Presence Detection of Surgical Tool Via Densely Connected Convolutional Networks

Xiao-guang LIN¹,²,⁴, Yu-wen CHEN¹,²,⁴, Bao-lian Qi²,⁴, Peng WANG³,* and Kun-hua ZHONG¹,²,⁴,*

¹Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China
²Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu, China
³The First Hospital Affiliated to AMU (Southwest Hospital), Chongqing, China
⁴University of Chinese Academy of Sciences, Beijing, China

*Corresponding author

Keywords: Surgical tool, DenseNet, Presence detection.

Abstract. Surgical tool detection is important to surgical workflow recognition. It is considered as an essential task in surgical phase recognition. Recently, Densely Connected Convolutional Networks have gained a huge success in computer vision applications, especially in object detection and image classification. In this paper, we proposed a method to solve the surgical tool presence detection problem as a multi-label classification problem based on Densely Connected Convolutional Networks. The performance of the proposed method has been evaluated in the surgical tool presence detection challenge dataset held by Modeling and Monitoring of Computer Assisted Interventions workshop. The result shows that our proposed model has achieved significant success in detecting surgical tool and got a mean average precision of 62.9% on the testing data. The technology studied in this paper has broad application prospects in computer-aided surgical systems and is a core component of the artificial intelligence medical operating room in the future.

Introduction

More than half a million surgeries are performed every day worldwide [1], which makes surgery one of the most important component of global health care. Competing demands are motivating a better understanding of surgical processes: surgical procedures are getting more complex [2], residents now have to be trained while performing less procedures [3], the surgical interventions have to be more and more justified [4] and the procedures have to cost less money[5]. A better understanding of surgical practices is the key component to addressing these issues. Automatic recognition of surgical workflow is a core unresolved problem of computer assisted intervention (CAI)[6]. Various applications e.g. context aware surgical systems, staff assignment, automated guidance during intervention, surgical alert systems etc. can benefit from fully automatic recognition of surgical phases. It could also be used after the surgery, to automatically generate reports on the procedure, allowing for a better understanding of the interventions and eventually saving a lot of time for the medical staff. Among the recent studies for surgical phase recognition, many studies [7] use surgical triplets [8] (the utilized tools, the anatomical structure, and the surgical actions) of each frame in the videos to represent each time in surgery. Extracting this feature leads to the surgical tool presence detection problem to detect what tools are used at each time in surgery. Presence detection of tool can be helpful to respond to several limitations of minimally invasive surgery and to assist surgeons. During a medical intervention it could provide the surgeon with important information help prevent unsafe situations or even injuries.

Different from traditional surgical tool detection [10] problems, surgical tools presence detection does not require the awareness of the locations of surgical tools. Instead, surgical tool presence detection only detects what kinds of surgical tools are used in each time step during surgery. This
detection problem can be viewed as an image classification problem while it has several difficulties than the traditional classification problem. For instance, it is probable that multiple tools are used at the same time during the surgical workflow and the frequency of use of various surgical tools is very different, so this is imbalanced problem, which requires higher generalization ability for the model. It is almost impossible to manually design feature for tool presence detection. That is why more powerful classification method is necessary to overcome such difficulties.

In recent years, Convolutional Neural Networks (CNNs)[11] are the current trend in machine learning and computer vision tasks. One of the tough problems in general machine learning and computer vision is feature extraction. Traditional feature extraction requires expert knowledge about the data and the methodology differs in each task. We can automatically extract more complicated and useful features by training CNNs. It has beaten other machine learning methods in many areas especially when the data set is large enough. Supervised learning is the most common form of machine learning which deep learning improves the state-of-the-art of most problems. Especially Densely Connected Convolutional Networks [12] (DenseNet) architecture. Compared to a standard CNN architecture, each layer within DenseNet architecture concatenates all preceding layer feature-maps as input. Such architecture can alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. Fig.1. illustrates this concept, where arrows indicate reused feature-maps from previous layers in a five-layer dense block.

In this paper, we propose a Densely Connected Convolutional Networks based multi-label classification method for surgical tool presence detection in laparoscopic videos. We show that a good detection result can be obtained only use this network. Experiments were performed on a dataset M2CAI 2016 tool challenge contains 15 videos of cholecystectomy procedures from University Hospital of Strasbourg/IRCAD (Strasbourg, France). The remaining sections are organized as follows. In Section II, we present the overall method for surgical phase and tool detection. Section III describes our proposed method. Section IV presents the experiment and result. Finally, we conclude our work and discuss our future work in Section V.

Related Work

Within this field, the development of new methods for analyzing surgical procedures is an important issue. Many researches have been conducted for developing methods for recognition of surgical workflow. Bardram et al. [13] proposed a system using embedded and body-worn sensor data to train a decision tree in order to predict surgical phases. They studied sensor significance in order to identity the most important features for surgical phase prediction. Stauder et al. [14] used Random Forest (i.e., a bag of decision trees) to predict surgical phases from sensors measurement. Other models like Hidden Markov Model (HMM) were also considered by Padoy et al. [15,16] for online recognition of surgical steps. Bouarfa et al. [17] used HMM with a pre-processing on the input sensor data in order to improve the detection of high-level surgical tasks. SVM classifier was also considered by Lalys et al.[18] to detect phases and low-level surgical tasks using cameras in pituitary surgery. Varadarajan et al. [19] used HMM to recognize and segment surgical gestures for surgical assessment and training. Learning the topology of an HMM is however still challenging and improving this step continues to be investigated [20].

Early efforts focusing on surgical tool detection made sometimes use of passive [21] or even active [22] markers on the tools. But these are not convenient for use in real surgeries and are mostly designed for training and skills evaluation on test-bench. [23] based on color segmentation and [24] based on a supervised classification method, do not require to alter the tools and are more suited for real procedures. In recent years, CNN has gained a huge success in computer vision applications, especially in object detection. CNN is a powerful tool that can retrieve hierarchical vision of features. Twinanda et al [9] based on CNN architecture presented an EndoNet, which have got good results in surgical phase detection. Sahu [25] proposed a transfer learning method for generating features for
surgical tools and phase recognition from the ImageNet classification features [11], which also got good results. Recently, GAO Huang [12] proposed a new convolutional network architecture, which we refer to as Dense Convolutional Network (DenseNet). It introduces direct connections between any two layers with the same feature-map size. DenseNets scale naturally to hundreds of layers, while exhibiting no optimization difficulties. Under multiple settings, it achieved state-of-the-art results across several highly competitive datasets. Moreover, DenseNets require substantially fewer parameters and less computation to achieve state-of-the-art performances. Given the recent success of these networks, in this paper we propose a Densely Connected Convolutional Networks based multi-label classification method for surgical tool presence detection in laparoscopic videos.

Proposed Method

We consider these problem as multi-label classification problems [26]. Traditional multi-class classification is the problem of classifying instances into one of the more than two classes, and each instance belongs to only one class. Different from multi-class classification, multi-label classification allows each instance to belong to one or more than one classes. Multi-label classification is a generalization to multi-class classification. In real-world problems, multi-label classification tasks are ubiquitous. For instance, in text categorization, each document can belong to more than one predefined topics, such as sport and health. Thus, the surgical tool presence detection problem can also be viewed as a multi-label classification problem. It is because that each image which we extract as image frames from the surgery videos may contain one or more than one surgical tools. Thus, each image can belong to one or more than one classes. In this way, we can use multi-label classification methods for surgical tool presence detection. The two common methods for multi-label classification are problem transformation and algorithm adaption. Problem transformation decomposes the multi-label classification problem into multiple independent binary classification problems. Algorithm adaptation methods [27] design or adapt algorithms to solve multi-label classification directly. In the proposed method, we use problem transformation method to convert the multi-label classification problem into several independent binary classification problems. Each of the binary classifiers is to detect if one kind of the tools is used in the images. Thus, we use Densely Connected Convolutional Networks to classify these images. In this section, we explain the detail of the model, loss function, our evaluation metric and optimization method used for network training.

DenseNet201 Model

Convolutional Neural Networks (CNNs) are used in all of the state-of-the-art vision tasks such as image classification, object detection and localization, and segmentation. The most recent new architecture is from Facebook AI Research (FAIR) and won best paper at the most prestigious computer vision conference: Computer Vision and Pattern Recognition (CVPR) in 2017. Their architecture was titled DenseNet[12] which introduced a new block called a Dense Block and stacked these blocks on top of each other, with some layers in between, to build a deep network. These dense blocks take the concept of residual networks a step further and connect every layer to every other layer. In other words, for a dense block, we consider all dense block before it as input and we produce an output that we feed into all subsequent dense blocks. To make the layers compatible with each other, we apply convolutions and batch normalizations. The benefit of doing this is that we encourage feature reuse, resolve the vanishing gradient problem, and have fewer parameters overall. This structure ensures the maximum information flow among all layers in the network, and directly connects all the layers (matching the size of the feature map). In order to preserve the characteristics of forward propagation, each layer gets additional input from all the layers ahead, and passes its own feature mapping to all subsequent layers. (See Fig. 1)

The L layer receives the feature-maps of all preceding layers, \( x_0, \ldots, x_{i-1} \) as input:

\[
x_i = H_i([x_0, x_1, \ldots, x_{i-1}])
\] (3.1.1)
where \( x_0, \ldots, x_{l-1} \) refers to the concatenation of the feature-maps produced in layers 0,.., (l-1).

Figure 1. Five layers of a DenseNet block with a growth rate of 4 feature-maps per layer (source [12]).

We use densenet201, which is made up of 4 dense blocks consisting total of 201 layers. Each layer involves applying a convolutional filter, followed by ReLU[28] activation and row-wise batch normalization [29]. A growth rate of [6, 12, 48, 32] feature-maps was used for each dense block. Details of the network structure are shown in Table 1.

Table 1. DenseNet architectures for Presence Detection of surgical tool.

<table>
<thead>
<tr>
<th>Layers</th>
<th>Output Size</th>
<th>DenseNet-201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>112x112</td>
<td>7x7 conv, stride 2</td>
</tr>
<tr>
<td>Pooling</td>
<td>56x56</td>
<td>3x3 max pool, stride 2</td>
</tr>
<tr>
<td>Dense Block(1)</td>
<td>56x56</td>
<td>[1x1 conv, 3x3 conv] X6</td>
</tr>
<tr>
<td>Transition Layer(1)</td>
<td>56x56</td>
<td>1x1 conv</td>
</tr>
<tr>
<td>Dense Block(2)</td>
<td>28x28</td>
<td>[1x1 conv, 3x3 conv] X12</td>
</tr>
<tr>
<td>Transition Layer(2)</td>
<td>28x28</td>
<td>1x1 conv</td>
</tr>
<tr>
<td>Dense Block(3)</td>
<td>14x14</td>
<td>[1x1 conv, 3x3 conv] X48</td>
</tr>
<tr>
<td>Transition Layer(3)</td>
<td>14x14</td>
<td>1x1 conv</td>
</tr>
<tr>
<td>Dense Block(4)</td>
<td>7x7</td>
<td>[1x1 conv, 3x3 conv] X32</td>
</tr>
<tr>
<td>Classification Layer</td>
<td>1x1</td>
<td>7x7 global average pool</td>
</tr>
</tbody>
</table>

Transfer Learning Pre-trained Model Features

Training a CNN from scratch is not a common practice when the datasets are not large enough. Instead the pre-trained convolutional networks are used either as a fixed feature detector or the weights are used to initialize the similar network followed by fine tuning the network for a specified task. For the tool detection tasks, we use the CNN architecture (see Table 1) which consists of convolutional layers similar to DenseNet architecture but the output of this layer corresponds to the confidences of the presence of the seven tools in the image. By applying thresholds to these confidences, we can determine the presence of the surgical tools in the image. All the weights of our model are initialized with the densenet201 pre-trained weights except the output layer which is
initialized randomly. The learning is formulated as a multi-classification problem with cross entropy as the loss function. The network is trained using stochastic gradient descent (learning rate of 0.01) with momentum of 0.9 until convergence. During learning, random cropping and flipping is performed for artificial data augmentation.

**Evaluation Metrics**

The tool presence detection is evaluated using mean average precision (mAP). This metric is obtained by computing the area under the precision-recall curve. The metric is first computed for each tool and then averaged over all the tools.

**Experiment and Result**

**Data Description.**

All the data we used in our training, validation and testing stages are from the data given by the M2CAI 2016 tool challenge. This dataset contains 15 videos of cholecystectomy procedures from University Hospital of Strasbourg/IRCAD (Strasbourg, France). The dataset is split into two parts: training subset (containing 10 videos) and testing subset (5 videos). We didn’t use any extra data. For the training data, since the raw data are videos with frames annotated at 1 fps. We extract frames from the video and took only the frames that are given in the annotation file. The seven tools are: (1) grasper, (2) bipolar, (3) hook, (4) scissors, (5) clipper, (6) irrigator, and (7) specimen bag. The annotation file contains a table, consisting of 8 columns. Every row (except for the header of the table) contains an annotation for an image in the video. The first column indicates the frame index of the annotated image in the video. The frame index is defined under a 0-based system. The other columns are the binary labels for the tools (0=not present; 1=present). (See Fig 2)

![Figure 2](image1.png)

Figure 2. Different images and their corresponding vector labels. The first image from left has the label which is of the form [grasper, bipolar, hook, scissors, clipper, irrigator and specimen bag].

**Training and Result**

We extract the images which have ground truth labels from the ten training videos and resize them into the same size (224 x224) since the videos have different dimensions. We use the data from the ten training videos as training and validation sets. For the five testing videos, we extract the images as required by the challenge as the testing set. We also resize them into 224 x224.

We introduce three kinds of data augmentation methods: horizontal flipping, vertical flipping, and rotation. In the implementation, we do not generate the augmented data set before training. Instead, we dynamically augment each image via each of the three augmentation methods in each epoch of the training process. For each image in a certain training epoch, it has 0.5 probability to be horizontal flipped. It also has 0.5 probability for other two augmentations. The three augmentation methods are
taken independently. Thus, we augment our training data set in a dynamic way to better train the models. We do not augment our validation set or testing set.

All the images are resized to \(224 \times 224 \times 3\) so the images can be directly used in the DensNet. The images are later normalized by dividing each pixel by 255 so that the range of pixels will be between \([0, 1]\) rather than \([0-255]\). The size of the dataset is increased by performing real time image augmentations randomly in each epoch of the whole training process. The model was implemented using mxnet\[30\] and the fine tuning was done for 100 iterations with a learning rate of 0.01. we trained the model on images from 15 videos and tested on 5 videos. At the same time, we randomly initialize a DensNet for training comparison.

From Fig.3, it can be seen that the accuracy of pre-trained DensNet201 and DensNet201 are both increased as the training gradually. The difference is that the model of pre-trained model converges faster. The loss function has the same situation. On testing data, the pre-trained model shows more smoothness and higher accuracy. The network accuracy without pre-trained is more volatile and it is not easy to converge. Through the comparison of the two models, we can see that DenseNet can detect surgical tools very well, but the initialization of network parameters will lead to different models performance. In addition, this is also related to the size of the dataset. The dataset of this experiment has only 15 videos, and the parameters of the network model reach 20,242,984. We only trained 100 iterations, which has reached 96% accuracy on the training set, and the best accuracy rate is 68% on the test set.

Result of tool Detection is presented in Table 2. In particular, Average Precision (AP) for each tool is reported, one can notice the low recognition results for scissors, Bipolar, and irrigator. So we analyzed the dataset. From Fig.4,5, It is clearly showed that, the distribution of the training data and testing data are imbalanced. This is most likely due to the fact that these tools are only present during short period of times in the procedure. For this reason, the training images for these tools are scarce in the dataset. In addition, these tools have appearance similarities with other tools that appear very often during the procedures However, a detailed investigation is needed for further understanding. In testing dataset, mean AP for our proposed pre-trained DensNet201 method is 62.9%. On the DenseNet model without pre-trained, the Average Precision of each surgical tool is very low, and the level of Average Precision was only 25%. The specific reasons are the same as we have analyzed above. Because our data sets are small and the model parameters are large, the training of such models.
is easy to overfit. From Fig.3, we can see without pre-trained densenet201 in the 100 iteration, the accuracy of the training dataset with the number of iterations increases gradually, and the accuracy of the testing on the training dataset is not convergent, and the loss function model has been reduced, this shows that our model has over fitting.

Table 2. Tool presence detection results.

<table>
<thead>
<tr>
<th>Tool</th>
<th>pre-trained DensNet201</th>
<th>DensNet201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grasper</td>
<td>0.866</td>
<td>0.572</td>
</tr>
<tr>
<td>Bipolar</td>
<td>0.110</td>
<td>0.027</td>
</tr>
<tr>
<td>Hook</td>
<td>0.939</td>
<td>0.733</td>
</tr>
<tr>
<td>Scissors</td>
<td>0.482</td>
<td>0.019</td>
</tr>
<tr>
<td>Clipper</td>
<td>0.708</td>
<td>0.206</td>
</tr>
<tr>
<td>Irrigator</td>
<td>0.533</td>
<td>0.038</td>
</tr>
<tr>
<td>SpecimenBag</td>
<td>0.772</td>
<td>0.271</td>
</tr>
<tr>
<td>Mean</td>
<td>0.629</td>
<td>0.256</td>
</tr>
</tbody>
</table>

Figure 4. The distribution of the training data.

Figure 5. The distribution of the testing data.

**Discussions and Conclusion**

In this paper, we have addressed the problem of presence detection of surgical tool based on Densely Connected Convolutional Networks. The results show that for tool presence detection pre-trained DenseNet architecture achieved the best accuracy is 68% and a mAP of 62.9% on the test set. The recognition results for scissors, Bipolar, and irrigator are low due to the unbalance of the dataset.

The tool Presence detection task is processed in the way of frame. That is to say, there is no temporal information in the detection process. For the future work it would be interesting to see whether the temporal information plays a role in the detection process. Setting up end - to - end architecture, temporal information can be combined with the use of recurrent neural networks.
Acknowledgement
This research was financially supported by the National Key Research & Development Plan of China (2018YFC0116704).

References


