Whack-a-Mole - a study on different approaches on automatic skin lesion segmentation

Vincenzo Claudio Pierro, Giovanni Scafuri, Sergio Siniscalchi

***Abstract*—This study aims to analyze different approaches on automatic skin lesion segmentation, to find the advantages and disadvantages of each one and to develop an algorithm capable of reaching performances comparable with the best found in the literature. We designed a pipeline to preprocess the images and to remove hair and three different algorithms, based on connected components and Watershed, segmentation using thresholding and a U-Net implementation. We finally developed a GUI to test those algorithms.**

I. Introduction

SKIN cancer is one of the most commonly occurring cancer in both men and women, with an increasing incidence over the years. However, its 5-years survival rate is 95% if recognized and treated early but can be fatal if it starts to advance and spread throughout the body, creating metastasis. Moles, and skin lesions more generally, can be symptoms of such disease and dermatologist differentiate benign from malignant lesion through dermoscopy, a non-invasive exam that allows the clinician to study even the skin lesions that are not visible with the naked eye. Each skin lesion is classified manually through the ABCDE principle, that considers the asymmetry, the borders, the colour, the diameter and the evolution of the skin lesion over time [1]. A CAD (Computer-Aided Diagnosis) system could automatically segment the skin lesions, helping the dermatologist in detecting skin cancer earlier and with greater accuracy. Such systems can also go beyond the segmentation task and be able to classify skin lesions in benign and malignant [2]. During our research phase in the literature, we discovered that this problem has been approached in several ways; for example, the ASLM algorithm achieves segmentation through thresholding in the yCbCr and HSV colourspace [2]. Deep CNN approaches are also popular and are based both on U-Net [3] and U-Net variations[4], but also on general-purpose image classification DNN such as YOLO [5]. Finally, we also documented ourselves on methods to solve common problems such as hair removal [6]. Thus, we decided to implement different approaches, and eventually combine them, to compare their performances and analyze strong and weak points.

First, we designed a preprocessing pipeline to remove hair and segment the image to reduce noise. Then we implemented three different algorithms: COCOAW, based on connected components and watershed, IDC, based on clustering on intensity domain, and MoleNet, an implementation of U-Net that we trained for skin lesion segmentation. Finally, we developed a web interface to test the different algorithms more interactively.

As for the data we used, we chose the ISIC 2017 and ISIC 2018 datasets, publically available online and containing both the skin lesion images and their segmentation. The final size of our dataset was of 4400 images, that we divided into a training set of 3600 images, a validation set of 400 images and a test set of 400 images.

The organization of the document is the following: in section II we will describe our preprocessing pipeline and our proposed approaches, in section III we will present our results, section IV will contain conclusions and section V the possible developments.

II. Methods

In this section, we will describe the different procedures we used to approach this problem. We developed our solutions using Python 3.7, the Python version of OpenCV and NumPy. We also used Tensorflow and Keras to implement, train and test our convolutional neural network. The organization of the section is the following: in subsection A we will talk about the preprocessing pipeline, in subsection B the COCOAW algorithm, in subsection C the IDC algorithm, in subsection D the MoleNet implementation and, finally, in subsection E we will briefly describe the user interface to test the algorithms.

A. Preprocessing pipeline

During our research phase, we noticed that a problem often addressed in any approach was image preprocessing to remove hair, to improve segmentation results. Checking our dataset, we noticed that hair covered many skin lesions; so we designed a pipeline to remove them, but also other useless details of the skin.

As for hair removal, we implemented two different approaches: in the first one, we converted the images to grayscale, to enhance the dark colour of hair, then applied a median blur to reduce noise while keeping the edges. Then, we tried the Canny edges algorithm and discovered it was able to highlight just hair contours. After that, we applied a dilation to better determine the hair surface, finally obtaining a hair mask. Then, considering the image as a matrix, we applied a linear interpolation on the rows and the columns and then we averaged the two results.

The second approach also starts with converting the colourspace of the image to grayscale; then we applied a black top-hat transformation using a cross structuring element. The idea behind this choice was to detect the intersection

Acknowledgement: this template was made starting from the *Template for LNT Seminar Report* available on https://www.overleaf.com/

between hairs but not the skin lesion contours. We compared that structuring element with others, noticing that it gave the best performances. Applying thresholding we obtained the hair mask and used the OpenCV FMM inpaint algorithm to infer the intensity value of the mask points using the surrounding pixels.
We present the results of those two methods in figure 1. Comparing the results of the two approaches on several images, we decided to use the second one; the last step of the preprocessing pipeline is to apply the mean-shift algorithm, to exploit local homogenization and make the image more uniform while keeping the edges.
After seeing the results, we noticed that our method was able to effectively remove noise and imperfections in the image. We also tried to increase the contrast of the image to make the skin lesions more evident using the gamma intensity transformations or histogram equalization. However, this left unchanged the images with dark lesions and performed poorly in the ones where the colour of the moles was skin-like, making them more difficult to recognize.

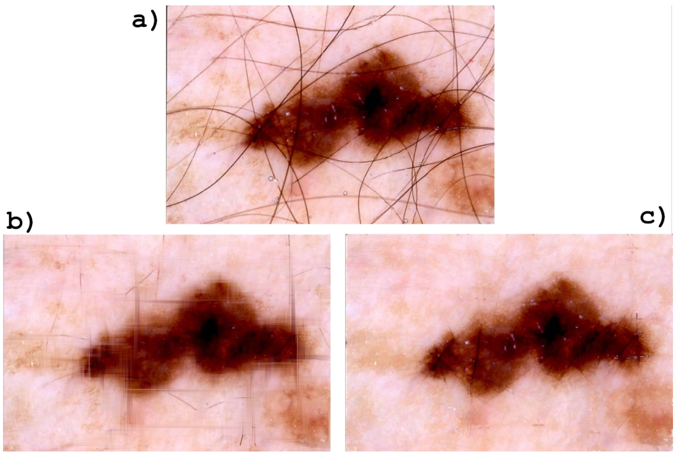

Fig. 1. *a) the original image* b) the result of the first preprocessing pipeline *c) the result of the second preprocessing pipeline*

B. COCOAW

One of the algorithms we developed is the COCOAW (COnnected COmponents And Watershed) algorithm; the idea behind it is to start from the binary image, find the connected components and use them to create the markers for a marker-controlled watershed.
After preprocessing the image using the pipeline described in subsection A, the first step is to convert it to the HSV colourspace and take the saturation channel, exploiting the fact that skin lesions have a lower saturation value because of their darker colour. Then, we applied thresholding using Otsu's method to obtain a binary image. Since images in our dataset have frequently black margins, we created a mask to remove those. Starting from the preprocessed image, we converted it to grayscale then applying a low threshold, to take just those darker parts and removing them through a bitwise and operation. We show the result of this operation in figure 2.

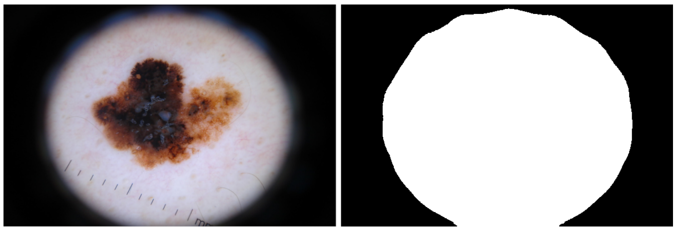

Fig. 2. The original image (on the left) and the inverted margin mask (on the right)

After that, we found the connected components, filtering the ones with an area too small that could only be noise and the ones with a circularity near one, that have the highest probability of being stickers. Then, we applied the distance transform to find the internal markers and dilation to find the external markers. The last step was to use those markers to apply the marker-controlled watershed algorithm to find the lesion contours. We evaluated the watershed algorithm on the original image, on the image preprocessed using our pipeline, on a smoothed version of the image using the ASF (Alterinating Sequential Filter) and on the gradient of the image, obtained applying the Laplacian filter. We obtained the best performances on the preprocessed version of the image, that was our final choice for this algorithm.

C. IDC

An alternative to the COCOAW approach is the IDC, also known as Intensity Domain Clustering.
By doing some experiments, we found out that Otsu and triangle thresholds could not perform well in most of the images we used, while the binary (and binary inverted) threshold would do a better job; however, it had the problem of being unable to automatically define a threshold value.
To overcome this limit, we defined a clustering approach: it makes use of the K-Means algorithm to find four different clusters based on the intensity levels of the pixels, which are then used to define the proper threshold to apply.
In particular, we tried to do this in two different ways: one based on the BGR version of the preprocessed image and another based on its greyscale version.
In the first case, we tried to find the main colours of the BGR image and convert them into their respective grey levels, by using the following formula:

$$grey = 0.21 * R + 0.71 * G + 0.07 * B$$

However, this method didn't work well for all the images, since some of them had coloured stickers that would affect the clustering result.
In the second case, instead, we tried to find the main grey levels of the greyscale image: because this solution was less sensitive to the stickers, we decided to use it for the approach presented.
After the clustering and threshold processes, we find the connected components of the image, filtering the margins and the noise using the same technique adopted for the COCOAW

algorithm. Finally, only the largest connected component is taken.

D. MoleNet

For our final approach, we tried to implement and train a Deep Convolutional Neural Network as the effectiveness of this method was proven by several articles such as [3], [4], and [5]. We decided to use the U-Net architecture as this network is specialized for image segmentation tasks. Its features are described in more details in [3]. Our dataset contains 4400 image/mask pairs coming from ISIC 2017 and ISIC 2018 challenges, divided into a training set of 3600 instances and a validation set and a test set of 400 images, resized to 400 x 600 pixels as suggested by [4]. Figure 3 describes the structure of our network.
After several attempts in training it, we found that the best configuration involved:

- Using Stochastic Gradient Descent as the optimizer.
- Using binary cross-entropy as the loss function, as suggested by [4].
- Using the sigmoid activation function instead of the softmax one (as proposed in [3]).

Since the network gives as output a matrix with floating-point values between 0 and 1, we rescaled it to the 0 to 255 range and applied thresholding using the Otsu's method to obtain the binary mask.
The network was trained both on raw images and images transformed using the preprocessing pipeline described above and we discovered that using the preprocessed images increased the overall performances.

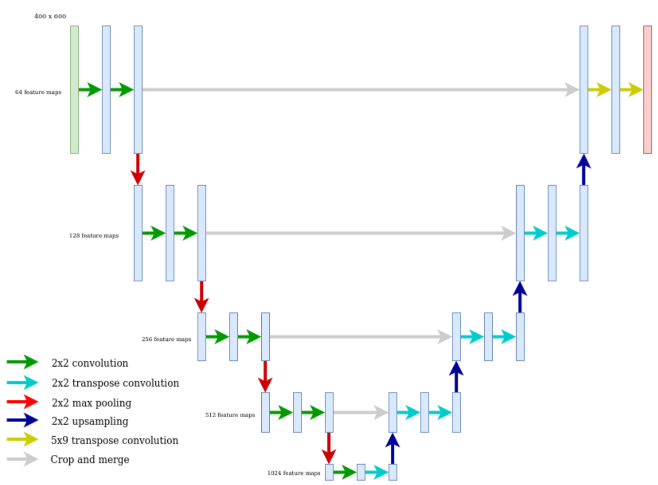

Fig. 3. The MoleNet architecture

E. Graphical User Interface

At the end of our project, we decided to develop a web-based graphical user interface to test and compare the results of our algorithms more quickly and interactively.
This tool allows you to choose an image and eventually its ground truth, choose what kind of operation to perform

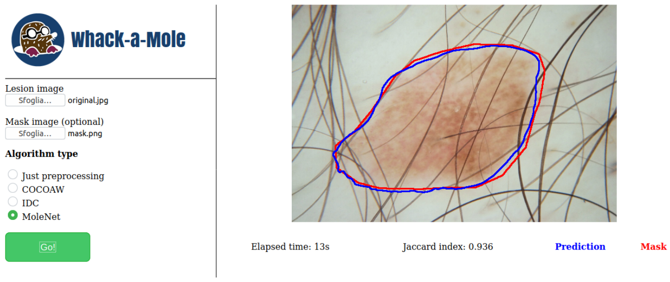

Fig. 4. The graphical user interface

(preprocessing, COCOAW, IDC, MoleNet), visualize the prediction compared to the real result and obtain numerical information (Jaccard index, elapsed time) about it. The GUI is presented in figure 4.

III. Results

We tested all the algorithms on the same dataset made up of 4400 images from ISIC 2017 and ISIC 2018. Image 5 shows the different outputs obtained from the same image and compares them with the real result.
The metric used to evaluate the performances was the Jaccard index, also known as IoU (intersection over union) index. The Jaccard coefficient measures the similarity between finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets. In our case, the real mask and the predicted one are the sample sets, the intersection and union operations are related to their white pixels, so the Jaccard index represents how much they overlap.

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

After the experiments, we were able to understand the strong and weak points of our approaches:

- COCOAW was proven to be robust to dark margins and able to identify more regular mask borders but is strongly affected by the presence of stickers to mark the skin lesions;
- IDC, on the other hand, is robust to stickers too, but usually underestimates lesion area and the predicted contour is more irregular;
- MoleNet was able to overcome the problems of the two algorithms without any strong drawbacks.

Our reference performance is the one reached by the YOLO net in [5] (0.79 on the ISIC dataset) and our results are described in table I. The neural network outperformed the other two algorithms reaching an average Jaccard index comparable with YOLO.

TABLE I
PERFORMACES ON TEST SET

Algorithm	Average Jaccard Index
YOLOv3 [5]	0.79
C-UNet [4]	0.77
MoleNet	0.76
COCOAW	0.62
IDC	0.52

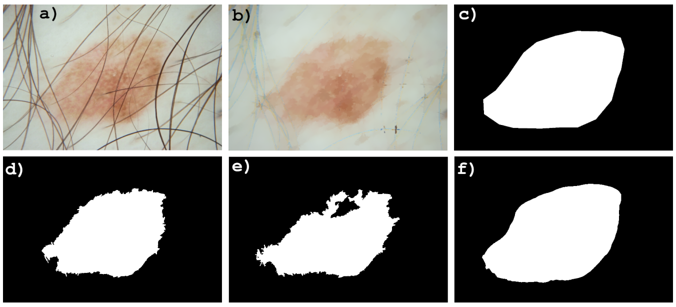

Fig. 5. *a) the original image* b) the preprocessed image *c) the ground truth mask* d) the COCOAW prediction *e) the IDC prediction* f) the MoleNet prediction

IV. Conclusions

This project allowed us to work with the techniques and algorithms studied during the course to solve an actual real-life problem, understanding their advantages and drawbacks in various situations.

In the first part of our study, we searched in the literature for systems to solve problems similar to ours, to understand what steps were proven most useful for this kind of application. Then, we presented three approaches, the ones that performed best among all our attempts, based on three different pipelines and we ultimately managed to obtain results comparable with the most recent and best ones found in the literature.

V. Further work

As possible future developments, we could concentrate on further improving our U-Net implementation by making it more complex and expanding the dataset used to train it.

This system could also be the first part of a classificator capable of automatically recognize malignant skin lesions from benign ones.

References

[1] M.E. Vestergaard, P. Macaskill, P.E. Holt and S.W. Menzies Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting. *British Journal of Dermatology*, vol. 159, no. 3, pp. 669-676, Sept. 2008.

[2] A. Pennisi, D.D. Bloisia, D. Nardia, A.R. Giampetruzzi, C. Mondinoc, A. Facchiano Skin Lesion Image Segmentation Using Delaunay Triangulation for Melanoma Detection. *Computerized Medical Imaging and Graphics*, vol. 52, May 2016.

[3] O. Ronneberge, P. Fischer, T. Brox U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, vol. 1055, 2015.

[4] J. Wu, E.Z. Chen, R. Rong, X. Li, D. Xu, H. Jiang Skin Lesion Segmentation with C-UNet.*2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).*

[5] H.M. Ünver, E. Ayan Skin Lesion Segmentation in Dermoscopic Images with Combination of YOLO and GrabCut Algorithm. *Diagnostics*. vol. 9, pp. 79, Jul. 2019.

[6] T.S. Cho, W.T. Freeman, H. Tsao A reliable skin mole localization scheme. *2007 IEEE 11th International Conference on Computer Vision.*