Physica Medica xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Physica Medica

journal homepage: http://www.physicamedica.com

Original paper

Deep architecture neural network-based real-time image processing for image-guided radiotherapy

Shinichiro Mori

Research Center for Charged Particle Therapy, National Institute of Radiological Sciences, Inage-ku, Chiba 263-8555, Japan.

ARTICLE INFO

Article history:
Received 5 March 2017
Received in Revised form 19 June 2017
Accepted 13 July 2017
Available online xxxx

Keywords:
Radiation therapy
Fluoroscopy
Neural network model
Image processing
Computer-assisted

ABSTRACT

Introduction: To develop real-time image processing for image-guided radiotherapy, we evaluated several neural network models for use with different imaging modalities, including X-ray fluoroscopic image denoising.

Methods & materials: Setup images of prostate cancer patients were acquired with two oblique X-ray fluoroscopic units. Two types of residual network were designed: a convolutional autoencoder (rCAE) and a convolutional neural network (rCNN). We changed the convolutional kernel size and number of convolutional layers for both networks, and the number of pooling and upsampling layers for rCAE. The ground-truth image was applied to the contrast-limited adaptive histogram equalization (CLAHE) method of image processing. Network models were trained to keep the quality of the output image close to that of the ground-truth image from the input image without image processing. For image denoising evaluation, noisy input images were used for the training.

Results: More than 6 convolutional layers with convolutional kernels >5 $\times$ 5 improved image quality. However, this did not allow real-time imaging. After applying a pair of pooling and upsampling layers to both networks, rCAEs with >3 convolutions each and rCNNs with >12 convolutions with a pair of pooling and upsampling layers achieved real-time processing at 30 frames per second (fps) with acceptable image quality.

Conclusions: Use of our suggested network achieved real-time image processing for contrast enhancement and image denoising by the use of a conventional modern personal computer.

1. Introduction

The latest image-guided radiotherapy algorithms have improved positional accuracy by using medical imaging techniques such as fluoroscopy [1,2]. Patient positional verification is routinely performed by 2D-3D image registration using X-ray images and reference digitally reconstructed radiography images before irradiation [3–5].

Tumour or implanted fiducial marker position can be detected in real time by X-ray fluoroscopic imaging during treatment [6–8], solving the problem of the inconsistency of the relationship between patient surface and internal tumour motion [9]. Although the deliberately high contrast of fiducial markers makes them easy to detect on fluoroscopic images, implantation is invasive. A second approach is to detect tumour motion directly without fiducial markers (markerless tracking technique) [10–14]. This technique detects tumour position using fluoroscopic images, but tumour detection accuracy is strongly affected by image quality. Several types of image processing (e.g., dynamic range compression, denoising, contrast enhancement, frequency modulation) have already been integrated into medical imaging modalities, but obtaining good image quality may require that computation time be increased to an unrealistic degree. Commercial imaging systems have achieved real-time image processing using specialized hardware, such as a graphics processing unit (GPU) or field programmable gate-array (FPGA), but both GPU and FPGA require more advanced programing skills and longer development time than CPU-based programming.

Deep learning has improved performance in artificial intelligence more than conventional machine learning. It is employed in everyday technologies such as search engines and speech recognition. In medical imaging, deep learning has been used for image denoising [15,16] and bone-density suppression [17]. Several deep learning frameworks are now publicly available and do not require specialised programing skills or extended development time. Once a network model is trained, computation time is rapidly decreased by the utilization of GPU.

E-mail address: mori.shinichiro@qst.go.jp

http://dx.doi.org/10.1016/j.ejmp.2017.07.013
1120-1797/

In this study, we evaluated several types of network model to achieve real-time image processing. Various types of image processing are needed for different purposes; therefore we selected Contrast-limited Adaptive Histogram Equalization (CLAHE) [18] and extended it to add image denoising function, because it allows changes in pixel values to improve image contrast by analysing pixel information in large areas rather than pixel by pixel. Our goal was for the deep learning network model to be able to perform CLAHE in real time, allowing it to be extended to other image processing algorithms.

2. Materials and methods

2.1. Image preparation

A total of 430 X-ray images were acquired in 42 prostate cancer patients during carbon-ion scanning beam treatment in our hospital. The images were acquired by two oblique X-ray fluoroscopic units used in patient setup procedures [19]. More than two pairs of X-ray images were acquired for each treatment fraction. The X-ray imaging system is an indirect type of dynamic flat panel detector (DFPD) (PaxScan 3030 + ©, Varian Medical Systems, Palo Alto, CA) with an image area size of 296 × 296 mm, pixel size of 388 μm, and dynamic range of 14 bits. Image matrix size was 768 × 768. A DFPD is installed on either side of the vertical irradiation port at 35 and 325 degrees, respectively, with the X-ray tubes installed under the floor. Distances from the room isocenter and source-image receptor distance are 1690 mm and 2390 mm, respectively. Imaging X-ray conditions are 100 kV tube voltage, 100 mA tube current, and 4 ms of radiation exposure per image.

Two types of input image were prepared; the first was a mean of ten fluoroscopic image frames to reduce the magnitude of image noise for image processing performance evaluation, and the second was a single frame image (noisy image) for image processing with image denoising evaluation. The ground-truth image was the first input image applied to CLAHE with the number of histogram bins used for histogram equalization of 256. CLAHE performs histogram equalization in pixel values within a whole image or image regions. Output small image regions were combined using image interpolation. By doing this, CLAHE improved image contrast. CLAHE was performed using a commercial programming environment (MATLAB R2016a©, Mathworks, Natick MA, USA).

2.2. Network architecture

Neural network architectures such as autoencoder (AE) [20–23] and convolutional neural network (CNN) [24–26] were often used for medical image processing. In the early days, these networks were shallow due to technical limitations. However, current CNN and AE have achieved better performance by applying deep architecture (details were described in the next section). We, therefore, designed two types of network structure to extend to both image denoising and CLAHE processing; a residual convolutional autoencoder (rCAE) and a residual convolutional neural network (rCNN). These network structures were involved in convolution, batch normalisation (BN), rectified linear units (ReLU), pooling, and upsampling layers. The network template was defined to express the respective network structures clearly as shown in Fig. 1a, and the following five parameters were defined: the parameter k is the convolution kernel size, m and p are the number of convolutional layers before the first pooling layer and after the last upsampling layer, respectively. The parameter n is the number of convolutional layers between pooling and upsampling layers. The parameter q is the number of pairs of pooling-upsampling layers.

Pixels of zero value were inserted into the input image boundary region before generating convolution to keep output image size and input image size to be the same. A default number of the convolutional filter was 64, and the convolutional filter size was multiplied by a factor of 2 and 0.5 after pooling and upsampling layers, respectively. The convolutional stride was 1. The number of the convolutional filter size at the last layer was 1. A pooling layer using max-pooling was selected, with kernel and stride sizes 2 × 2 pixels and 2 pixels, respectively. The upsampling layer selected used a bilinear weight filter and had a kernel size of 4 × 4 pixels and stride of 2 pixels.

2.2.1. Residual convolutional autoencoder (rCAE)

The autoencoder learns a low-dimensional representation (compressed approximation) of the input image by encoding and decoding through the network [20–23]. To improve autoencoder performance, a deeper autoencoder was introduced [15]. While a convolutional layer is better for image processing to generate an image feature map than the inner product layer used in an autoencoder, CAE was introduced by replacing the inner product layer with a convolutional layer [16].

The basic structure of the CAE in this study placed in order as convolution plus pooling and upsampling plus convolution, and added input data. By doing this, a CAE with residual net was constructed. Eight types of rCAE was designed by changing the number of pooling-upsampling layers (the range of tested parameters $k = 3$, $m = 2, 3$, $n = 2, 3$, $p = 1, 2$, $q = 1–3$) (Nos. 1, 2, and 5), and the number of convolutional layers (the range of tested parameters $k = 3$, $m = 1–6$, $n = 1–6$, $p = 1–5$, $q = 1$) (Nos. 3–8). Convolutional kernel size in rCAE was 3 × 3 pixels. For example, an rCAE with two pairs of pooling-upsampling layers and three convolutions is shown in Fig. 1b, and can be expressed by the parameter ($k = 3$, $m = 3$, $n = 3$, $p = 2$, $q = 2$).

2.2.2. Residual convolutional neural network (rCNN)

The rCNN was involved in one or multiple sets of convolution, BN, and ReLU layers, the last layer being a convolutional layer with a single feature map and the input image being added after the last layer. We changed the convolutional kernel size and the number of convolutional layers; thus, a total of nine networks were designed (the range of tested parameters $k = 3, 5, 7$, $m = 3, 6, 9$, $n = 0$, $p = 0$, $q = 0$) (Nos. 9–17). In one example, three sets of convolutions with the kernel size of 7 × 7 pixels, BN, and ReLU layers are shown in Fig. 1c; this can be described by the parameters ($k = 7$, $m = 3$, $n = 0$, $p = 0$, $q = 0$).

In order to shrink input data size, pooling layer was set after importing input data, and then multiple convolutional layers were set, and after or between them, an upsampling layer was set. We changed the number of convolutional layers, all with a kernel size of 3 × 3 pixels (the range of tested parameters $k = 3$, $m = 0$, $n = 3, 6, 9, 12$, $p = 0, 3, 5$, $q = 1$) (Nos. 18–23). One example of this included a pooling, six convolutions, upsampling, three convolutions (Fig. 1d). This network can be expressed by the parameters ($k = 3$, $m = 0$, $n = 5$, $p = 2$, $q = 1$).

2.3. Network training

The input image and ground-truth image sets were normalized to the range of 0–1, except the irradiation port cover edge region shown as a white straight line on the image (marked as arrow in Fig. 3a). Due to the limited GPU memory, all images were resized to 384 × 384 pixels using MATLAB before importing to the network. Network models were trained to be close to the quality of the output image to that of the ground-truth image from the input image. All images were rotated 90 and 180 degrees to increase the number of training images (data argumentation), and not applied image flip because of symmetric structures in pelvic region. A total of 3544 image pairs (input images and ground-truth images) were used