought to have the ability to work on samples that are somehow similar to the samples in training dataset.

The limitation of our model is obvious. The training of our model needs massive data to learn various features. That is the main reason why we have not tested our model on clinical data because we do not have huge clinical database. We still need more clinical data to verify if our model can work in clinic. Tracking tumors without any aids only on single x-ray projection images is still very challenging. But we have make our first step and our model might be a milestone in this exploration process.
7. Conclusion

In this work, we proposed a neural network model that can track tumors on fluoro images of 10 phases according to their motions and movements. Our network combine the most advanced technique such as Convolutional LSTM, fine-to-coarse structure, hybrid loss, and GAN method. Tumors motion conforms to cosine curve. Our model certainly can learn this tumor motion pattern. Our model works pretty well on X-CAT phantom, no matter on massive samples or on specific set of samples. Thus, we validate the feasibility to track tumors on single x-ray projection images of 10 phases, using massive patients’ data or a specific patients’ data. Our model shows great potential in tracking tumors in treatment delivery in the future.
References


