Deep Learning for Skin Lesion Segmentation and Classification: ISIC Challenge 2018

Frederico Guth*, Teófilo E. de Campos †
Departamento de Ciência da Computação, Universidade de Brasília (UNB)
*fredguth@fredguth.com, †teodecampos@unb.br

Abstract—**This report describes our approach for the ISIC Challenge 2018 - Skin Lesion Analysis Towards Melanoma Detection. For Part 1 - Lesion Segmentation, we developed a U-net based convolutional neural network pretrained with the ImageNet dataset [1] and applied several data augmentation and hyperparameters optimization strategies, obtaining threshold(0.65) jaccard of 0.775. For Part 3 - Lesion Classification, we developed an ensemble strategy that leverages pretrained convolutional networks with our results from Part 1, obtaining an online score of xxxx.**

I. Introduction

According to the World Health Organization, between 2 and 3 million non-melanoma skin cancers and 132,000 melanoma skin cancers occur globally each year [2]. Despite representing less than 6.5% of all skin cancers, melanomas are the most dangerous type, accounting for aproximately 75% of all skin cancer related deaths [2], [3]. Visual inspection still is the most common diagnostic technique and early detection is critical to increase survival expectancy.

Deep convolutional neural networks (CNNs) already exceed human performance in visual classification [4]. In an attempt to improve the scalability of diagnostic expertise, CNNs have been developed to locate and classify skin cancers in images with dermatologist-level accuracy [3].

Dermoscopy is a technique for examination of skin lesions that, with proper training, increase dianostic accuracy from 60% (unaided expert visual inspection) to 75%-84% [**?**]. The International Skin Imaging Collaboration (ISIC) has a large-scale publicly acessible dataset of more than 20,000 dermoscopy images and host an annual benchmark challenge on dermoscopic image analysis since 2016. The challenge comprise 3 tasks of lesion analysis: Part 1 - Segmentation, Part 2 - Dermoscopic feature extraction, Part 3 - Classification.

In this paper, we describe our approach for the ISIC Challenge 2018:

- **Section II** describes our methodology for **lesion segmentation**;
- **Section III** describes our methodology for **lesion classification**;

II. Lesion Segmentation

A. Computational Resources and Development Framework

We used a Paperspace GPU Cloud service for running all our experiments. The instance server used had: 8 cores CPU with 30GB of RAM, Quadro P4000 8 core GPU with 8GB of RAM. The code was developed in PyTorch and Fast.ai [5] on Jupyter Notebooks.

B. Data and Augmentation

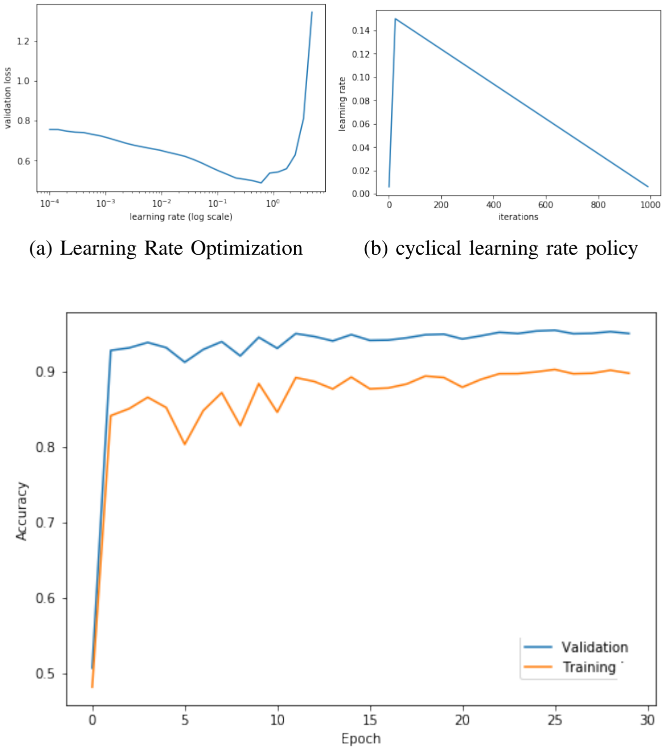
We used "ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection" grand challenge datasets [6], [7] and no aditional external data. All images were first resized to 128x128 pixels, 256x256 and 512x512; and preprocessed to adjust color balance. Random transformations on input images to agument the dataset were made: dihedral transformation, rotation (up to 44 degrees), zooming (up to 1.05), fliping and random lightining changes. The official training dataset was then splited in 3-folds of training and validation datasets.

C. Segmentation Model: Unet34

Introduced in 2015, U-net is an encoder-decoder architecture designed for biomedical image segmentation [8]. In an U-net the output is an image with the same dimension of the input, but with one channel. The encoder path is a typical CNN, where each downsampling step doubles the number of feature channels. But what makes this architecture unique is the decoder path, where each upsampling step input is a concatenation of the output of the previous step with the output of the corresponding (same height) downsampling step. This strategy enables precise localization with a very simple network.

Resnet is a very successful architecture in several visual recognizing tasks [9]. It mitigates the degradation problem that happens when very deep networks starts converging. Instead of learning a direct mapping $H(x) = y$, it learns the residual function $F(x) = H(x) - x$, which can be reframed into $H(x) = F(x) + x = y$, where $F(x)$ is a stack of non-linear layers and x is the identity function(input=output). The formulation of $F(x) + x$ can be implemented by feedforward neural networks with "shortcut connections" (see Figure **??**). Resnet34, specifically, is composed of an initial convolutional layer, 16 blocks of 2 layers and a final fully connected layer.

Unet34 is the idea of using a pretrained Resnet34 model as an Unet encoder path [5]. First, everything from the adaptive pooling onwards is removed, keeping only Resnet backbone. Then we save the output of results of the initial layer, 3rd , 8th, and 14th blocks (of 16 in total). During the upsampling we concat the output of those with the ouputs of upsampling steps. We used Adam optimizer and Binary Cross Entropy with Logits as the loss function.



(a) Learning Rate Optimization

(b) cyclical learning rate policy

(c) superconvergence

Figure 1: colocar aqui

D. Segmentation Experiments and Training

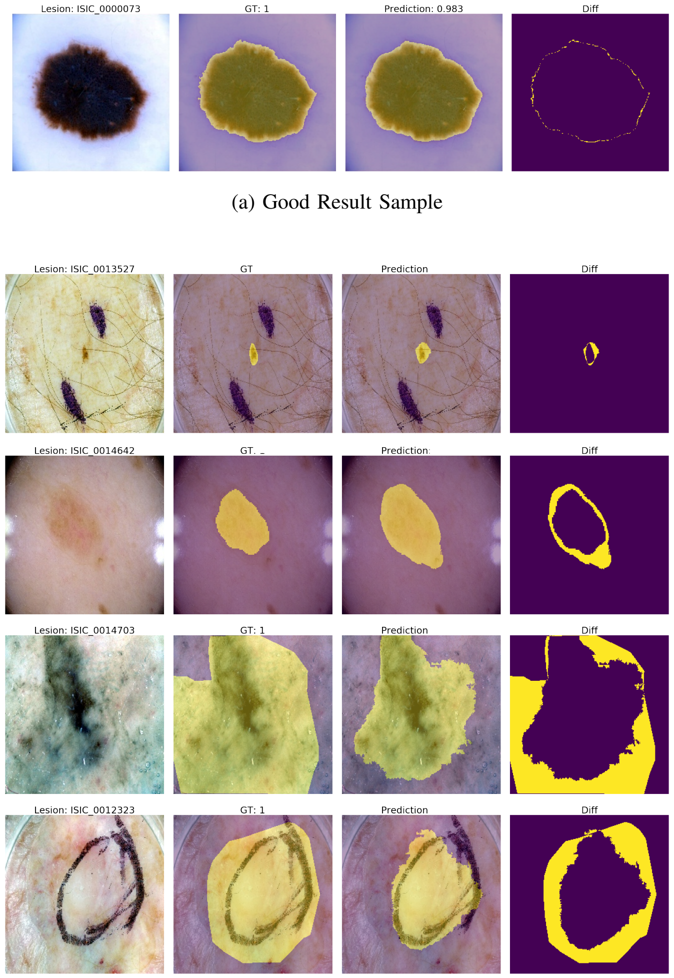
Our training strategy was to first train the model with 128x128 images and transfer this learning to train the same model with images with 256 x 256 images. Initially we thought of using the same strategy to go from 256x256 to 512 x 512 images, but due to GPU memory constraints, we didn't manage to acccomplish this last step.

The training procedure was the same on 128x128 and 256x256:

1) Freeze the first layer group.
2) Define the optimal learning rate with the method proposed by [**?**] and implemented by [5], where one batch is trained with different learning rates, starting at very low and linearly increasing it at every iteration and ploting a chart of the learning rate versus loss (see figure 1).
3) We use the 1 cyclical learning rate policy, also proposed by [**?**], to obtain training convergence in only 30 epochs (what is called superconvergence).
4) Unfreeze the model, keeping only the batch normalization layers frozen, and repeat steps 2 and 3.

(explain 3-fold pretrained with best 256x256)

We have tried to change the loss function to make it more similar to the avaliation criteria. As jaccard is not differentiable, we used the differentiable soft jaccard variation. Despite of that, we couldn't make it work better than the Binary Cross Entropy with Logits loss function.



(a) Good Result Sample

(b) Bad Result Samples

Figure 2: ??.

E. Segmentation Results

The best result within our validation set was obtained using the 3-fold ensemble. It scores 85.39% Jaccard index and 78.43% Threshold Jaccard index (with cut at 65%). Surprisingly, it did not score so well with the online score and given official validation set, scoring 71.4%. The small size of the official validation set and the threshold at 65% may be the reason for that (one change in prediction from 64% or 66% Jaccard makes a huge difference in the average threshold jaccard index). Our best online score with the official validation set was 75.5%.

Visually the best segmentations are almost indentical to the ground truth (see 2a). But we can learn even more from our mistakes. Analysing the worse segmentations there are cases where as non specialists is hard to say if the algorithm was wrong or the ground truth was; there are cases where our algorithm got confused by the pen marker or the glass used by the doctor; and it is clear that in general it doesn't do a good job when the lesion is small relative to the overall image.

III. Lesion Classification

A. Computational Resources and Development Framework

See section II-A.

B. Data and Augmentation

We used "ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection" grand challenge datasets [6], [7]. When we used a pretrained Task1 model, we considered that external data was used. All images were first resized to 224x224 and a color balance copy of the dataset was made. Random transformations on input images to agument the dataset were made: dihedral transformation, rotation (up to 35 degrees), zooming (up to 1.05), fliping and random lightining changes. The official training dataset was then splited in 3-folds of training and validation datasets.

C. Classification Model

We used Resnet50, an architecture which has been very succesful in image classification problems (cite?). We tried using Resnet101 and Resnext101, but due to the lack of computer resources they didn't suit the fast iteration process our research needed. We used the fastai implementation of Resnet50, which has some improvements over the original model proposed by [**?**]. We customized the Resnet50 model pretrained with the Imagenet dataset, by first, removingeverything from the adaptive pooling onwards keeping only Resnet backbone as a feature extractor. Then we plugged a very simple network consisting of:

- BatchNorm1d(1024, momentum=0.1)
- Dropout(p=0.25)
- Linear (input=1024, output=512)
- ReLU
- BatchNorm1d(1024, eps=1e-5, momentum=0.1)
- Dropout(p=0.5)
- Linear (input=512, output=7)
- LogSoftmax

We used Adam optimizer and the negative log likelihood loss function.

D. Classification: Experiments and Training

We didn't use the color balanced dataset copy as using it gave us worse results. The training procedure was similar to Task 1 (II-D), but we used a cosine annealing learning rate strategy (see figure **??**) instead of the one cycle policy (both proposed by [**?**]). We then tried 3 strategies:

*SnapEsem: We saved snapshots of our model during training and then made an ensemble of those snapshots, as proposed by [**?**].*

CropBest: We applied our segmentation algorithm from Task1 to the dataset of Task3, croping the images by the lesion mininum bounding rectangle. We used a pretrained model from SnapEsem and retrained it using the croped images.

KFoldCrop: We repeated CropBest strategy to 3 different folds and ensemble them.

E. Classification Results

The best result within our validation set was obtained using the 3-fold ensemble. It scores 85.39% Jaccard index and 78.43% Threshold Jaccard index (with cut at 65%). Surprisingly, it did not score so well with the online score and given official validation set, scoring 71.4%. The small size of the official validation set and the threshold at 65% may be the reason for that (one change in prediction from 64% or 66% Jaccard makes a huge difference in the average threshold jaccard index). Our best online score with the official validation set was 75.5%.

Visually the best segmentations are almost indentical to the ground truth (see 2a). But we can learn even more from our mistakes. Analysing the worse segmentations there are cases where as non specialists is hard to say if the algorithm was wrong or the ground truth was; there are cases where our algorithm got confused by the pen marker or the glass used by the doctor; and it is clear that in general it doesn't do a good job when the lesion is small relative to the overall image.

IV. Conclusion and Future Work

Neste trabalho, implementamos dois algoritmos para rastreamento visual de objetos em vídeo. O algoritmo KCF apresenta bons resultados de acurácia e robustez. Entretanto, mostramos que seu modelo pode melhorar se incorporar incerteza, o que pode ser feito com um filtro de Kalman. Ao amortecer o resultado do KCF com um filtro de Kalman, tivemos resultados de robustez entre 60 e 90% melhores, com perdas de acurácia menores que 12%, esse resultado serve de inspiração para melhorias na implementação do rastreador KCF da OpenCV.

References

[1] *M. Guillaumin and V. Ferrari, "Large-scale knowledge transfer for object localization in ImageNet," in* 2012 IEEE Conference on Computer Vision and Pattern Recognition. *IEEE, jun 2012. [Online]. Available: https://doi.org/10.1109/cvpr.2012.6248055*

[2] *B. Stewart,* World cancer report 2014. *Lyon, France Geneva, Switzerland: International Agency for Research on Cancer,WHO Press, World Health Organization, 2014.*

[3] *A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks,"* Nature*, vol. 542, no. 7639, pp. 115–118, jan 2017. [Online]. Available: https://doi.org/10.1038/nature21056*

[4] *J. D. L. Fei-Fei, "Where have we been? where are we going?" http://image-net.org/challenges/talks_2017/imagenet_ilsvrc2017_v1.0.pdf, 2017, [Online; accessada 28 de Junho de 2018].*

[5] *J. Howard* et al.*, "fastai," https://github.com/fastai/fastai, 2018.*

[6] *N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. K. Mishra, H. Kittler, and A. Halpern, "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),"* CoRR*, vol. abs/1710.05006, 2017. [Online]. Available: http://arxiv.org/abs/1710.05006*

[7] *P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,"* Sci. Data*, vol. 5, p. 180161, 2018.*

[8] *O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation,"* CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597*

[9] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 5987–5995. [Online]. Available: https://doi.org/10.1109/CVPR.2017.634*