Automatic Skin Lesion Analysis with Deep Networks

Yuan Xue[1*†], Lijun Gong[3†], Wei Peng[2*], Xiaolei Huang[1], Yefeng Zheng[3]
[1]Penn State University, [2]Harvard University, [3]Tencent Youtu Lab
https://youtu.qq.com/

Abstract

Melanoma is one of the most clear-cut cases of cancer in which early detection is key in ensuring the effective treatment [10]. While dermoscopic images using skin surface microscopy provide a new method for malignant melanoma examination, early detection and accurate identification remain to be a challenging problem. Moreover, the automatic diagnosis of such melanoma cancer on dermoscopic images can reduce the clinicians' workload greatly and help improve the diagnostic accuracy. To this end, we develop and evaluate several deep learning models on the "ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection" grand challenge dataset [4]. With the data from [15], we present models and results on all three tasks, i.e., lesion segmentation (task 1), lesion attribute detection (task 2), and disease classification (task 3).

1. Lesion Segmentation

Based on our previous SegAN [16], we build a deep learning framework with adversarial training for skin lesion segmentation for ISIC2018 task1. Specifically, we train our models using the fully annotated ISIC 2018 task1 training dataset, which consists of 2594 dermoscopic images and 2594 corresponding ground truth response masks. Since these training images have various sizes but most of them have an aspect ratio of roughly $4/3$, we first resize an input image to size 336 (width)$\times$296 (height). Then we further randomly crop the input to size 224×224 during training for the purpose of data augmentation. We also randomly flip the input images horizontally and vertically with probability 0.5 and randomly rotate the image with $90, 180, 270$ degrees. Color Jitter, including randomly changing the brightness, contrast, saturation and hue values of the input image, is also applied for data augmentation. We utilize a 5-fold cross validation during the training to evaluate the quality of the trained models since we do not have access to the ground truth of the test set.

We choose the input size to be 224×224 due to the reason that we want to incorporate some feature extractors which are pre-trained on the ImageNet [12] into the segmentation framework and also for the consideration of training speed and memory usage. With a smaller image size, we could have a larger batch size and faster convergence rate. We did not add dropout during the training since we have adopted several data augmentation techniques during the training.

We train all networks using the Adam optimizer[8] with batch size = 25. The initial learning rate is set to be 0.001, and a linear learning rate schedule is implemented. For post-processing, we leverage the Conditional Random Fields (CRFs) [9] with morphological operations to refine the generated segmentation masks and remove the isolated small regions.

The proposed SegAN consists of two parts: the segmentor network S and the critic network C. The segmentor is a fully convolutional encoder-decoder network that generates probability label maps from input images. The task for the segmentor S is to generate the segmentation mask corresponding to an input image; the task for the critic network C is to distinguish two types of inputs: original images masked by ground truth label maps, and original images masked by predicted label maps from S. During an adversarial training process, the critic forces the segmentor to learn to generate more accurate segmentation results for training images. During testing, only the segmentor S is utilized to generate the predicted label map for a test image.

The S and C networks are alternately trained by back-propagation in an adversarial fashion: the training of S aims to minimize the multi-scale L_1 loss between the feature maps of the predicted masked image and the groundtruth masked image, while the training of C aims to maximize the same loss function. More specifically, we first fix S and train C for one step using gradients computed from the loss function, and then fix C and train S for another step using gradients computed from the same loss function passed from C to S. After each iteration of training C, the weights of our critic network are clamped to some certain range (e.g., [0.05, 0.05] for all dimensions of the parameter,

*Work done during an internship at Tencent Youtu Lab.
†Authors contributed equally.

Table 1. Task1 Results on ISIC2018 Validation Set

Metric	Deeplab V3+	VGG+Adversarial	VGG+Adversarial+Ensemble
Threshold Jaccard	0.77	0.80	0.81

refer to [17] for more details). As training progresses, both S and C become more and more powerful. And eventually, the segmentor will be able to produce predicted label maps that are very close to the ground truth.

In our proposed SegAN, given a dataset with N training images x_n and corresponding ground truth label maps y_n, the multi-scale objective loss function $\mathcal{L}$ is defined as:

$$\min_{\theta_S}\max_{\theta_C}\mathcal{L}(\theta_S,\theta_C) = \frac{1}{N}\sum_{n=1}^{N}\ell_{\text{mae}}(f_C(x_n\circ S(x_n)), f_C(x_n\circ y_n)) \ , \tag{1}$$

where ℓ_{mae} is the Mean Absolute Error (MAE) or L_1 distance; $x_n\circ S(x_n)$ is the input image masked by segmentor-predicted label map (i.e., pixel-wise multiplication of predicted_label_map and original_image); $x_n\circ y_n$ is the input image masked by its ground truth label map (i.e., pixel-wise multiplication of ground_truth_label_map and original_image); and $f_C(x)$ represents the hierarchical features extracted from image x by the critic network. More specifically, the ℓ_{mae} function is defined as:

$$\ell_{\text{mae}}(f_C(x), f_C(x')) = \frac{1}{L}\sum_{i=1}^{L}||f_C^i(x) - f_C^i(x')||_1 \ , \tag{2}$$

where L is the total number of layers (i.e. scales) in the critic network, and $f_C^i(x)$ is the extracted feature map of image x at the ith layer of C.

We used a U-net [11] like encoder-decoder structure as the segmentor for skin lesion segmentation. The backbone encoder used in our SegAN framework is a VGG19 [14] pre-trained on ImageNet. To make better use of the channel information and selectively enhance the useful features, we applied SE blocks [7] to the feature extractor. The critic network used in SegAN is a convolutional network with 5 global convolution blocks. In each global convolution block, we use a two-way 1×9 and 9×1 instead of the original 9×9 convolution kernel to reduce the number of parameters while remaining a large receptive field along with a batchnorm layer and a relu activation. Each global convolution layer is also followed by a normal 3×3 convolution, a batchnorm layer and a relu activation.

During the training, we first fix the weights of the pre-trained VGG net for 50 epochs and only train other parts of the segmentor and the critic network. After 50 epochs, we train the whole network together to fine-tune the weights in the VGG feature extractor.

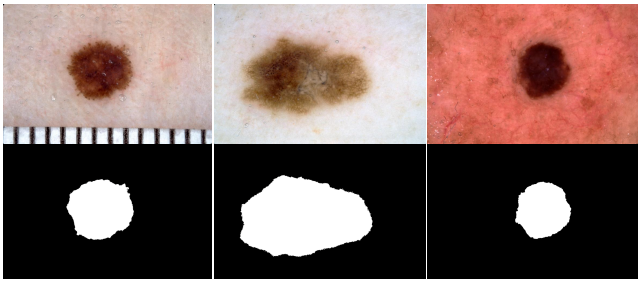

Figure 1. Example results of our SegAN on ISIC 2018 task 1 validation set. The first row is the original dermoscopic images and the second row is the generated skin lesion masks.

In this year's ISIC task 1, the final evaluation metric for each image is computed as a threshold of the Jaccard index, where all Jaccard indexes less than 0.65 are counted as 0. In this case, the segmentation results of hard cases will affect the final score significantly. To force our model to focus more on the hard cases during the training, we design a hard-mining loss as an auxiliary loss. Specifically, we will compute the soft dice score of each image and rank the dice loss (i.e., $1-$ soft dice score) then take the 5 cases with highest dice losses and do the back-propagation. Intuitively, this strategy should help the model focus more on hard cases and achieve better performance. Unfortunately, we did not observe any improvements either on our validation set or the official validation set, so we give it up in our final submission. We believe that this hard-mining method still has potential and we will keep tuning hyper-parameters and do more experiments in the future.

For final submission, we also calibrate the result by model ensembles. In addition to the main model with the VGG encoder and adversarial training, we also train a deeplab v3 plus model [3] from scratch and the final result is an ensemble of 4 VGG models trained by adversarial loss and 2 deeplab models trained by pixel-wise cross-entropy loss.

2. Lesion Attribute Detection

In task 2, we propose a multitask learning approach to automatically predict the locations of dermoscopic attributes within dermoscopic images. The task 2 skin lesion attribute detection aims at predicting the locations of the dermoscopic attributes, i.e., pigment network, negative network, streaks, milia-like cysts, and globules within images. Since different attributes lie in different locations of the dermoscopic images, we treat this task as a multi-segmentation problem. Based on task 1 result, we modify the pre-trained

VGG19 as a feature extractor. Unlike the model in task 1 which only generates a binary segmentation mask, a $1\times1\times5$ convolution filter is added to the last block of the segmentation network to output the segmentation masks for each of the 5 classes in task 2. In addition, a fully connected layer is added to the segmentation network to classify different attributes. In the last layer, we apply two parallel element-wise softmax layer to generate both the segmentation and the classification outputs.

In the field of computer vision, prior works have shown that multi-task learning (MTL) [5] is able to significantly improve the performance of multiple tasks by learning all tasks jointly, since learning multiple relevant tasks could potentially learn more general and robust features compared to learning each of the tasks independently. Inspired by the idea of MTL and consider the fact that task 1&2 share a lot in common, we try to improve the performance of our model in task 2 by jointly learning to predict the image attributions (i.e., classification) and the segmentation masks from task 1.

Data As in task 1, the training dataset consists of $2,596$ images, and each image has 5 corresponding segmentation mask images with different attributes. According to the mask, we get attributes label presented as a one-hot vector. We scale the size of the input to 224×224. Similar to task 1, we randomly flip training data horizontally and vertically, randomly rotate the data and apply the Color Jitter for the purpose of data augmentation and reducing over-fitting.

Training As described above, we adopt a multi-task learning strategy for this task. We train our model to minimize the cross-entropy loss for segmentation task and train via a multi-label soft-margin loss for the classification task using the Adam optimizer for both tasks with batch size 32. The maximum number of training iterations is set to be 300 epoch. The initial learning rate is set to be 0.001 and is declined by the factor of 0.1 for every 100 epochs.

Validation results The validation data provided by the challenge includes 100 images, which are the same as the validation data in task 1. The final submission is evaluated by the Jaccard index. We can see that the Jaccard index is not satisfying on the validation set which indicates the difficulty of this task. However, the validation results show that the multi-task learning strategy improve attribute detection performance considerably from 0.37 to 0.43.

Table 2. Task2 Results on ISIC2018 Validation Set

Metric	VGG	VGG+Multitask training
Jaccard	0.37	0.43

3. Disease Classification

In task 3, we propose an end-to-end framework to classify the dermoscopic images. The object of task 3 is to classify skin lesion images into seven categories: melanoma (MEL), melanocytic nevus (NV), basal cell carcinoma (BCC), actinic keratosis / Bowen's disease (intraepithelial carcinoma) (AKIEC), benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis) (BKL), and dermatofibroma (DF), and vascular lesion (VASC). The training set contains $10,015$ images, but the given data exhibit highly-skewed class distribution. While most of the data $(6,705)$ belong to the NV class that is the majority class, the minority classes only contain a scarce amount of samples. For instance, DF and VASC classes only contain 155 and 142 images, respectively. To mitigate the imbalanced class distribution issue, we acquire another 162 images (65 VASCs and 98 DFs) from the Dermofit Image Library [1] and add them to the two minority classes. Based on this observation, the model is expected to learn a deep representation feature of the imbalanced data to maintain both the inter- and intra-class margin between feature maps. By adding the feature distance as an extra constraint, we can guarantee the inter- and intra-class margins during the learning of our proposed model through the metric learning such as triplet loss[13].

Data The task 3 dataset contains $10,015$ images which belong to 7 different classes (i.e., MEL, NV, BCC, AKIEC, BKL, DF, and VASC), where all classes are mutually exclusive. Considering the training speed and GPU memory usage, we first resize our input images to 360 (width) $\times270$ (height) then randomly crop them to 256×256. Besides, each image is randomly flipped vertically or horizontally during the training for data augmentation.

Triplet sampling Since the training dataset is highly imbalanced, our target is to learn a deep feature embedding $f(x)$ mapped from an image x to some d-dimensional feature space $\mathbb{R}^d$, so that the features are discriminative enough to be distinguished from other classes. To achieve this goal, we first draw triplet samples from the imbalanced data where 2 of them are from the same minor class and 1 is from the major class. Then we can compute the feature distance between these triplet samples and make sure the inter-class distance is greater than the intra-class distance. The details of our triplet sampling strategy are discussed in the following sections.

The randomly drawn triplet samples are defined as:

x_i : an anchor from minor class,

x_i^p : image from the same class with anchor,

x_i^n : image from major class.

Based on the triplet samples, we add an auxiliary constraint so that:

$$\|f(x_i) - f(x_i^p)\|_2^2 + \alpha \leq \|f(x_i) - f(x_i^n)\|_2^2, \quad (3)$$

whereα is the feature margin between major class and minor class. $\|f(x_i) - f(x_i^p)\|_2^2$ is computed via Euclidean distance.

Triplet loss To enforce the previous relationship during the deep feature learning, a triplet loss is applied alongside the cross-entropy loss as:

$$loss = \sum_{i=1}^{n} (\|f(x_i) - f(x_i^p)\|_2^2 + \alpha - \|f(x_i) - f(x_i^n)\|_{0^+} \tag{4}$$

The triplet loss has been widely applied in many vision tasks[13], which ensures that the data belonging to the same class are closer to each other compared to the data from different classes. To satisfy this constraint, we generate mini-batches of triplets, i.e. an anchor x_i, a positive instance x_i^p from the same class with anchor, and a negative instance x_i^n from a different class. There are several ways to generate such triplets: offline or online triplets, triplets with hard or hardest negative mining[13]. In this task, we use the online version with hardest negative mining to generate triplets during training of our model.

Training We use the ResNet-101[6] feature extractor which is pre-trained on ImageNet as our framework. We add a fully connected layer as the 256 dimension feature vector after the last pooling layer, and modify the last fully connected layer to have 7 outputs as the final classifier. The whole network is trained by the cross-entropy loss as well as the triplet loss on the 256 dimension feature vector alternately. The triplet sampling strategy provides a strong constraint so that the model can better distinguish the minor classes from the major classes. The triplets are generated in an online fashion every 10 epochs and the features are trained by the triplet loss as shown above. We use the SGD optimizer[2] with momentum 0.9 and the initial learning rate is set as 0.001 which will be declined by the factor of 0.1 every 100 epochs. The batch size is set to be 64, and the training stops at the 500 epoch.

Results The final output of our model given a new input image is a vector of predicted probabilities of 7 classes. Instead of picking the class with the highest probability as the final prediction, we make predictions based on the similarities between the features of newly given input images learned by our model and those of the randomly sampled images in each class. To do so, we first draw 100 samples for each class from the labeled training data and extract their features from the trained model before the last softmax layer. Then we obtain the feature vectors of the newly given images. After calculating the mean Euclidean distance between the feature vectors of the new inputs and the 100 samples from each class, the class with the lowest mean Euclidean distance is chosen as the final prediction label.

Now we show our validation results. To demonstrate how triplet sampling helps to learn the intra-class differences, we compare the balance accuracy on 193 validation data with and without the triplet sampling, in which one model is trained by both the cross-entropy loss and the triplet loss, while another is trained by cross-entropy loss only as a baseline. Both experiments are conducted using the same hyper-parameters as shown above. The balanced accuracy of a single model for each method can be found in Table 3.

Table 3. Task3 Results on ISIC2018 Validation Set

Metric	CE loss only	Triplet sampling & CE loss
Balanced Accuracy	0.88	0.92

References

[1] L. Ballerini, R. Fisher, B. Aldridge, and J. Rees. A color and texture based hierarchical k-nn approach to the classification of non-melanoma skin lesions. *Color Medical Image Analysis*, pages 63–86, 2013.

[2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018.

[4] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, pages 168–172. IEEE, 2018.

[5] R. Girshick. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[10] D. S. Rigel, R. J. Friedman, A. W. Kopf, and D. Polsky. Abcdean evolving concept in the early detection of melanoma. *Archives of dermatology*, 141(8):1032–1034, 2005.

[11] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[13] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[15] P. Tschandl, C. Rosendahl, and H. Kittler. The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *arXiv preprint arXiv:1803.10417*, 2018.

[16] Y. Xue, T. Xu, and X. Huang. Adversarial learning with multi-scale loss for skin lesion segmentation. In *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, pages 859–863. IEEE, 2018.

[17] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang. Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. *Neuroinformatics*, pages 1–10, 2018.