FAST AUTOMATED LESION DETECTION

Aniruddha Tamhane*
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
atamhan3@jhu.edu

Parv Saxena*
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
psaxena6@jhu.edu

Wei-Lun Huang*
Department of Robotics
Johns Hopkins University
Baltimore, MD 21218
whuang44@jhu.edu

*Equal contribution by all three authors

ABSTRACT

Developing and applying deep neural networks for medical image processing[1], [2] is an active and promising area of research. Deep medical imaging has been applied to detecting, segmenting and classifying abnormalities (tumors, injuries, malignancies etc) in medical images across modalities such as MRI, CT-scans, X-ray images, fMRI etc. outperforming the average doctor in specific cases. Our work focuses on detecting lesions from Computed Tomography (CT) images for scans in the liver, lungs, abdomen, chest, kidneys, bone, soft tissue and pelvis using deep learning and machine learning. We use the DeepLesion dataset [2] released publicly by the National Institutes of Health to test our approach. We first set up our benchmarks by implementing the Faster RCNN [3], a state of the art object detection algorithm on the DeepLesion dataset. We then modify the PANet [4], another region-proposals based objection detection network to suit our problem of single-class object detection by removing the region proposals and utilizing the Feature Pyramid Network (FPN)[5] to extract multi-scale feature maps. We observe that one of the key challenges in parsing CT images is of occlusion by other internal body structures that are irrelevant to the diagnosis. To enable efficient training and accurate lesion detection, we propose the EncoderNet, an auto-encoder based lesion detection deep network that performs a deep image reconstruction to enable occlusion mitigation and feature map generation to localize the detected lesion. We also observe that a training set containing a mixture of lesion types (pelvis, bone etc) can potentially hamper effective learning. To reduce the training variance, we perform a dimensionality reduction step by extracting pre-trained VGG16 features for the training images followed by a K-means [6] clustering. We test and compare the effectiveness of the Faster RCNN, EncoderNet, the modified PANet and the clustering-based training approach through a comparative analysis and discuss the results. We see that in general, Faster RCNN outperforms the other models with a maximum mAP of upto 0.602, while the EncoderNet converges faster with a maximum mAP of upto 0.25. We also see that the clustering based training gives comparable results in a significantly lesser time.

Our code has been implemented in PyTorch [7] from scratch and is available in our GitHub repository. We also use the publicly available code for Faster RCNN from the Facebook research GitHub.

1 Introduction

1.1 Problem Statement

Our work focuses on developing an efficient and accurate methodology to automate lesion detection in CT images using deep learning and machine learning approaches. We aim to construct a deep learning architecture that detects lesions in presence of occlusion and a more efficient deep network training methodology. Our endeavours in this direction have been greatly facilitated by the public release of the annotated DeepLesion dataset released by NIH [2]. Before we explain our work, we will give brief overview of the Faster RCNN [3] and PANet [4] objection detection neural networks.

1.2 Faster RCNN

The Faster RCNN is a multi-class, multi-object detection deep network. It has the following information processing flow: generate feature maps from the input images, generate regions of interest (regions that possible contain an object), classify the proposed regions of interest into multiple classes. This is achieved by two modules: a Region Proposal Network (RPN) and the Fast RCNN module. The RPN generates the feature maps using a pre-trained model (eg: ResNet, VGG16) which are then used to classify possible regions spanning the entire image as "interesting" or "uninteresting". The proposed regions are then fed to the Fast RCNN [8] deep network to generate the 4 coordinates to determine a bounding box around the object and a label to classify the object within.
We observe that the region proposal network is essentially a low precision, high recall object detection module for a single - class object detection. We therefore propose that in our problem of detecting a lesion, it would suffice to simply extract rich feature maps and with an accurate bounding

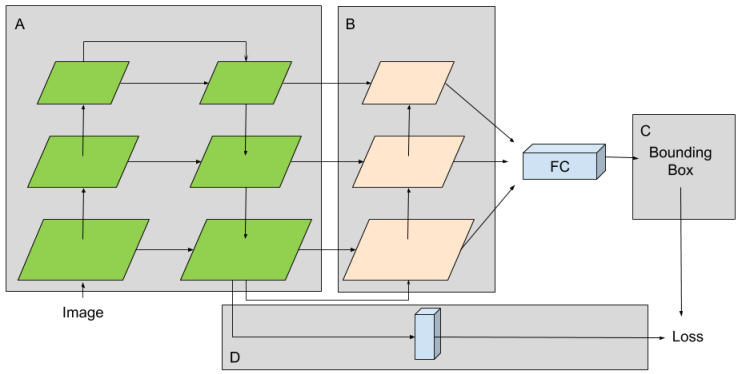

Figure 1: Schematic representation of EncoderNet architecture containing FPN based autoencoder (A), bottom-up path augmentation (B), bounding box regression network (C), occlusion removal (D)

box regressor module. We incorporate these modifications in our proposed EncoderNet lesion detection network.
The Faster RCNN also has a segmentation module which we will not focus on since it is not relevant to this paper.

1.3 PANet

The Path Aggregation Network for Instance Segmentation or PANet [4] is also a multi-class, multi-object detection deep network. It has a similar information processing flow as the Faster RCNN. A Feature Pyramid Network (FPN) [5] is used as the backbone for generating rich, multi-scale feature maps. It is characterized by skip-connections (similar to a U-Net) between lower level and higher level features. This aids in effective backpropagation of the gradient and multi-scale pooling of features. It further has an added bottom - up path augmentation module that generates features through an interaction of high-level and low-level features using the skip-connections. An RPN identical to the one used in Faster RCNN generates proposals from multi-scale feature maps in the FPN. Finally, a classifier and a box regressor are used to generate bounding-box coordinates along with classification scores.
We believe that the PANet has a superior method of generating feature maps as compared to the Faster RCNN. We gain confidence in this belief since PANet outperformed multiple variations of Faster RCNN in the 2017 COCO object detection challenge. Therefore, we utilize the PANet to build the architectural backbone of our proposed EncoderNet. However, we remain consistent with our earlier claim that an RPN is unnecessary for a single - class object detection and hence do not include it in our proposed model.

1.4 Our work

We try to solve the problem of lesion detection as follows: we implement the Faster RCNN object detector on the DeepLesion dataset. We use the implementation available on the Facebook research GitHub for the same. This helps us create benchmarks for comparing our other approaches. We then bifurcate our efforts: on one end to create a fast and accurate object detection deep network (EncoderNet) and on the other to improve model training by clustering the training images using deep feature extraction to reduce variance. We justify our efforts as follows: it has been reported by [2] that certain lesion types such as abdomen, mediastinum and pelvis are more difficult to detect because of occlusion by other internal body structures, and thus an effort to remove occlusions (using the autoencoder) can improve results for these lesion types; also, since the training dataset is a mixture of multiple lesion types, we can improve training by clustering training samples, with the clusters strongly correlating with lesion categories.
We perform the following experiments to test the efficacy of our approaches: training the Faster RCNN, EncoderNet and modified PANet on the DeepLesion dataset, and training Faster RCNN on the deep feature clustered dataset.
The EncoderNet is essentially the PANet without an RPN but with a customized occlusion-removal autoencoder module. We run experiments on the "modified PANet" i.e. a PANet without an RPN module to get an understanding of the individual contributions of the incremental changes made.

2 Proposed Solution

2.1 The EncoderNet architecture

A schematic flow of the EncoderNet architecture has been given in Fig 1. The EncoderNet has the following modules: an occlusion removal autoencoder, bottom-up path augmentation, multi-level feature pooling, box regression layer. The information processing flow is as follows: use the occlusion-removal autoencoder to reconstruct the input image to remove occlusion, use the bottom-up path augmentation to generate multi-scale feature maps and then the box regression layer to compute the bounding box coordinates. Each module is described as follows:

Table 1: mAP for Faster RCNN

Lesion Type	Faster RCNN (8k iters)	Faster RCNN with clustering (8k iters)	Faster RCNN (16k iters)
adomen	19.49%	30.34%	32.47%
mediastinum	27.76%	34.19%	48.29%
pelvis	16.76%	25.37%	38.14%

2.1.1 Occlusion-removal autoencoder

We observe that the FPN structure of the PANet can be used as an autoencoder in sync with the object detection modules to reconstruct the input image to remove occlusions. This would serve a dual purpose: a reconstructed input image will facilitate better feature map generation, and the autoencoder module trained on the object detection task can be used independently for CT image occlusion removal.
We try to achieve this objective by using the output of the FPN (which is downsized from the input image by a factor of 2) and passing it through a single transposed convolutional layer to obtain a 3-channel image of the same dimensions as the input image. We calculate the RMSE loss of this image with the input image and add it to the final box regression RMSE loss after scaling it with a factor of λ, which is a hyperparameter. Therefore, the final loss would be:

$$L(I;\lambda) = \sum_{i=1}^{N}(||b - \hat{b}||_2^2 + \lambda||I - \hat{I}||_2^2) \tag{1}$$

Here, $b, \hat{b}$ are the original and predicted bounding boxes while $I, \hat{I}$ are the original and reconstructed images. We hypothesize that with the correct hyperparameter, the autoencoder will learn the representation of the input image that will benefit the object detection task. Thus, it will try to remove occlusions from the input image, or perhaps highlight the lesion in a distinct manner.

2.1.2 Bottom-up path augmentation, feature pooling and box regression

The bottom-up path augmentation and feature pooling layers are identical to the ones implemented in PANet and have been briefly described in 1.3. We do not generate region proposals from the multi-level features as we hypothesize that it is a computationally expensive step and unnecessary for a single class, single object detection. The box regression layer is a 3 layer fully-connected deep network with ReLU activations.

2.2 The Deep Feature Unsupervised Clustering

We observe that the given dataset does not include labels for different types of lesions in the training set. Training the model on a dataset containing a mixture of multiple lesion types is not as efficient due to a large variance. We aim to reduce this variance by clustering the training dataset, mapping the individual clusters to a specific lesion class and training the model on specific clusters.
As a preliminary step, we first reduce the dimensionality from $512 \times 512 \times 3$ to 4096 by extracting the image features by passing through a pre-trained VGG16 network. We have confidence in this approach for dimensionality reduction since it has been demonstrated in [9] that VGG16 pretraining on ImageNet work really well for medical image transfer learning. We follow the procedure highlighted in [10] for extracting the features. We then use K-means clustering [6] for generating 8 hard clusters. We correspond each cluster to lesion type by using a majority vote after implementing the same clustering model on the annotated test and validation sets.

3 Dataset Description and preprocessing

We use the publicly available DeepLesion dataset for all our experiments. We also borrow the VOC annotations of the same dataset from the PASCAL VOC homepage. The orignial dataset contains 32,735 annotated lesion slices. We subset the original dataset to contain a single lesion per image, reducing the dataset to contain $\sim$ 22,000 training images, and $\sim$ 5000 test and validation images each.
We preprocess the images as follows: we subtract 32768 from each pixel intensity to obtain the Hounsfield Unit (HU) values, linearly scale the HU values from 0 to 255 and convert them into *int8* datatype. This form of preprocessing preserves a sufficient image quality while reducing the dataset size by a factor of 10. We generate a 3 channel image per key slice by concatenating the key slices with a slice 5mm above and below to gain a 3D perspective.

4 Results

The results have been reported for the following experimental setting: we have run all experiments on a Tesla p100 GPU provided by Google Cloud with 16GB RAM.
We use a learning rate of 0.004 and batch size of 4 for implementing Faster RCNN and a learning rate of 0.001, batch size of 8 and $\lambda = 0.0001$ for training the modified PANet and EncoderNet. We also decrease our learning rate every epoch by a factor 0.5 while training the EncoderNet and modified PANet. Our bencmark mAP value for the Faster RCNN model over the entire dataset is 46.89%. The maximum mAP achieved by the same model was 60.17% for the lung lesion detection. We compare this value with the state of the art performance achieved by Faster RCNN on the COCO dataset (65% - 75%) to come to a conclusion that the model was sufficiently trained and tuned.
Table 1 gives a comparison of the mean average precision (mAP) scores for the Faster RCNN model trained on the complete dataset with the same

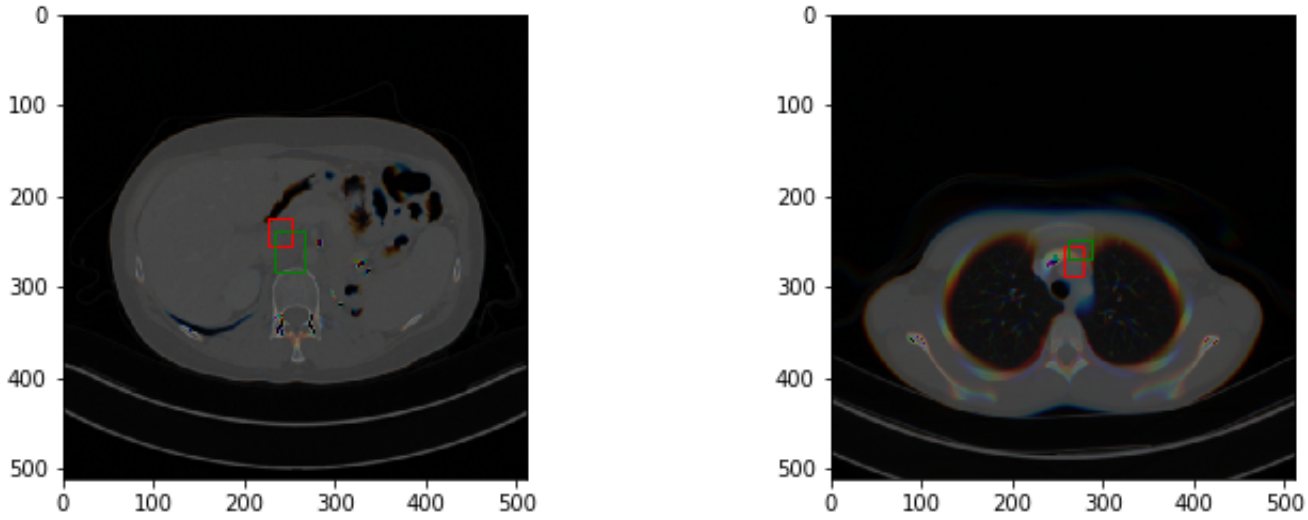

Figure 2: Examples showing actual lesion (green) and lesions detected by EncoderNet (red)

Table 2: mAP for modified PANet and EncoderNet

Lesion Type	mod-PANet	EncoderNet
bone	16.75%	13.22%
adomen	11.75%	13.70%
mediastinum	13.10%	11.45%
liver	17.51%	17.56%
lung	18.53%	16.06%
kidney	24.78%	24.82%
soft tissue	19.97%	17.26%
pelvis	8.72%	9.11%

model trained on the clusters. We give the results for the clusters corresponding to the adomen, mediastinum and pelvis lesions since they are the most challenging to detect. We see that clustering gives a superior detection accuracy for the same number of iterations and a comparable accuracy for twice the iterations. This can be explained as follows: clustering reduces the variance in training dataset enabling a lesion specific training. Thus, the model learns the features relevant to a specific lesion type faster, as can be seen from the results.
Table 2 gives a comparision of the mAP values of the modified PANet and the EncoderNet for all lesion types. We observe that the performances are comparable and that the EncoderNet is marginally better than the mod-PANet for the difficult lesions types suffering from occlusion. At this moment, we are unable to arrive at a concrete conclusion; however, the autoencoder structure of the EncoderNet seems promising in detecting occluded lesions.
A sample of our obtained results can be seen in Figure 2.

5 Conclusions and suggestions based on successful and failed experiments

We conclude that clustering the training dataset with features extracted using deep networks gives significantly improved results in a shorter time. We attribute this to a smaller, specific training dataset created thorugh clustering. This also confirms the correctness of our clustering approach. Further, we observe that the EncoderNet outperforms the modified PANet for detecting difficult, occluded lesions in the abdomen, mediastinum and pelvis. Though it is premature to arrive at a concrete conclusion, we see a promising performance in detecting occluded lesions.
We observe that the occlusion-removal autoencoder does not work as expected for generating an occlusion-free image for human interpretation. This may be because of poor hyperparameter tuning and interference caused by bounding box MSE loss gradients with the reconstructed image MSE loss gradients. These shortcomings can be addressed by tuning hyperparameters and adding richer feature extraction layers to offset the task of feature map generation from the autoencoder layers to obtain a highly interpretable image. We also observe that the hyperparameter λ is highly sensitive, and hence adjusting λ every epoch can be extremely beneficial.
We also observed in course of training that the Intersection Over Union (IOU) loss does not converge unless the network is initialized to give coordinates close to the actual coordinates. This might be because the IOU loss has a very small gradient away from the actual solution, as seen in Fig 3. We suggest that using MSE loss for initial few epochs and IOU loss for the latter epochs can result in excellent training.

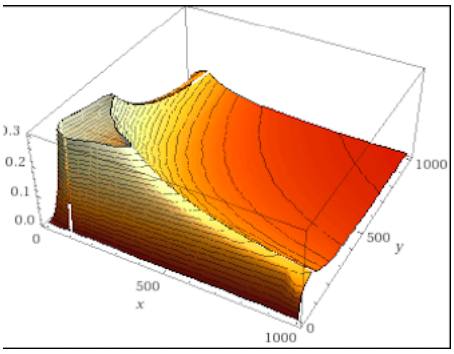

Figure 3: IOU loss for a bounding box having coordiates at (0,0) and (512, 512) and one of the predicted coordinates fixed at (0,0)

We believe that the EncoderNet and modified PANet perform poorly compared to Faster RCNN due to insufficient training and tuning.
We suggest using the PyTorch implementations provided by Facebook research as it supports stable batch training unlike other commonly available implementations.

6 Acknowledgement

We are grateful to Dr Mathias Unberath (course instructor), Cong Gao and Jie Ying Wu (course TAs), Sonakshi Grover (project mentor), Intuitive Surgical (sponsor) and Google Cloud (cloud sponsor) for guiding and supporting this project.

References

[1] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[2] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M Summers. Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *Journal of Medical Imaging*, 5(3):036501, 2018.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[4] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[5] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] Chuen-Kai Shie, Chung-Hisang Chuang, Chun-Nan Chou, Meng-Hsi Wu, and Edward Y Chang. Transfer representation learning for medical image analysis. In *2015 37th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 711–714. IEEE, 2015.

[10] Joris Guérin, Olivier Gibaru, Stéphane Thiery, and Eric Nyiri. Cnn features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700*, 2017.