Skin Lesion Segmentation using Deep Hypercolumn Descriptors

Dhanesh Ramachandram
Graham W. Taylor

Abstract

We present a new image segmentation method based on deep hypercolumn descriptors which produces state-of-the-art results for the segmentation of several classes of benign and malignant skin lesions. We achieve a Jaccard index of 0.792 on the 2017 ISIC Skin Lesion Segmentation Challenge dataset.

1 Introduction

One of the fundamental and most challenging tasks in digital image analysis is semantic segmentation, which is the process of assigning pixel-wise labels to regions in an image that share some high-level semantics. In this paper, we focus on the task of accurately segmenting benign and malignant skin lesions in dermatoscopy images as a means of lesion quantification. Among the skin lesions considered in our work is melanoma which is an aggressive malignant tumour originating from melanocytes cells — skin cells responsible for the production of melanin. The American Cancer Society estimates that in 2017, in the United States alone, more than 87,000 new melanoma cases will be diagnosed with an estimated 9,300 fatalities [1]. Skin melanoma lesions share similar visual characteristics with other benign skin lesions such as nevus and seborrhoeic keratosis as shown in Fig. 1.

Fig. 1: Examples of various skin benign and malignant skin lesions

Skin lesion segmentation is challenging due to a variety of factors, such as variations in skin tone, uneven illumination, partial occlusion due to the presence of hair, low contrast between lesion and surrounding skin, and the presence of freckles or gauze in the image frame, which may be mistaken for lesions. A successful lesion segmentation technique should be robust enough to accommodate this variation. Fig. 2 shows an example dermatoscope image of a skin lesion and its corresponding binary mask.

Fig. 2: Left: An example of a skin lesion image with a blue fiducial marker in the background. Right: The corresponding ground truth binary segmentation mask.

2 Related Work

Skin lesion segmentation is a widely researched topic in medical image analysis [2]. Until recently, most skin lesion segmentation approaches were based on meticulously designed image features [3, 4, 5]. Such approaches often require extensive pre-processing and post-processing approaches such as hair removal, edge-preserving smoothing and morphological operations. Therefore the robustness of such approaches could be limited as each new scenario may require custom tuning.

An alternative approach to manually crafting image features for segmentation is to instead leverage deep neural networks to automatically learn robust image features given sufficient labeled examples. Deep learning is becoming the dominant approach for many medical imaging problems [6] and has seen tremendous success for the related skin lesion classification task [7]. Early deep learning approaches to image segmentation used a patch-wise training strategy [8, 9], where overlapping patches are used to train a convolutional neural network to predict the value of the pixel centered on each patch. While this approach overcame the requirement of having a large labeled dataset, the approach was computationally inefficient due to obvious redundancies in information contained in overlapping patches. Long et al. [10] proposed the Fully Convolutional Neural Network (FCN) architecture which has become the mainstream approach to deep semantic segmentation and many variants have been proposed since. In FCNs, the usual fully-connected and final prediction layers of convolutional neural networks (CNNs) are replaced with convolutional layers to facilitate dense prediction. In order to learn contextual information contained in images, CNNs use pooling operations (e.g. max-pooling) or strided-convolutions, that produce downsampled outputs across the layers of the network resulting in a much smaller prediction mask. Therefore, FCNN architectures require a single upsampling or several progressive upsampling or “de-convolution” layers to upscale the pixel-wise predictions of the network to match the dimensions of the input image. In the 2017 skin lesion segmentation challenge (ISIC2017: Skin Lesion Analysis Towards Melanoma Detection), 70% of the submissions, and 9 out of the top 10 submissions employed deep learning strategies for segmentation.

Deep learning architectures often require a large labelled dataset, which is uncommon in the medical domain. Recently, transfer learning approaches (i.e. fine-tuning a pre-trained network on a limited, but different dataset) has been successful [11, 12, 13]. Nevertheless, the mechanisms of transfer learning and why such approaches work on vastly different domains has been challenging to interpret [14].

Our approach to skin lesion segmentation is based on the idea of using hypercolumn descriptors first proposed by [18]. Hypercolumn descriptors for a given pixel are formed by extracting activations from multiple convolutional layers of a CNN that correspond to the same pixel. These multi-scale descriptors, which capture rich semantic, localization and transformation information, can then be used to train a non-linear classifier to perform pixel-wise predictions. Hypercolumn descriptors have been applied to problems such as semantic segmentation [18], edge detection, surface normal estimation [19] and auto-colourization of grayscale images [20]. We demonstrate its effectiveness for the challenging skin lesion segmentation problem and show state-of-the-art performance on the ISIC2017 Skin Lesion Segmentation Challenge [1], which is larger and more challenging than the previous dataset used in a similar challenge (ISBI 2016). The training dataset consists of 2000 dermatoscopy images of three types of skin lesions: nevus, seborrhoeic keratosis and melanoma — the latter lesion being malignant — and their binary masks. The masks were created by an expert clinician, using either a semi-automated process (using a user-provided seed point, a user-tuned flood-fill algorithm, and morphological filtering) or a manual process (from a series of user-provided polyline points). Fig. 2 shows an example lesion and its corresponding binary mask.

3 Methodology

Our skin lesion segmentation model is an adaptation of the PixelNet architecture [19]. The PixelNet architecture uses the convolutional layers of the VGG16 [21] architecture to form hypercolumn descriptors using sparsely sampled pixels from input images during training.

http://challenge2017.isic-archive.com
These descriptors are then used to train a 2-layered multi-layer perceptron (MLP) to perform pixel-wise prediction. We demonstrate that it is possible to achieve state-of-the-art segmentation performance by training the network from scratch using a relatively small dataset.

### 3.1 Preprocessing

For this application, the input images and the corresponding ground-truth masks are first resized to 224 by 224 pixels to match the resolution of images in the pre-training stage. When using a pre-trained VGG16 net to extract the hypercolumns, we retained the normalization of the input images using the mean channel-wise pixel intensities computed for the entire ImageNet dataset. We perform data augmentation on-the-fly by randomly rotating both the image and its mask by 90-degree increments as well as flipping the images. In addition, we also randomly varied the image brightness, hue and contrast (within a small range) for each minibatch.

### 3.2 Deep Hypercolumns

During the training phase, we feed the input image to a VGG16 network and extract the sparse hypercolumn descriptors from selected convolutional layers. The hypercolumns are formed by concatenating a series of activations of the convolutional layers. In our implementation, we chose the activations from the final convolutional layer from each convolutional and fully-connected block in the VGG16 architecture (i.e., $\text{conv}1_2, \text{conv}2_2, \text{conv}3_3, \text{conv}4_3, \text{conv}5_1$, and $\text{FC}_2$ layers) to form the hypercolumn. The fully connected layers in the original VGG16 network are implemented as $1 \times 1$ convolution layers. As each convolutional block is preceded by a max-pooling operation that downsamples the activations, we perform bilinear upsampling using an appropriate scaling factor such that the resulting resolution for the activations of each layer forming the hypercolumn is 224 by 224. Then, we sparsely sample random points from the dense hypercolumns to form rich descriptors for a given pixel in the input image. The sparse hypercolumn descriptors are then used to non-linear classifier, in our case, a 2-layered MLP (again, implemented as $1 \times 1$ convolutions) with 4096 and 2048 neurons respectively. We use a sparsely-sampled output mask, whose pixels correspond to the location of the sparse hypercolumns to learn pixel-wise class predictions.

### 3.3 Training

We implemented our network using TensorFlow and experimented with fine-tuning an ImageNet pretrained VGG16 as well as training the entire network from scratch. In both cases, we used batch normalization and ADAM optimization with an initial learning rate of $10^{-3}$ using a per-pixel cross-entropy loss function. Since we construct hypercolumn descriptors from sparsely sampled pixels, we empirically found that using a sample size of 1600 pixels from a batch size of 5 images provided best results. Training typically converges after 120 epochs on a NVIDIA Titan-Xp GPU with 12Gb of RAM. This takes around 3 hours. During inference, we turn off sparse random sampling and use the dense hypercolumns for image segmentation.

### 4 Results and Discussion

Example segmentation outputs from our lesion segmentation architecture are shown in Fig. 4. It can be seen that the model produces accurate segmentations for a wide range of skin lesion appearances in the dermoscopy images. Tab. 1 shows the comparative performance of the method on the 2017 ISIC lesion segmentation challenge. Our model achieves significantly higher Jaccard score than the best submissions for the 2017 ISIC skin lesion segmentation challenge and can be effectively trained from scratch using a relatively small dataset. The ranking of segmentation quality is based on the Jaccard Index, as used in the challenge, which measures the degree of overlap between the predicted segmentation and the expert-annotated ground truth masks. It is defined as: $J_A = \frac{TP}{TP + FN}$, where $TP$ is the number of True Positives, $FN$ is the number of False Negatives and $FP$ is the number of False Positives. We found that a network trained from scratch produces significantly better segmentation performance as opposed to fine-tuning a pre-trained network. This may be attributed to the misalignment between the distribution of images in ImageNet and the distribution of dermoscopy images in our target dataset.

Despite the impressive results obtained for the skin lesion segmentation, the model has a large number of parameters and the computation of dense hypercolumns during inference is computationally intensive. We are currently working to reduce the model size and inference times so that a relatively lean and fast model can be deployed on mobile-phones.

### Acknowledgements

We thank OCE and NSERC for jointly funding this research project, and CFI for funding computational infrastructure. We also thank NVIDIA for the generous hardware donation for use in the project.

### References


**Fig. 3:** An illustration of the network architecture used in this work. The hypercolumn descriptors are constructed by concatenating a series of multi-scale activations from convolutional layers of a VGG16 net.

**Table 1:** Results for 2017 ISIC Skin Segmentation Challenge dataset against the top-3 submissions

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Method</th>
<th>Fine-tuned</th>
<th>Accuracy</th>
<th>Jaccard Index</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>Deep hypercolumn descriptors</td>
<td>No</td>
<td>0.916</td>
<td>0.792</td>
<td>0.790</td>
<td>0.954</td>
</tr>
<tr>
<td>Ours</td>
<td>Deep hypercolumn descriptors</td>
<td>Yes</td>
<td>0.900</td>
<td>0.769</td>
<td>0.783</td>
<td>0.937</td>
</tr>
<tr>
<td>23</td>
<td>Fully convolutional-deconvolutional</td>
<td>No</td>
<td>0.934</td>
<td>0.765</td>
<td>0.825</td>
<td>0.975</td>
</tr>
<tr>
<td>24</td>
<td>U-Net</td>
<td>No</td>
<td>0.932</td>
<td>0.762</td>
<td>0.802</td>
<td>0.978</td>
</tr>
<tr>
<td>25</td>
<td>ResNet (with additional 8K training images)</td>
<td>Yes</td>
<td>0.934</td>
<td>0.760</td>
<td>0.802</td>
<td>0.985</td>
</tr>
</tbody>
</table>

**Fig. 4:** Examples of segmentation output using our approach.


