A Morphological Convolutional Autoencoder for Segmenting Pigmented Skin Lesions

Alberto M. Ceballos-Arroyo, Sergio Robles-Serrano, German Sanchez-Torres

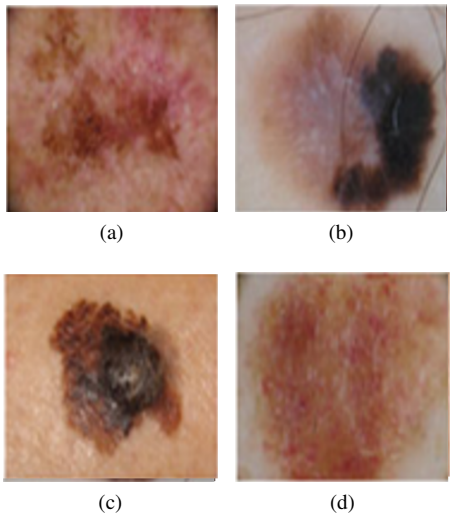
***Abstract*—Pigmented skin lesions can be benign or malignant, such as skin cancer. Malignant melanoma is the most dangerous kind of skin cancer, and it causes 75% of related deaths. Early diagnosis can result in preserving the lives of most patients, but in most countries, the analysis is done based on a manual inspection by specialists, which can be inaccurate. Digital dermatoscopy is a non-invasive methodology that allows in-vivo evaluation of different skin conditions at the macroscopic level using histological features, and it can be automatised by means of computational tools. In this paper, we propose a U-Net-based architecture including morphological layers, called Morpho-U-Net, for the automatic segmentation of skin lesions. The output of this architecture consists of binary masks that can be used to separate the lesions from the rest of the dermoscopy image and can serve as input for either human-based analysis or other algorithms for skin lesion classification. Our strategy was tested on the ISIC 2017, ISIC 2018, and NH2 data-sets. Our experiments showed that our work is above several state-of-the-art proposals, with an average thresholded Jaccard score of 0.93. Furthermore, we believe our architecture could be used as the basis for addressing other image segmentation problems.**

***Index Terms*—Melanoma, visual computing, machine learning, auto-encoders, deep learning.**

I. Introduction

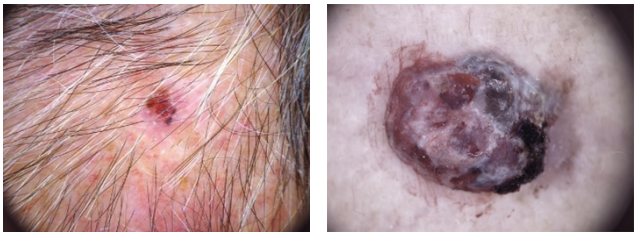
Melanoma is a type of skin cancer which constitutes one of the main causes of skin cancer mortality. Every year, melanoma causes up to 66.000 deaths worldwide [1], [2], [3], including around 75% of skin cancer-related deaths in Colombia [4]. Autier and Doré [5] categorize melanoma as a highly dangerous disease. Furthermore, it is one of the illnesses to which humans are the most exposed, since it is often caused by exposure to solar radiation [6]. This is the case for regions where sunlight incidence is high, such as the Caribbean coast of Colombia. Most manifestations of melanomas are pigmented skin lesions, as observed in Figure 1, so they are often detected and diagnosed by experts through the visual analysis of dermoscopy images [7].

Albeit common, melanoma is not the only type of skin cancer. Basal cell carcinomas also exist, which appear as sores and other skin lesions. Health researchers have studied basal cell carcinomas and determined that they are the type of cancer that is both the most prevalent and the easiest to treat [8]. Other relevant skin lesions include squamous cell carcinomas (Bowen's disease), benign keratosis, benign fibrous histiocytoma, melanocytic nevus, and benign vascular lesions in general [3].



(a) (b) (c) (d)

Fig. 1: Pigmented skin lesions indicating the presence of melanomas [1].



(a) (b)

Fig. 2: (a) A basal cell carcinoma case, (b) A melanoma case [1].

Mortality rates for melanoma are much higher than those related to other skin lesions. However, they often manifest themselves similarly on the skin (see Figure 2). This makes it harder for medical specialists to detect skin cancer, forcing them to resort to slower, costlier procedures (e.g., molecular analysis of the affected tissue) in order to classify skin lesions [9]. In turn, an incorrectly classified skin lesion results in incorrect treatment and, sometimes, in patient death. In contrast, the benefits of early detection allow most patients to outlive melanoma if it is detected in time [10].

The importance of correctly detecting skin lesions is such that many competitions have been organized in order to motivate the development of algorithms capable of automatically discriminating melanomas from other lesions. One such example is the ISIC 2018 competition [1]. The goal

Manuscript received February 03, 2020; revised May 01, 2020.

This work was supported by the Universidad del Magdalena, Santa Marta, Colombia.

G. Sánchez-Torres is with Systems and Computing Research Group, Faculty of Engineering, Universidad del Magdalena, Santa Marta, Colombia e-mail: gsanchez@unimagdalena.edu.co

A. Ceballos-Arroyo is with Systems and Computing Research Group, Faculty of Engineering, Universidad del Magdalena, Santa Marta, Colombia e-mail: albertoceballosma@unimagdalena.edu.co

S. Robles-Serrano is with Systems and Computing Research Group, Faculty of Engineering, Universidad del Magdalena, Santa Marta, Colombia e-mail: sergioroblesas@unimagdalena.edu.co

of this challenge was to correctly segment and classify the HAM10000 data-set, which comprises 10.015 labeled and anonymized dermoscopy images of skin lesions. Of these, 1.099 correspond to melanoma lesions, while the rest belong to the categories described in Figure 3 [11]. Furthermore, 2.600 of the images in the HAM10000 dataset have ground-truth segmentation masks, where pixels belonging to the lesion (foreground) are labeled as 1, and healthy skin pixels (background) are labeled as 0.

Another dataset which has been widely employed in the literature is the PH2 dataset [12], which comprises 200 images divided into three categories: common lesions, atypical nevi, and melanoma. Thanks to these datasets, it is considerably easier to develop algorithms for segmenting skin lesions, as there are no ethical considerations with respect to confidential patient information. Although no datasets like HAM10000 and PH2 have been released in Colombia, the amount of data in publicly available datasets is enough for both developing and testing new algorithms.

Equally relevant for the analysis of pigmented skin lesions is the description of the image acquisition method. Both datasets we employ in this work consist of images acquired by means of dermoscopy imaging. There exist two main types of dermoscopy devices used in practice [13]: immersion fluid dermoscopes and cross-polarized light dermoscopes. Most dermoscopes can be easily attached to digital cameras, and although the physical principles are outside the scope of this work, both categories capture images unaffected by most bodily fluids and allow the user to zoom-in and visualize many structures in skin lesions, such as globules, streaks, or pigment networks. Furthermore, the acquisition system greatly enhances contrast between the lesion and healthy skin, as well as between the lesion and itself and healthy skin. A prevalent issue, however, is that hairs, air bubbles, or ruler markings can make it harder for computer vision algorithms to correctly segment the images if no preprocessing is carried out [13].

One of the most employed criterions today for early melanoma detection is the *ABCDE* rule, which was designed for humans to analyze skin lesions based on several visual cues: (A) Asymmetry, (B) irregularity of borders, (C) presence of specific colors, (D) lesion shape and size, and (E) evaluation of the lesion evolution over time [14]. In Colombia and most of Latin America, many skin cancer studies still rely on similar criterions [15][16][17]. Since automatization is not yet widespread in the Colombian health sector, it is of great interest to develop tools for automatizing this task in the local context.

Many systems for computer-aided skin lesion diagnosis exist in the literature [18][19][20], and some have greatly facilitated detection of skin cancer by dermatologists. Some challenges remain, however, in that dermoscopy images often have issues that make it difficult for computers to discriminate lesions from healthy skin. These include presence of hair, variations in skin tonality among patients, reflections, and oil bubbles, among other noise sources [1][18].

There exist multiple image segmentation approaches in the literature [18][21][22][23]. Traditional segmentation methods are based on typical digital image processing techniques and have proven useful for problems with partly controlled conditions such as melanoma segmentation. More modern deep learning approaches, however, have been proven to be more efficient in recent melanoma segmentation challenges, as they can fit extremely complex functions [24]. Even among deep learning techniques, there exist several approaches for image segmentation, of which autoencoders are the most efficient. The reason is they require relatively small training sets and can be trained (and deployed) faster than other architectures such as the FCN [25].

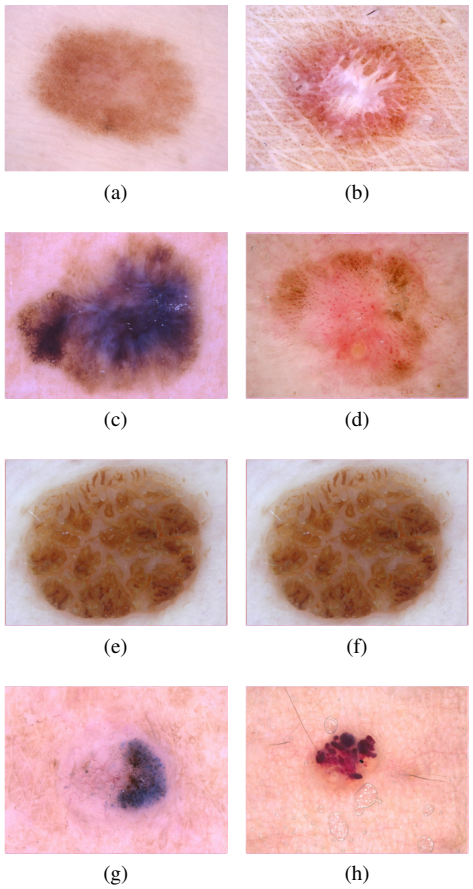


Fig. 3: Disease classification of pigmented skin lesions within dermoscopic images, (a) Nervus, (b) Dermatofibroma, (c) Melanoma, (d) Pigmented Bowen's, (e) Pigmented benign keratoses, (f) Basal cell carcinoma, and (g) Vascular lesion [1].

In this paper, we propose a U-Net-based, morphological autoencoder deep learning architecture for segmenting skin lesions from healthy skin. The output from this architecture consists of binary masks that can be used to separate the lesions from the rest of the dermoscopy images. Such masks can serve as input for either human-based analysis or other algorithms for skin lesion classification.

The paper is organized as follows: in Section 2, we explore previous work on pigmented skin lesion segmentation. Section 3 centers around the proposed methodology. Sections 4 comprises our results and discussion, while in Section 5 we provide some conclusions and insight for future work.

II. Previous Work

Based on our review of previous work, methods for cutaneous skin lesion segmentation can be categorized into one of two categories: traditional segmentation methods and deep

learning-based segmentation methods. While the latter have become more popular recently, the traditional approaches are still employed with good results by many researchers.

A. Traditional segmentation methods

This category comprises all methods based on digital image processing techniques such as thresholding, mathematical morphology, graph-cuts, and super-pixel merging, among others. Despite varying considerably in the way segmentation is carried out, they do not require using any training data and have reduced execution times, making them stand out when performance is critical [18].

Traditional segmentation methods achieve high segmentation accuracy when the conditions are appropriate. However, dermatologists currently do not have a standardized acquisition protocol, meaning they take pictures with different resolutions, zoom-levels, and illumination conditions [26]. Thus, in most cases, there is not enough contrast between the lesions and healthy skin in the input image. In addition, body hair might occlude some details. Such conditions make it harder to correctly segment image regions, causing many of the traditional methods to fail unless the image is properly preprocessed [27]. For instance, Bi et al. [27] used hair templates at 12 orientations in order to detect hair and then replaced the associated pixels with non-hair pixels derived from neighboring pixels.

Patiño et al. [18] developed a segmentation method which does not require preprocessing steps such as hair removal or lighting correction. The method is based on super pixel segmentation, which means they first over-segment the image into super pixels (sets of pixels grouped into units with similar features) prior to merging them based on some criteria. They employed the SLIC (Simple Linear Iterative Clustering) super pixel segmentation algorithm. SLIC clusters pixels based on the k-means algorithm, where the considered features are color intensities weighted by an alpha factor. The authors segmented RGB dermoscopic images with parameter k set as 400 empirically. The SLIC output is a label image, where each label corresponds to one super pixel and for each label there is a list of properties derived from the RGB intensity values, such as mean color, total color, and the pixel count. Such properties are then employed to merge the super pixels into two regions: background and foreground. The method was evaluated on the ISIC 2017 [24] and PH2 [12] datasets with results comparable to several state-of-the-art proposals, albeit the algorithm failed to correctly segment lesions comprised by two or more unconnected regions.

Lezoray et al. [28] developed a multispectral graph-based skin lesion segmentation method. For preprocessing, the method can detect hair on infrared images (hair is darker than the surrounding zones on IR images) and remove it by inpainting on the visible spectrum images. Additionally, light reflection areas are detected with a comparison between the blue intensity level for each pixel and the average blue intensity in its neighborhood. Then, the pixel is inpainted with the proposed inpainting method. After preprocessing the image, the authors over-segment the image into superpixels using the Eikonal-based Region Growing Clustering algorithm (ERGC). They then use a graph-based regularization approach for optimizing the boundaries between super-pixels first, and then between single pixels, making segmentation more precise. The algorithm was evaluated on a set of multispectral images acquired from the Grenoble Hospital, and it achieved a sensitivity of 0.9362, a specificity of 0.9854, and an error probability of 0.0279.

B. Deep learning-based segmentation methods

Although there exist several approaches for deep learning, the most common of them is to employ Artificial Neural Networks (ANNs) [29][30]. ANNs are supervised machine learning algorithms consisting of several layers of nodes called artificial neurons. Nodes are connected by weights: signal-transmitting connections which allow the network to model arbitrary functions. Except for input nodes, ANN nodes contain activation functions for processing incoming data. The layers can be classifying into three categories:

- *Input layer:* consists of input nodes representing input data and dimensionality.
- *Hidden layers:* they consist of nodes whose activation functions receive high-dimensional input and reduce dimensionality.
- *Output layer:* consists of one or more nodes representing the target function.

ANNs can learn patterns by processing labeled data. Each training iteration adjusts weight values through a process comprised by two steps: forward-propagation and back-propagation. In forward propagation, the input is run through each layer l. Each layer is comprised by a linearity and an (optional) activation function g through which the data (either input data X or the output of a previous layer $A^{[l-1]}$) is passed prior to being fed to the next layer, thus:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]} \tag{1}$$

$$A^{[l]} = g(Z^{[l]}) \tag{2}$$

Where $W^{[l]}$ is the weight tensor associated to the connection between layer l and layer $l-1$, $b^{[l]}$ is the bias term, $A^{[l-1]}$ is the activation tensor output by layer $l-1$, and $g()$ is a non-linear activation function.

The final step of forward-propagation is to calculate prediction loss. This usually done by evaluating a given loss function on the activations output by the last layer. In binary segmentation and classification problems, the activation function for the last layer is usually a logistic function which outputs values between 0 and 1 for the prediction $\hat{y}$ (Eq. 3) and the loss function is the binary cross-entropy function L (Eq. 4). The total cost J is the average of the loss function L evaluated on the full dataset (Eq. 5).

$$\hat{y}^{(i)} = \frac{1}{1 + e^{-z^{(i)}}} \tag{3}$$

$$L^{(i)} = -(y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})) \tag{4}$$

$$J = \frac{1}{M} \sum_{i=1}^{M} L(y^{(i)} - \hat{y}^{(i)}) \tag{5}$$

For a training set with M training examples, where $z^{(i)}$ is the output of function (Eq. 1) for the last hidden layer, $\hat{y}^{(i)}$

is the predicted value, and $y^{(i)}$ is the ground truth value, for training example i. In the case of image segmentation, $\hat{y}^{(i)}$ is a segmentation mask for image i and loss L is the average of function (Eq. 5) for the pixels in the predicted segmentation mask and the ground truth segmentation mask.

Back-propagation consists in minimizing prediction loss as a function of the weight and bias values by means of optimization algorithms. One of the most basic optimization methods is batch gradient descent, where the weights are updated once for every training iteration (epoch) based on the whole dataset, as in (Eq. 6). Once the training process finishes, the ANNs can predict labels or calculate regressions for previously unseen data.

In batch gradient descent, the weights and bias terms are updated as in (Eq. 6) and (Eq. 7).

$$W^{next} = W - \delta \Delta W \tag{6}$$

$$b^{next} = b - \delta \Delta b \tag{7}$$

Where W^{next} is the updated weight tensor, W is the current tensor value, b^{next} is the updated bias term, b is the current bias term, δ is the learning rate (a hyperparameter of the network that determines how fast weights should be updated), and ΔW and Δb are the gradients for W and b respectively.

Due to the lack of computational resources and sufficiently solid backpropagation methods, researchers believed there was no benefit in going beyond a couple hidden layers. As they overcame these obstacles, they now propose deep learning (DL) techniques [31]. Deep neural networks consist of a succession of many hidden layers, allowing us to model computationally complex problems with relative ease [31].

The deep networks which are employed the most for image-based deep learning are Convolutional Neural Networks (CNNs), whose multi-layered structure stores *deep* visual features in the form of convolutional filters. CNNs, as well as other kinds of deep architectures, can be employed for different tasks in the pigmented skin lesion characterization pipeline, including lesion segmentation, dermoscopic feature detection, and disease classification [20][26][32].

The structure of layers in convolutional neural networks is different from that of traditional neural networks. Instead of a having hidden layer nodes, the hidden layers comprise trainable convolutional filters, which are applied similarly to traditional edge detection filters, and non-trainable pooling filters which can be employed for reducing dimensionality in the height and width dimensions when going deeper into the network. Such filters are usually not fully connected, meaning that each filter handles localized information, greatly reducing computational costs and making it easier for the neural network to learn relevant features. In this way, the input data X is transformed and stacked into differently shaped volumes of data when passed through the layers. One of the earliest implementations of a convolutional neural network is the *Le-Net 5*, which was employed for document recognition and includes convolutional layers, pooling layers, and standard fully-connected layers [33].

The result of a applying a convolutional layer on the input data is described thus:

$$A^{[l]} = g(W^{[l]} * X + b^{[l]}) \tag{8}$$

Where $*$ is the convolution operator, $g(.)$ is a non-linear activation function, W is the weight tensor, b is the bias, and A is the output of the layer l. The convolution operation is also based on the stride s and padding p parameters, as well as kernel size k, which all determine the shape the data takes after being passed through a convolutional layer. The relationship between an input channel of height and width i, and the height and width o of the output, is defined by:

$$o = \left| \frac{i + 2p - k}{s} \right| + 1 \tag{9}$$

The max-pooling operation is also defined based on stride s and padding p. It consists in passing a sliding window over each channel of the input volume such that the pixels in each channel of the output volume are the maximum value in each neighborhood specified by s, p, and k, thus:

$$o = \left| \frac{i - k}{s} \right| + 1 \tag{10}$$

Another important operation in our architecture is the transposed convolution operation, which maps lower dimensional into bigger higher dimensional content by reverting the order of the forward and backward passes in the standard convolution, thus enabling us to generate segmentation masks at the resolution of our input images.

When carrying out segmentation, deep CNNs with an autoencoder architecture are often employed [26][32]. Autoencoders learn to compress data and to reconstruct it in diverse manners depending on the objective function. In the case of segmentation, they learn to assign a label (e.g., 0: background, 1: foreground) to every pixel in the image [21].

Zhang *et al.* [32] employed a deep autoencoder network for segmenting pigmented skin lesions. The autoencoder comprises multiple convolutional/pooling layers for the encoder part and dropout/up-sampling layers for the decoder part. The authors employed non-linear ELU activation functions between each hidden layer and a soft-max layer for the flattened output layer. Furthermore, the authors employed batch normalization in order to increase training speed. The network was trained with the dataset provided in the 2017 ISBI Challenge on Skin Lesion Analysis Towards Melanoma Detection [24] and achieved 0.95 sensitivity and 0.94 specificity, making it more accurate than architectures such as SegNet [21] and FCN [25].

Attia *et al.* [26] proposed a hybrid autoencoder method where convolutional and recurrent hidden layers are employed and no pre/post processing is necessary. The architecture encodes input images using seven convolutional layers and two max-pooling layers. The encoded feature maps from the convolutional layers are fed into two LSTM layers, which model backward and forward spatial relationships between pixels along the x and y-axes. This is done by processing the input data as flattened non-overlapping patches. The network then extracts the information on deep encoded featured maps to reconstruct the segmentation mask at the same resolution of the input images. A segmentation accuracy of 0.98 and an associated raw Jaccard index of 0.93 were achieved on the dataset provided for the first task of the ISBI 2016 challenge [34].

III. Materials and Methods

The methodology followed in this paper is comprised of several steps. First, preprocessing (resizing and normalization) is carried out on a dataset comprised of the images in the *ISIC 2018* and *PH2* datasets. Then, a modified version of the *U-Net* architecture with morphological layers is trained on the dataset and used to segment the images. Finally, postprocessing in the form of thresholding is carried out to get binary segmentation masks. This process is summarized in Figure 4, while the detailed network architecture is depicted in Figure 6.

A. Preprocessing

Our main preprocessing step consisted of normalizing the dataset by subtracting the mean of the dataset and dividing it by the standard deviation of the dataset for every image, as shown in Eq. (11). This is done as per the literature in order to improve performance by sending only centered, whitened data to the network [29], thus:

$$X_{normed} = \frac{X - \bar{X}}{\sigma_x^2} \tag{11}$$

The other preprocessing steps were to resize every image in the dataset to a resolution of $256x256x3$ pixels and to split the dataset as shown in Table I.

In the literature, there exist several proposals in which the authors carry out significant preprocessing prior to segmenting images with deep learning architectures [35][36], whereas others apply some degree of postprocessing on the resulting binary masks. Although preprocessing does result in improved results for images with defects such as hair occlusion, it must be adequately tuned and might result in reduced accuracy for images with little or no noise. For this reason, we include both preprocessing and postprocessing in the form of morphological layers at both ends of the network. Other than resizing and normalization, the only step of the pipeline outside of the deep learning framework is thresholding, which is further described at the end of this section.

B. Autoencoder architecture

An autoencoder is a type of neural network model comprised of a codifier and a decodifier (see Figure 5). The main task of such networks is to learn a compressed representation of a given input that can be later used to decodify it in a specific way. Just as other neural network architectures have been employed to map non-linear relationships between data and response variables, autoencoders have been employed in the literature as a non-linear method (as opposed to linear ones like Principal Component Analysis) for dimensionality reduction [29].

There exist several kinds of autoencoders, and they can be employed to solve various problems. For instance, it is possible to train an autoencoder to return modified versions of the input data. One such example is the denoising autoencoder, which codifies data (that is, it reduces it to its basic components) and then reconstructs a denoised version of it through the decodifier half of the architecture [37]. A model more relevant to our interests is the *SegNet* architecture, which is capable of learning to segment images pixel-wise when fed (x, y) pairs of input images and their corresponding labeled (sometimes, but not always binary) images [21]. Networks like *SegNet* learn to segment images in their compressed representation and to up-sample (*decode*) them in order to map low-dimensional features into input resolution for pixel-wise segmentation [21]. Our architecture, which we named *Morpho-U-Net* (see Figure 6), is based on the *U-Net* architecture. Like *SegNet*, *U-Net* is an autoencoder architecture which has been employed for many segmentation problems in the literature [38].

C. Weight initialization

All weights in our architecture are initialized using the Glorot uniform initialization [39]. In this initialization scheme, weight values for a given layer are sampled uniformly within $[-a, a]$, with the value of a defined in (Eq. 2), based on the size of the current and previous layers.

$$a = \sqrt{\frac{6}{n_w^{[l-1]} + n_w^{[l]}}} \tag{12}$$

where $n_w^{[l-1]}$ is the number of weights for layer $l-1$ and $n_w^{[l]}$ is the number of weights for layer l.

D. Batch size and epochs

In order to make training feasible on larger datasets, it is usual for researchers to employ a version of gradient descent known as mini-batch gradient descent. Unlike standard gradient descent, mini-batch gradient descent carries out backpropagation independently on several size $m < M$ subsets (known as *batches*) of the training set, where m is the batch size parameter and M is the size of the training sets. Weights are thus updated once for each batch for every training epoch, resulting in lower memory requirements while retaining much of the computational speed associated with running the full training set through the network for every epoch. Furthermore, since weights are updated on each mini-batch instead of on the whole training set, the backpropagation algorithm is less likely to get stuck in local minima [40]. The odds of getting stuck in local minima can be further reduced by shuffling the dataset at the start of every epoch [41]. We set the batch-size hyperparameter $m = 16$ after testing several values, while the number of epochs was set to 100.

E. ADAM optimizer

ADAM stands for Adaptive Moment Estimation. Unlike the basic gradient descent algorithm, computes adaptive learning rates, meaning it is less likely to get stuck on local minima for many epochs. The authors of this method [42] achieved so by storing an exponentially decaying running average of the past gradients $V_{\Delta W}$ (an estimate of the first momentum) and the square of past gradients $S_{\Delta W}$ (an estimate of the second momentum), as per Eqs. 13 and 14.

$$V_{\Delta W} = \beta_1 V_{\Delta W} + (1 - \beta_1)\Delta W \tag{13}$$

$$S_{\Delta W} = \beta_2 S_{\Delta W} + (1 - \beta_2)\Delta W^2 \tag{14}$$

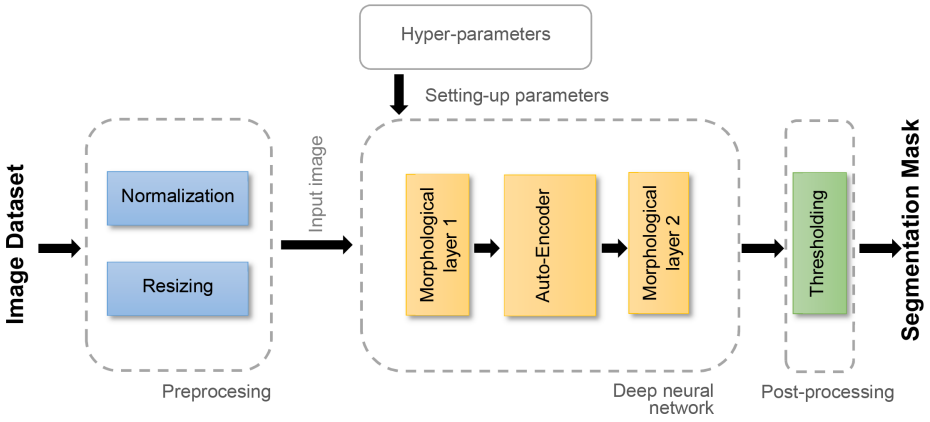


Fig. 4: Proposed methodology for segmentation map generation.

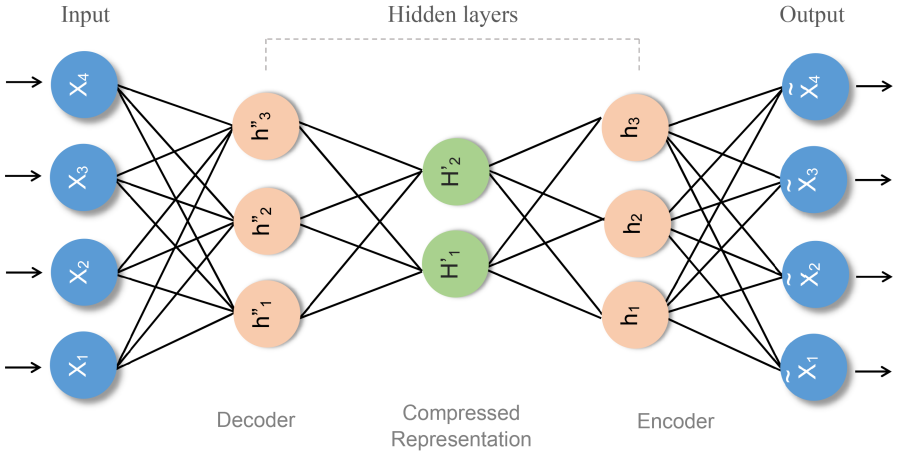


Fig. 5: A typical autoencoder architecture showing the input, hidden, and output layers.

Where β_1 and β_2 are hyperparameters defining the decay of the running averages. We set them as 0.9 and 0.999 respectively as per the literature. Since both running averages are biased toward 0, a bias-corrected version of both is computed in Eqs. 15 and 16.

$$V_{\Delta W}^{corrected} = \frac{V_{\Delta W}}{(1-\beta_1^t)} \tag{15}$$

$$S_{\Delta W}^{corrected} = \frac{S_{\Delta W}}{(1-\beta_2^t)} \tag{16}$$

With t being defined as equal to the current epoch. Finally, the weight tensor W is updated as per Eq. 17.

$$W^{next} = W - \delta \frac{V_{\Delta W}^{corrected}}{\sqrt{S_{\Delta W}^{corrected}} + \varepsilon} \tag{17}$$

Where ε is a constant term we set to $10^{(-8)}$ to avoid division by zero and δ is the learning rate, which we set to 0.0001.

F. Trainable parameters

The full network consists of 21.238.261 parameters (of which 21.228.783 are trainable) divided into two morphological layers prior to the encoder, nine convolutional blocks for the encoder, five transposed convolution blocks for the decoder, one convolutional block at the end of the decoder, and two morphological layers at the end of the network. The detailed neural network architecture is described in Figure 6.

G. Morphological layers 1

The first morphological segment takes in the preprocessed image and processes it with trainable morphological filters as in [43]. First, the image goes through a dilation layer with 3 filters (one for each channel), kernel size 15, stride 1, and same padding. Then, the output goes through an erosion layer with the same parameters resulting in a version of the original image filtered with the learned morphological filters. The dilation and erosion operations are defined in Eqs. 18 and 19, respectively.

$$(I \oplus S_d)(x,y) = \max_{1\leq i\leq a, 1\leq i\leq b} (I(x-i, y-i) + S_d(i,j)) \tag{18}$$

$$(I \ominus S_e)(x,y) = \min_{1\leq i\leq a, 1\leq i\leq b} (I(x+i, y+i) - S_e(i,j)) \tag{19}$$

where S_d and S_e are the dilation and erosion structuring elements, I is the input image or channel, and a and b are the height and width of the structuring elements.

H. Batch Normalization

Usually, data is normalized as a preprocessing step in order to ensure zero-mean and unit variance. However, as the data is processed in the network, these properties change and negatively impact network performance as well as training times. Thus, in [44] proposed *batch normalization* (BN). In

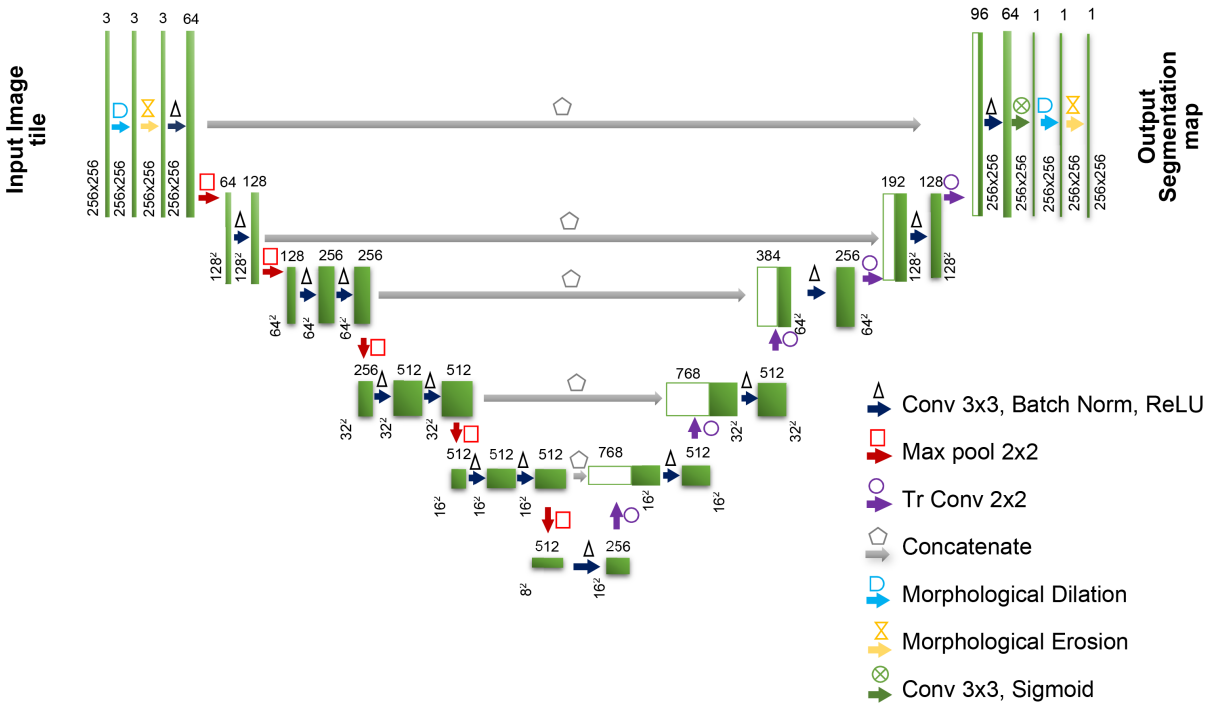


Fig. 6: Morpho-U-Net, the proposed segmentation autoencoder architecture with morphological layers at both ends.

BN, instances are normalized based on the mean and variance of the batch (be it a mini-batch or the whole dataset) they belong to prior to being fed to the activation function, as in Eq. 20.

$$Z^{[l]}_{norm} = \frac{Z^{[l]} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{20}$$

where μ is the mean of the current batch, σ^2 is the variance, and ε is a constant term for avoiding zeroes in the denominator.

The value of Z which is fed to the activation of a given layer is calculated according to Eq. 21.

$$\tilde{Z}^{[l]} = \gamma^{[l]} Z^{[l]}_{norm} + \beta^{[l]} \tag{21}$$

where $\gamma^{[l]}$ and $\beta^{[l]}$ are trainable parameters for layer l and updated by means of gradient descent in the same fashion as the weights and the bias terms.

Some of the positive effects of BN include the reduction of the dependence of gradient descent on the scale of values taken by the data, which facilitates the calculation of gradients. Furthermore, BN has a regularizing effect and reduces the need for employing other regularization techniques.

I. Encoder

After the first set of morphological layers, data goes through the encoder part of the network. The purpose of the encoder is to reduce dimensionality in order to carry out further tasks on the compressed representation. For the encoder, we employ the convolutional layers of the *VGG11* architecture, based on the results described in [22]. The input from the previous set of layers is $256x256x3$ and gets compressed by successive convolutional blocks, starting with a convolutional layer with 64 filters, *same padding*, and a $3x3$ kernel. Batch normalization is applied for every convolutional layer, whose output is received by a RELU activation function and then passed through a $2x2$ max pooling layer with *same* padding in the convolutional block 1. The number of filters for the convolutional layer increases to 128 in convolutional block 2 while the rest of the parameters are kept the same. Convolutional blocks 3 and 4 both have 256 filters, while convolutional blocks 5 to 9 all have 512 filters, with the rest of the parameters being identical to the first two blocks.

J. Decoder

The decoder is comprised of 5 transposed convolution blocks which take in the compressed output of the encoder. Although the decoder is not symmetrical with the encoder regarding the number of blocks, it retains the characteristic U-shape of the U-net architecture in terms of the shape of the data. Transposed convolution block 1 consists of a transposed convolution layer with 256 filters, kernel size $2x2$, and stride 2. The output of this layer is concatenated with the convolutional layer output of convolutional block 8. It is then fed through a convolutional layer with 512 filters, $3x3$ kernel size, and *same* padding. Batch normalization is then applied prior a *RELU* activation function. Transposed convolution block 2 is identical to the first one and is concatenated with convolutional block 6. Transposed convolution blocks 3, 4, and 5 are concatenated with convolutional blocks 4, 2, and 1, respectively. For these blocks, filter size decreases in powers of 2 up 32 in the last transposed convolution layer and 64 in the last convolutional layer. The output of the decoder is then fed to the one last convolutional layer with sigmoid activation which results in a grayscale segmentation mask of size $256x256$.

K. Morphological layer 2

The second morphological layer takes in the grayscale segmentation mask output by the decoder and passes it through a dilation layer with a single filter, kernel size $15x15$, and *same* padding. Finally, the data goes through a last erosion layer with the same parameters as the dilation layer resulting in an improved grayscale segmentation mask of size $256x256$.

L. Postprocessing

The output of the proposed autoencoder architecture is a segmentation mask. However, the masks are not binary and instead include values in the continuous $[0, 1]$ range, sometimes in the form of undesired holes in the segmentation mask. For this reason, we carry out postprocessing, we apply binarization with threshold $t = 0.49$ set empirically in order to obtain the final binary mask.

M. Evaluation

We evaluate our algorithm based on a thresholded (eq. 22) version of the Jaccard index, or Intersection Over Union, criterion (eq. 23), which was proposed in the first task (lesion segmentation) of the *ISIC 2018 challenge* [1], defined as follows:

$$TJAC = \begin{cases} 0, & \text{if } JAC < 0.65. \\ JAC, & \text{otherwise.} \end{cases} \tag{22}$$

$$JAC = \frac{|A \cap B|}{|A \cup B|} \tag{23}$$

Where A is the ground truth segmentation mask, B is the segmentation mask output from a given algorithm, JAC is the raw Jaccard index, and $TJAC$ is the thresholded Jaccard index. The motivation for employing a thresholded variant lies in the fact that segmentations with JAC values lower than 0.65 were deemed as failed segmentation cases by experts, and the value corresponding to correctly segmented sections in these cases should be ignored when computing average efficiency [24].

For the purposes of better comparing our algorithm with other methods in the literature, we also consider several other metrics, which are based on the number of True Negatives (TN), False Negatives (FN), True Positives (TN), and False Positives (FP), resulting from applying the algorithm on an image:

- *Accuracy (ACC):* Measures the proportion of pixels in the image which the algorithm segments correctly.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{24}$$

- *Sensitivity (SE):* Measures the proportion of pixels belonging to the foreground which the algorithm segments correctly.

$$SE = \frac{TP}{TP + FN} \tag{25}$$

- *Specificity (SP):* Measures the proportion of pixels belonging to the background which the algorithm segments correctly.

$$SP = \frac{TN}{TN + FP} \tag{26}$$

- *Dice coefficient (DSC):* Measures similarity between a ground truth mask and a mask generated by a given algorithm.

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{27}$$

IV. Results and Discussion

The proposed autoencoder architecture was trained and validated using $2236/278$ images with their respective ground truth segmentation masks which were provided in the HAM10000 (ISIC 2018 segmentation challenge) and PH2 datasets. We compare the performance of our algorithm with that of several classic architectures in the literature with a test set comprised of 280 images belonging to these datasets. In order to test whether our architecture generalized well to other datasets, we also tested it on the test set provided for the ISIC 2017 segmentation challenge. This also allowed us to compare our architecture with the reported performance of several proposals in the literature that were evaluated on said test set. Performance was measured pixelwise based on the five metrics described before.

A. Experimental setup

We designed the Morpho-U-Net architecture in the Python programming language (3.7 version) using the Keras framework. For the morphological layers, we used the implementation provided in [41]. The network was trained on an i7 workstation with 64 GB RAM and a NVIDIA GTX 1080ti GPU.

B. Results

In the preprocessing steps we split the dataset as shown in Table I. As seen in Tables II and III, Morpho-U-Net outperforms both general-purpose deep learning architectures for image segmentation as well as several of the proposals designed specifically for melanoma segmentation in the literature. While the network has relatively lower accuracy and specificity, it attained the best thresholded Jaccard, Dice and sensitivity scores by a fair margin. Several success cases as well as a failure case can be seen in Figure 7, showing that the network sometimes generates more than one lesion label for a given image. As for the encoder architecture itself, attempting to employ deeper networks such as VGG16 and VGG19 for the encoder did not lead to improved results when compared to the VGG11 encoder. Likewise, as seen in Figure 8, dropout (except when set to 0.1) and traditional regularization techniques were not as effective as employing batch normalization in every layer.

TABLE I: Training, development, and test dataset splits.

Set	Percentage (%)	Images
Training (ISIC2018+NH2)	80%	2236
Development (ISIC2018+NH2)	10%	278
Test 1 (ISIC2018+NH2)	10%	280
Test 2 (ISIC2017)	–	600

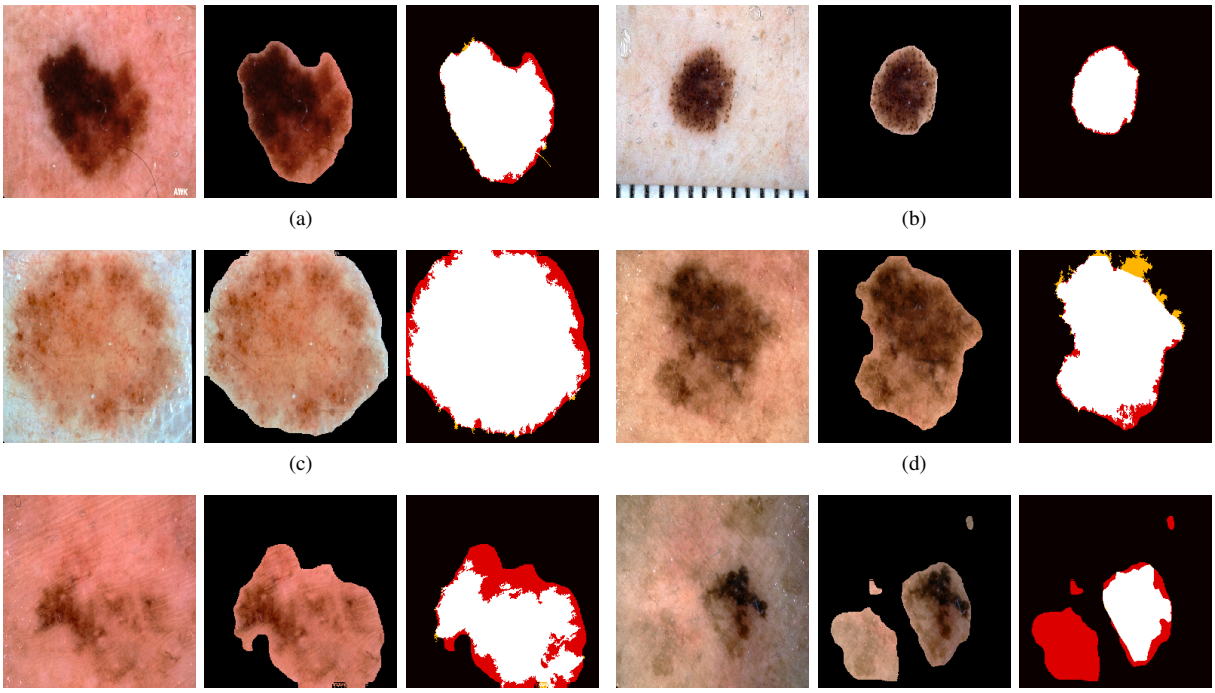


Fig. 7: The result of using the Morpho-U-Net network to segment several skin lesion images. For each case: to the left, the original image; in the center, the segmented portion of the image; to the right, a comparison of the output mask and the ground-truth mask, where white pixels are correctly segmented lesion areas, black pixels are correctly segmented healthy skin areas, red pixels are over-segmented areas, and yellow pixels are under-segmented areas.

TABLE II: Results on test set Training (ISIC2018+NH2), comparing our results with segmentation architectures in the literature.

Method	TJAC	SE	SP	ACC	DSC
Morpho-U-Net (ours)	**0.9153**	**0.9613**	0.8365	**0.9306**	**0.8557**
FCN[25]	0.8056	0.8240	0.9130	0.8459	0.7444
SegNet[21]	0.5525	0.6386	0.8886	0.7001	0.5929
U-Net[38]	0.8605	0.8632	**0.9536**	0.8855	0.8097
TernausNet (VGG11)[22]	0.8931	0.9217	0.9064	0.9180	0.8446
TernausNet (VGG16)	0.7742	0.8063	0.9224	0.8348	0.7330
TernausNet (VGG19)	0.7384	0.7750	0.9329	0.8138	0.7113

TABLE III: Results on test set 2, comparing our proposal with state-of-the-art methods in the literature which were evaluated on the ISIC 2017 set.

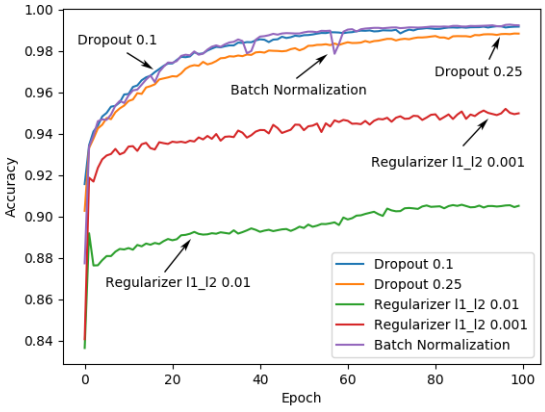
Method	TJAC	SE	SP	ACC	DSC
Morpho-U-Net (ours)	**0.9388**	**0.9708**	0.8749	0.9482	**0.8886**
TernausNet (VGG11) [22]	0.9181	0.9252	0.9230	0.9248	0.8528
Encoder-Decoder CNN [32]	-	0.9500	0.9400	0.9100	-
Ensemble of 10 U-Nets [45]	0.8410	-	-	**0.9510**	-
SLIC segmentation [18]	0.6060	-	-	0.8690	-
Rotation equiv. network [46]	0.7723	0.8540	0.9715	0.9355	0.8560
LIN [20]	0.7530	0.8550	**0.9740**	0.9500	0.8390
Ensemble-A [47]	0.7930	0.8990	0.9500	0.9410	0.8710
Semi-superv. method [48]	0.7980	0.8790	0.9530	0.9430	0.8740

Figure 9 shows the training and validation performance across epochs for several versions of the final architecture, testing different weight initialization schemes. As seen in the figures, Glorot uniform initialization provides the best validation performance when compared to other configurations, albeit by a slight margin. On the other hand, Figure 10 depicts all weight initialization schemes as being similar in terms of the ROC curve. As for the size of the network, Morpho-U-Net is comprised of around 2 million parameters, whereas deep neural network ensembles such as the ones in [49] and [47] amount to more than 20 million parameters. On the other hand, networks such as [46], have a similar number of parameters, but do not achieve similar TJAC scores, whereas more traditional image segmentation methods are of much less computational complexity but fall behind neural networks when taking into account the results in Table 3.

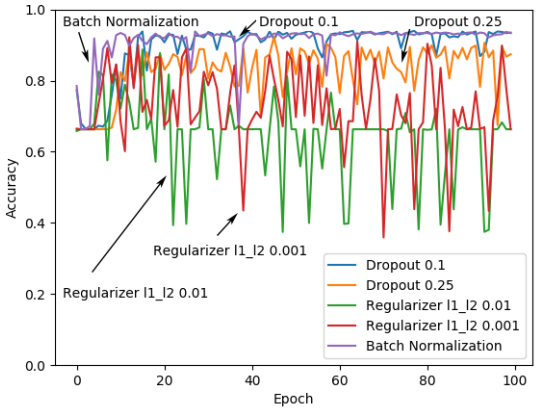
V. Conclusions

We achieved state-of-the-art performance on the melanoma segmentation problem with our relatively small Morpho-U-Net architecture. This was done by combining the ideas behind segmentation architectures such as U-Net and TernausNet with trainable morphological operators. Further improvements were done by adequate selection of weight initializers and by employing batch normalization. We trained our network on the ISIC 2018 and NH2 datasets and reported a 0.9153 TJAC score on the combined ISIC 2018/NH2 test set as well as a 0.9388 TJAC score on the ISIC 2017 test set, outperforming other proposals which were tested on the same dataset. In contrast with work by other researchers, our architecture achieves higher specificity (i.e., there is little under-segmentation) while having somewhat lower specificity (i.e., there is some degree of over-segmentation).

Our main goal for future work is to determine ways to reduce over-segmentation. Also, since our proposal was

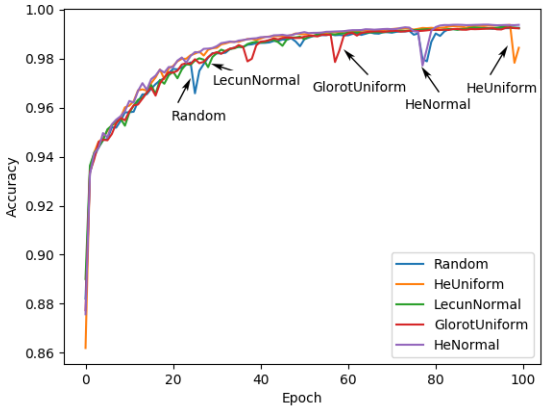


(a)

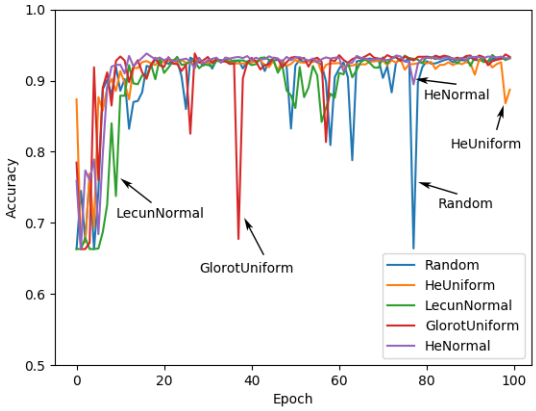


(b)

Fig. 8: a) Training accuracy, and b) Validation accuracy for several configurations of the proposed architecture (dropout, regularization, and batch normalization).



(a)



(b)

Fig. 9: a) Training accuracy, and b) Validation accuracy for the final Morpho-U-Net architecture, using several weight initialization schemes.

not centered in achieving fast computational speeds, we did not test significantly smaller versions of the network, but attempting to use deeper networks such as VGG16 and VGG19 for the autoencoder component lead to worse results, so future work could also focus on testing whether the use of a smaller encoder-decoder architecture would allow for increased segmentation accuracy, and whether the removal of connected components other than the biggest one would lead to improved results in cases where the network generates more than one label. Furthermore, we are interested in testing whether the proposed architecture generalizes well into other segmentation problems. It must be noted, however, that while the employed morphological kernel sizes were very effective for melanoma segmentation, they should be modified in order to employ Morpho-U-Net for the segmentation of differently sized and shaped objects.

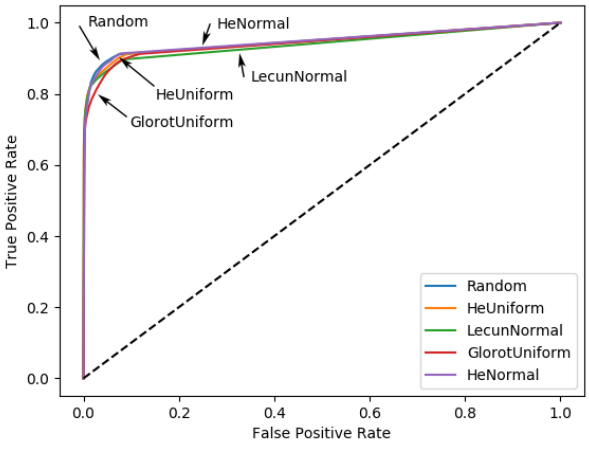


Fig. 10: The ROC curve performance for the Morpho-U-Net architecture, tested with several weight initialization schemes.

VI. Acknowledgments

This research was funded by the Vicerrectoria de Investigación of the Universidad del Magdalena, Colombia.

References

[1] Canfield Scientific, "ISIC Challenge 2018," 2018. [Online]. Available: https://challenge2018.isic-archive.com/

[2] N. H. Matthews, W.-Q. Li, A. Qureshi, M. Weinstock, and E. Cho, "Epidemiology of Melanoma," in *Cutaneous Melanoma: Etiology and Therapy*, W. Ward and J. Farma, Eds. Brisbane: Codon Publications, 2017, ch. 1, pp. 3–22.

[3] R. Gordon, "Skin Cancer: More than Skin Deep," *Clinical Manage-*

ment, vol. 22, no. 12, pp. 574–580, 2009.

[4] F. C. Pozzobon, Á. E. Acosta, and J. S. Castillo, "Cáncer de piel en Colombia: cifras del Instituto Nacional de Cancerología," *Revista de la Asociación Colombiana de Dermatología y Cirugía Dermatológica*, vol. 26, no. 1, pp. 12–17, 2018. [Online]. Available: https://revista.asocolderma.org.co/index.php/asocolderma/article/view/25

[5] P. Autier and J.-F. Doré, "Influence of sun exposures during childhood and during adulthood on melanoma risk," *Int. J. Cancer*, vol. 77, pp. 533–537, 1998.

[6] World Health Organization, "An Overview of the Evidence on Environmental and Occupational Determinants of Cancer," in *International Conference on Environmetial and Occupational Determinants of Cancer*, 2011, pp. 1–7.

[7] S. W. Menzies, K. Westerhoff, H. Rabinovitz, A. W. Kopf, W. H. McCarthy, and B. Katz, "Surface microscopy of pigmented basal cell carcinoma," *Archives of Dermatology*, vol. 136, no. 8, pp. 1012–1016, aug 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10926737

[8] J. M. Llovet, S. Ricci, V. Mazzaferro, P. Hilgard, E. Gane, J.-F. Blanc, A. C. de Oliveira, A. Santoro, J.-L. Raoul, A. Forner, M. Schwartz, C. Porta, S. Zeuzem, L. Bolondi, T. F. Greten, P. R. Galle, J.-F. Seitz, I. Borbath, D. Häussinger, T. Giannaris, M. Shan, M. Moscovici, D. Voliotis, J. Bruix, and SHARP Investigators Study Group, "Sorafenib in advanced hepatocellular carcinoma." *The New England journal of medicine*, vol. 359, no. 4, pp. 378–90, 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18650514

[9] S. A. Leachman, P. B. Cassidy, S. C. Chen, C. Curiel, A. Geller, D. Gareau, G. Pellacani, J. M. Grichnik, J. Malvehy, J. North, S. L. Jacques, T. Petrie, S. Puig, S. M. Swetter, S. Tofte, and M. A. Weinstock, *Methods of Melanoma Detection*, 2016, vol. 167.

[10] E. A. L. Enninga, J. C. Moser, A. L. Weaver, S. N. Markovic, J. D. Brewer, A. A. Leontovich, T. J. Hieken, L. Shuster, L. A. Kottschade, A. Olariu, A. S. Mansfield, and R. S. Dronca, "Survival of cutaneous melanoma based on sex, age, and stage in the United States, 1992–2011," *Cancer Medicine*, vol. 6, no. 10, pp. 2203–2212, 2017.

[11] P. Tschandl, C. Rosendahl, and H. Kittler, "Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, pp. 1–9, 2018. [Online]. Available: http://dx.doi.org/10.1038/sdata.2018.161

[12] T. Mendonca, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira, "PH2- A dermoscopic image database for research and benchmarking," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2013, pp. 5437–5440.

[13] K. Korotkov, "Automatic Change Detection in Multiple Pigmented Skin Lesions," PhD, Universitat de Girona, 2014.

[14] A. Bono, S. Tomatis, C. Bartoli, G. Tragni, G. Radaelli, A. Maurichi, and R. Marchesini, "The ABCD system of melanoma detection: A spectrophotometric analysis of the asymmetry, border, color, and dimension," *Cancer*, vol. 85, no. 1, pp. 72–77, 1999.

[15] L. F. Montenegro Morillo, "Estudio de los hallazgos en la dermatoscopía para carcinoma basocelular en personas de fototipos III y IV de una población de consultorio privado y hospital de II nivel en Bogotá, Colombia," Ph.D. dissertation, Universidad Nacional de Colombia, 2018.

[16] M. P. Guevara Novoa, "Utilidad de la dermatoscopía para el diagnóstico de carcinoma basocelular en pacientes con diagnóstico clínico que acuden a la consulta externa de dermatología de los hospitales: Carlos Andrade Marín, dermatológico Gonzalo González y de especialidades de las fuerzas armadas nº1, 2014." Specialist thesis, Universidad Central del Ecuador, 2015.

[17] B. Barroso Fleitas, T. Díaz, Lorenzo, R. García Gómez, and S. Castañeda Lezcano, "La dermatoscopia en el diagnóstico de cáncer de piel no melanoma," *Folia Dermatológica Cubana*, vol. 8, no. 1, 2014.

[18] D. Patiño, J. Avendaño, and J. W. Branch, "Automatic Skin Lesion Segmentation on Dermoscopic Images by the Means of Superpixel Merging," in *International Conference On Medical Image Computing & Computer Assisted Intervention*, 2018, pp. 728–736.

[19] H. Wang, R. H. Moss, X. Chen, R. J. Stanley, W. V. Stoecker, M. E. Celebi, J. M. Malters, J. M. Grichnik, A. A. Marghoob, H. S. Rabinovitz, S. W. Menzies, and T. M. Szalapski, "Modified watershed technique and post-processing for segmentation of skin lesions in dermoscopy images," *Computerized Medical Imaging and Graphics*, vol. 35, no. 2, pp. 116–120, 2010.

[20] Y. Li and L. Shen, "Skin lesion analysis towards melanoma detection using deep learning network," *Sensors (Switzerland)*, vol. 18, no. 2, pp. 1–16, 2018.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[22] V. Iglovikov and A. Shvets, "Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation," *CoRR*, vol. abs/1801.05746, 2018. [Online]. Available: http://arxiv.org/abs/1801.05746

[23] A. W. Rosyadi and N. Suciati, "Image segmentation using transition region and k-means clustering," *IAENG International Journal of Computer Science*, vol. 47, no. 1, pp. 47–55, 2020.

[24] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. 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)," in *Proceedings - International Symposium on Biomedical Imaging*, vol. 2018-April, 2018, pp. 168–172.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[26] M. Attia, M. Hossny, S. Nahavandi, and A. Yazdabadi, "Skin melanoma segmentation using recurrent and convolutional neural networks," in *Proceedings - 2017 IEEE 14th International Symposium on Biomedical Imaging*, 2017, pp. 292–296.

[27] L. Bi, J. Kim, E. Ahn, D. Feng, M. Fulham, N. Medicine, R. Prince, and A. Hospital, "Automatic Melanoma Detection via Multi-scale Lesion-biased Representation and Joint Reverse Classification," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016, pp. 1055–1058.

[28] O. Lezoray, M. Revenu, and M. Desvignes, "Graph-based skin lesion segmentation of multispectral dermoscopic images," in *2014 IEEE International Conference on Image Processing, ICIP 2014*, 2014, pp. 897–901.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[30] Q. Zheng, M. Yang, X. Tian, X. Wang, and D. Wang, "Rethinking the role of activation functions in deep convolutional neural networks for image classification," *Engineering Letters*, vol. 28, no. 1, pp. 80–92, 2020.

[31] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, no. October 2017, pp. 146–157, 2018. [Online]. Available: https://doi.org/10.1016/j.inffus.2017.10.006

[32] X. Zhang, "Melanoma segmentation based on deep learning," *Computer Assisted Surgery*, vol. 22, no. 1, pp. 267–277, 2017. [Online]. Available: https://doi.org/10.1080/24699322.2017.1389405

[33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 88, no. 11, 1998. [Online]. Available: http://ieeexplore.ieee.org/document/726791/{\#}full-text-section

[34] D. Gutman, N. C. F. Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, and A. Halpern, "Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)," in *Proceedings - International Symposium on Biomedical Imaging*, 2016, pp. 3–7. [Online]. Available: http://arxiv.org/abs/1605.01397

[35] J. P. Ebenezer and J. C. Rajapakse, "Automatic segmentation of skin lesions using deep learning," *CoRR*, vol. abs/1807.0, 2018. [Online]. Available: http://arxiv.org/abs/1807.04893

[36] M. H. Jafari, N. Karimi, E. Nasr-Esfahani, S. Samavi, S. M. Soroushmehr, K. Ward, and K. Najarian, "Skin lesion segmentation in clinical images using deep learning," in *Proceedings - International Conference on Pattern Recognition*, 2016, pp. 332–337.

[37] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 241–246.

[38] 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*, 2015, pp. 234–241.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, vol. 9, pp. 249–256, 2010.

[40] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.0, 2016. [Online]. Available: http://arxiv.org/abs/1609.04747

[41] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo, "Why Random Reshuffling Beats Stochastic Gradient Descent," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1510.08560

[42] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[43] R. Mondal, P. Purkait, S. Santra, and B. Chanda, "Morphological Networks for Image De-raining," in *International Conference on Discrete Geometry for Computer Imagery*, 2019, pp. 262–275.

[44] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, 2015, pp. 448–456. [Online]. Available: http://arxiv.org/abs/1502.03167

[45] N. C. F. Codella, Q. Nguyen, S. Pankanti, D. Gutman, B. Helba, A. Halpern, and J. R. Smith, "Deep learning ensembles for melanoma recognition in dermoscopy images," *CoRR*, vol. abs/1610.04662, 2016. [Online]. Available: http://arxiv.org/abs/1610.04662

[46] X. Li, L. Yu, C. W. Fu, and P. A. Heng, "Deeply supervised rotation equivariant network for lesion segmentation in dermoscopy images," in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, 2018, pp. 235–243.

[47] M. Goyal and M. H. Yap, "Automatic Deep Learning Ensemble Methods for Skin Lesion Boundary Segmentation," *CoRR*, vol. abs/1902.0, 2019.

[48] X. Li, L. Yu, H. Chen, C.-W. Fu, and P.-A. Heng, "Transformation Consistent Self-ensembling Model for Semi-supervised Medical Image Segmentation," *CoRR*, vol. abs/1903.0, 2019. [Online]. Available: http://arxiv.org/abs/1903.00348

[49] N. Codella, Q.-B. Nguyen, S. Pankanti, D. Gutman, B. Helba, A. Halpern, and J. R. Smith, "Deep Learning Ensembles for Melanoma Recognition in Dermoscopy Images," *IBM Journal of Research and Development*, vol. 61, no. 4-5, pp. 5:1—-5:15, 2017. [Online]. Available: http://arxiv.org/abs/1610.04662