Abstract—Early detection of malignant skin lesions is crucial for survival. In particular, the most aggressive form of skin cancer (melanoma) has a survival rate of 99% if detected in an early stage. Dermoscopy is a technique that allows a more accurate diagnosis of skin lesions by magnifying and clearing surface reflection of the skin. Automatic detection and segmentation of skin lesions can help the medical community perform a diagnosis. This thesis develops a deep learning multi-task network for automatic segmentation and classification of skin lesions. This multi-task network is based on a well-known segmentation architecture called U-Net, designed for biomedical image segmentation tasks. Independent models were created for both tasks and later joined, to study interactions between the models and joined benefits. The independent model for segmentation achieved a JAC of 70.6%, and the independent model for classification achieved a Balanced Accuracy of 64.1%. Both techniques reached good results. With the joined model, the system achieved a JAC of 66.3%, and a Balanced Accuracy of 59.6%.

Index Terms—Skin Lesion Classification, Skin Lesion Segmentation, Dermoscopy, Deep Learning

I. INTRODUCTION

Cancer can be described as an uncontrollable growth of abnormal cells, prone to metastasize and ultimately leading to death. According to the World Health Organization (WHO) [1], one in six deaths in the world is attributed to cancer, and in every three cancers diagnosed, one is skin cancer. Within skin cancer, melanoma is the deadliest type due to its high likelihood to metastasize. However, if detected early, the 5-year survival rate for melanoma is 99% [2]. Skin cancer can be identified by the presence of skin lesions, being susceptible to early diagnosis by experienced eyes. If detected early, a melanoma can be excised, eliminating the threat of new metastases. Nonetheless, the diagnosis of melanoma is challenging, and many times the diagnosis is only final after being excised and sent to a pathologist for definitive identification [3]. The diagnosis and respective excision of the skin lesion require the correct delineation of the lesion limits, known as segmentation [4].

Consequently, it is important to develop techniques and methods which allow the early detection of melanomas. Dermoscopy is a technique that allows a more accurate diagnosis of skin lesions by using a dermatoscope. Dermoscopy magnifies the lesion up to 100x, allowing visualization of deeper layers of the skin while eliminating surface reflection. With current advances in technology, digital dermatoscopes, and even dermatoscope attachments for smartphones, have become standard. This analysis of digital dermoscopy images renders a great improvement compared to standard photography.

Using these digital dermoscopy images, automatic systems for skin diagnosis and segmentation can be developed. Since segmentation and classification of skin lesions are crucial, this thesis focuses on creating a system that can automatically diagnose and segment skin lesions through dermoscopy images, in order to auxiliate doctors.

This work aims to develop an automatic tool based on deep learning for simultaneous classification and segmentation of different skin lesions in dermoscopy images. In the first place, a segmentation model is developed, using as a starting point the U-Net that was proposed by Ronneberger et al. [5] for biomedical image segmentation. Second, a classification model is developed, taking advantage of the encoder used for the segmentation model.

Finally, these two models are fused into a single model that simultaneously classifies and segments the lesions, while sharing weights in several layers. The main goal is to study the influence and benefits of a multi-task network.

There are two main contributions:

• Development of a dual-purpose model for segmentation and classification of skin lesions based on deep learning.

• Performance evaluation of the developed model on a state-of-the-art dataset.

In this paper, a brief analysis of state of the art architectures is done, followed by a detailed explanation of each task architecture. In the subsequent sections, the implementation and optimization of each task is described. Afterwards, the obtained results are evaluated and discussed, closing with the conclusions and future work.

II. STATE OF THE ART IN SEGMENTATION AND CLASSIFICATION

With the growing awareness of skin-related cancers, the research community has concentrated many efforts in developing diagnostic tools for segmentation and classification of skin lesions. Challenges created by the International Skin Imaging Collaboration (ISIC) have attracted global participation and yielded novel findings on these subjects. In recent years, convolution networks led to the best performance values. In light of this, ISIC 2017 challenge had most submissions using Convolutional Neural Networks (CNN) techniques. Table I shows the results of the best classified models and their performance values on the test set.

Joint Segmentation and Diagnosis of Skin Lesions

Francisco CastelBranco da Silveira de Sena Esteves
Instituto Superior Técnico
Universidade de Lisboa
Lisboa, Portugal
franciscosenae@tecnico.ulisboa.pt
TABLE I
CNN ARCHITECTURES USED FOR IMAGE SEGMENTATION

<table>
<thead>
<tr>
<th>Dataset Year</th>
<th>Ref</th>
<th>CNN base Architecture</th>
<th>JAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISIC 2017</td>
<td>[6]</td>
<td>FCN [8]</td>
<td>76.5%</td>
</tr>
</tbody>
</table>

Similarly to image segmentation, many efforts have been put on the development of automatic diagnostic tools. Table II resumes relevant works done in public databases and the corresponding performance evaluation.

TABLE II
CNN ARCHITECTURES USED FOR IMAGE CLASSIFICATION. SENSITIVITY AND SPECIFICITY GIVEN FOR MELANOMA CLASS. WHEN VARIOUS WORKS ARE SHOWN, PERFORMANCE VALUES ARE GIVEN FOR THE BEST CLASSIFIED. BACC: BALANCED ACCURACY OF THE THREE CLASSES, SE: SENSITIVITY, SP: SPECIFICITY

<table>
<thead>
<tr>
<th>Dataset Year</th>
<th>Ref</th>
<th>CNN Architecture</th>
<th>SE</th>
<th>SP</th>
<th>BACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISIC 2017</td>
<td>[12]</td>
<td>ResNet</td>
<td>73.5%</td>
<td>85.1%</td>
<td>-</td>
</tr>
<tr>
<td>ISIC 2017</td>
<td>[10]</td>
<td>Inception-v4</td>
<td>45.1%</td>
<td>97.0%</td>
<td>-</td>
</tr>
<tr>
<td>ISIC 2018</td>
<td>[15]</td>
<td>DenseNet</td>
<td>-</td>
<td>-</td>
<td>84.5%</td>
</tr>
</tbody>
</table>

III. PROPOSED MODEL

The main objective of this thesis is to analyze the benefits of a multi-task model that combines the segmentation and classification of dermoscopy images. This chapter overviews the full model architecture, and describes each of the independent models. Finally, the image pre-processing and post-processing steps are presented.

IV. SYSTEM ARCHITECTURE

Figure 1 depicts the diagram of the proposed multi-task model. The network consists of two tasks that can be combined. The pre-processing block transforms the dermoscopy image in order to feed the segmentation and classification tasks. The outcome of the segmentation is passed onto a post-processing block to obtain a definite segmentation mask. Each task is tested and improved independently.

V. SEGMENTATION ARCHITECTURE

Figure 2 depicts the network in use. The first part of the network, the encoder, converts the Red, Green, Blue (RGB) image in a set of features, and the second part, the decoder, converts the features in a binary image.

For the segmentation task, minor changes were made to the original architecture. The input layer was designed to take advantage of the three color channels RGB, from the dermoscopic images. Apart from this change, the remaining layers were implemented as proposed in the original network. The input image resolution and the depth of the filter space in each block is the same as originally proposed for the U-Net. It is important to note that the output of the U-Net doesn’t have the same resolution of the input. This is due to 18 unpaded convolutions that crop the input. Consequently, the segmented output map represents the cropped output. Some image pre-processing is done to adapt the dataset to this network architecture.

VI. CLASSIFICATION ARCHITECTURE
For the classification task, we take the output of the U-Net encoder and join a fully connected Neural Network (NN). Figure 3 shows the encoder used from the segmentation task and the novel layers for classification. With an output of $28 \times 28$ pixels on the lowest resolution, we follow a basic Artificial Neural Network (ANN) and reduce the resolution with a max-pooling layer, followed by two fully connected layers. The first fully connected layer transforms the features, and the last fully connected layer drives the final classification decision. This last layer takes the inputs from the feature analysis and applies weights to predict the correct label corresponding to the three classes represented in our dataset. This three neuron output gives the final probabilities for each class. To achieve the final probabilities, we use the Softmax activation function, which is given by:

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$$  \hspace{1cm} (1)

where $i = 1, \ldots, C$, with $C$ as the number of classes, and $\text{softmax}(x)_i$ represents each of the three outputs. This activation function simply turns the numeric output of the last fully connected layer into probabilities. This vector of probabilities adds up to one.

VII. Multi-task Architecture

![Model fusion for segmentation and classification](image)

The proposed model in this thesis consists of the adapted U-Net described above for segmentation and the novel layers used for classification. Figure 4 shows the fused model architecture, that corresponds to the assembly of the two models.

VIII. Pre-processing

For training and testing, the images need to be pre-processed to meet the model requirements. The first layer in the network is a three dimensional array of size: $height \times width \times depth$, where $height$ and $width$ are the resolution of the input image and depth is the channel dimension. Using the proposed size, the input image has a resolution of $572 \times 572$. Since most of the images in the dataset have a larger resolution and are not square, the images need to be resized to the above resolution. The pre-processing of the images has three steps.

- Scaling after equaling height and length
- Extra padding for lesion border
- Image normalization
- Data Augmentation

A fourth step is needed for the training process: data augmentation.

IX. Post-processing

The output of the network returns a rough segmentation map of the lesion that can be improved by an automatic process. The most useful operations in our model involve removing foreign objects, and clearing noise. The post processing method is composed of 5 steps:

- Rounding the values of the segmentation map to 1 (lesion label) or 0 (skin), to obtain a binary mask.
- Morphological close, to remove noise and to join nearby lesion areas.
- Choose the Largest Connected Component (LCC), since there is only one lesion per input image, there can be only one output lesion. Some lesions have a distinctive border but a difficult interior.
- Flood filling, as some lesions have a distinctive border but a difficult interior to detect.
- Original resolution restore. The predicted segmentation map is brought back to the original shape, from $388 \times 388$ to each image respective resolution.

X. Methodology

The dataset used in this thesis was released for a public challenge and is already subdivided in three sets: train, validation, and test. Since we want to analyze our model and compare with the challenge leaderboards, we use the three sets in the same way. The training set is used to train the network. The validation set is used to create benchmarks between different hyper-parameters choices, to stop the learning process, and to compare as well with the leaderboard. Finally, we use the test set to avail the model, compare with the leaderboard, and to confirm that it isn't biased toward the train and validation dataset.

The model is trained in mini-batch mode, with the number of examples in each mini-batch as hyper-parameter. The training data is shuffled at each epoch. This shuffling decreases the possibilities of having mini-batches that are not representative of the dataset. For example, a mini-batch with all images belonging to one class. The use of shuffled mini-batches has been proved to obtain faster convergence [16].
For this reason, the conducted experiments followed the best-first search algorithm, which works by repeatedly expanding the most promising node according to a specific rule. In this implementation, each level of the tree corresponds to a different hyper-parameter, where the nodes correspond to the discrete values for the parameters (for interval data, sensible and equally distant values are chosen). For this reason the conducted tests followed a basic tree search, where each improvement in the hyper-parameters is implemented to the baseline and therefore used in the next hyper-parameter test.

The combined model is generated from the optimization of each separate task model. Final hyper-parameters are tested, and the interactions between the models are studied. Performance values are obtained and compared with the separate models.

XI. Dataset

The proposed model was trained and tested on a public dataset of dermoscopic images ISIC 2017. This dataset is from the challenge “ISIC 2017: Skin Lesion Analysis Towards Melanoma Detection”, consisting of 2000 images provided as training data, 150 as validation and 600 for testing [19]. These images have dimensions between 576x768 and 4499x6748. Table III shows the lesion distribution for each set (training, validation, test) in the dataset.

<table>
<thead>
<tr>
<th>TABLE III</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATASET ORGANIZATION (NUMBER OF IMAGES AND PERCENTAGES)</strong></td>
</tr>
<tr>
<td>Melanoma</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td><strong>Train</strong></td>
</tr>
<tr>
<td><strong>Validation</strong></td>
</tr>
<tr>
<td><strong>Testing</strong></td>
</tr>
</tbody>
</table>

The proposed model was trained and tested on a public dataset of dermoscopic images ISIC 2017. This dataset is from the challenge “ISIC 2017: Skin Lesion Analysis Towards Melanoma Detection”, consisting of 2000 images provided as training data, 150 as validation and 600 for testing [19]. These images have dimensions between 576x768 and 4499x6748. Table III shows the lesion distribution for each set (training, validation, test) in the dataset.

XII. Evaluation Metrics

The proposed system aims to perform two tasks: lesion segmentation and lesion classification. We will briefly review the metrics used to assess the model performance in each of the tasks.

A. Segmentation Metrics

In order to evaluate the proposed network segmentation quantitatively, the metrics were required to focus on the lesion being well segmented.

The pixel accuracy is an easy concept to understand and to implement. However, pixel accuracy returns very optimistic values and can provide misleading results. In images where the skin lesion is small, the measure will be biased in reporting how well the larger class (skin) is represented (i.e., a 95% accuracy can show a segmentation without the lesion being discovered).

We admit the lesion in each example is a single region, connected. From the predicted segmentation map, the region A is obtained (classified as lesion) and compared with the area B, obtained from the ground truth mask.

With this in mind, the following metric was used:
• Jaccard Index (JAC), also known as IoU, measures the intersection over union between the predicted lesion with the ground truth mask [20]. This metric ranges from 0 to 1 with 0 standing for the model not finding the lesion whereas 1 representing exact overlap between lesion prediction and ground truth.

\[ JAC(A, B) = \frac{|A \cap B|}{|A \cup B|}, \]  
(5)

where A is the predicted probability map and B is the ground truth binary mask for each image. For the model performance, the mean JAC of all examples is taken.

B. Classification Metrics

There are standard performance measures for binary classification problems, e.g., sensitivity, specificity, accuracy.

However, the dataset considered in this thesis has three classes. We consider three binary problems (one-against-all) and compute the statistics for each binary problem. These statistics are combined to obtain the overall metrics. The performance of the model was obtained by comparing the output of the classifier with the ground truth label.

By comparing the two, and having x as one of the three classes, we can obtain:

- \( TP_c \): Number of skin lesions correctly classified as c
- \( TN_c \): Number of skin lesions correctly classified as not c
- \( FP_c \): Number of skin lesions wrongly classified as c
- \( FN_c \): Number of skin lesions wrongly classified as not c

Where c is the type of skin lesion: Melanoma, Seborrheic Keratosis or Nevus.

Accuracy is one of the most commonly used measures for the classification performance, and it is defined as a ratio between the correctly classified samples to the total number of samples as follows:

\[ Acc_c = \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c}, \]  
(6)

Sensitivity (SE), also known as recall or TPR, represents the proportion of correctly classified positive samples to the total number of positive samples, as follows:

\[ SE_c = \frac{TP_c}{TP_c + FN_c}, \]  
(7)

Specificity (SP), also known as TNR, is expressed as the ratio of the correctly classified negative samples, as follows:

\[ SP_c = \frac{TN_c}{TN_c + FP_c}. \]  
(8)

In summary, specificity represents the proportion of negative samples that were correctly classified and the sensitivity is the proportion of positive samples that were correctly classified.

Furthermore, in order to evaluate the model in respect to the three classes, the Balanced Accuracy was taken into account. The Balanced Accuracy is the average sensitivity obtained on each class, as follows:

\[ BACC = \frac{SE_M + SE_N + SE_{SK}}{3}, \]  
(9)

With M as Melanoma, N as Nevus and SK as Seborrheic Keratosis. Considering the dataset as a whole, the Categorical Accuracy can also be considered, but gives results that are far more optimistic. The Categorical Accuracy is given as follows:

\[ Acc = \frac{TP_M + TP_N + TP_{SK}}{TP_M + TP_N + TP_{SK} + \sum_c FN_c}, \]  
(10)

where \( \sum_c FN_c \) are the False Negatives for each one of the classes. This metric doesn’t take into account the imbalance between the classes, and so, the performance metric that better suits our dataset and challenge, is the Balanced Accuracy.

By computing the Balanced Accuracy instead of the Categorical Accuracy, each error in a poorly represented class has the same weight in the model performance as the remaining classes. For example, Seborrheic Keratosis lesions have a share as low as 12.7% (Table III). Using this metric, each error can have the same weight in the model performance as the most represented class. This way, we can have a better insight of the model.

C. Combined Model Metrics

Since the combined model performs simultaneously classification and segmentation, both tasks must be evaluated together. The chosen metric for model evaluation is given by:

\[ Mean \text{ Performance} = \frac{BACC + JAC}{2} \]  
(11)

which corresponds to the mean between the segmentation Jaccard Index for the dataset and the classification Balanced Accuracy. Although the metrics are not comparable, this metrics serves as a base point of comparison between hyper-parameter tests, allowing the a perception of how the model is behaving in both tasks.

XIII. COMPUTATIONAL EXPERIMENTAL CONFIGURATION

This thesis implementation was mostly developed and evaluated on a computer provided by IST. The computer has the following specifications:

- Processor: Intel Core i7-7700 CPU @ 3.6GHz
- Memory (RAM): 16GB
- GPU: NVIDIA GeForce GTX 1060 6GB

Additionally, Google Colaboratory was used as well, which is a cloud based environment with GPU included. The software used is available online and open-source. For training and evaluation the following libraries were used in python 3.7.6:

- Tensorflow
- Keras
- Scikit: Learn, Image. For image pre-processing and evaluation metrics.
XIV. Segmentation Results

A. Baseline

The baseline model for segmentation is a reproduction of the U-Net network, adapted to take advantage of the three color channels RGB. Higher image resolutions for the network input were tested since more information could be retained. However, as the GPU memory is limited, there is a trade-off between batch size, resolution, and size of the network. Higher image resolution would force the network to be simpler and smaller. This proved to be critical in the segmentation task. Table IV resumes the configuration adopted for the baseline with the corresponding JAC score obtained.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Depth</th>
<th>Parameters</th>
<th>Activation</th>
<th>Batch</th>
<th>Dropout</th>
<th>Loss</th>
<th>JAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>64</td>
<td>31M</td>
<td>ReLU</td>
<td>4</td>
<td>False</td>
<td>BCE</td>
<td>54%</td>
</tr>
</tbody>
</table>

B. Network Optimization

For the segmentation task, the hyper-parameter tuning yield the best results with a simpler and less dense network. This was done mostly by lowering the depth of the filters in each convolutional block. Additionally, data augmentation and the inclusion of dropout improved the network.

Our optimized network obtained an improvement of 35% over the baseline one, given the time constraint and the physical limitations of the GPU. Figures 5 and 6 shows the model performance. In figure 6, even though the JAC value is 85%, which is above average, we can note that the model is only able to segment part of the lesion. This is due to the cropping that is done by the convolutional layers, despite the pre-processing part addressing this issue. For very large lesions, this effect occurs. Table V summarizes the configuration after optimization with the statistical analysis of validation set. This JAC score presented is already with the image post processing step, which renders an increase in 2.5%.

C. Segmentation Test Set Evaluation

The test set performance results are very close to the validation set. Table VI shows the performance values and the improvement obtained with the post-processing step. This Jaccard Index of 70.6% on the test set is a good score. Comparing directly with the state of the art presented in chapter 2, this score is 5.9% short of the top participant.

XV. Classification Results

A. Baseline

Our classification network is based on the best U-Net encoder found for the segmentation task. We maintained all of the encoding path and added the following sequential layers: max pooling and two fully connected. The last fully connected layer has 3 neurons, corresponding to the three types of skin lesion.

Since the three lesion classes are not equally represented, the model is biased towards the most represented one. To avoid this, the loss function becomes a weighted average, where the weight of each sample is specified according to the belonging class, by:

$$ W_c = \frac{N}{n_{\text{classes}} \times n_c}, \quad (12) $$

where $c$ is the class (melanoma, seborrheic keratoses, or nevus), $N$ is count of images in the set, $n_c$ is the count
of images belonging to class \( c \) in the set, and \( n_{\text{classes}} \) is the number of classes. Therefore, Nevus has a weight of 0.486, Melanoma has a weight of 1.783, and Seborrheic Keratosis of 2.625. These weights mean that each melanoma example will be as important as almost 4 nevi examples. This solution deeply increased the sensitivity of the less represented classes. Additionally, the same data augmentation from the segmentation task was tested.

For the classification task, the cost function assigned was Categorical Cross Entropy, using the same ADAM optimizer with a decreasing step, starting at 1e-3 and lowering to 80% of its value every time the loss value doesn’t decrease for 5 epochs. In addition to these changes, the training images are fed to the network in a random order, with the goal of not learning one class at the time, since the dataset could be distributed this way. Our baseline used a batch size of 8, higher than the value used for the segmentation task, on the grounds that a bigger batch size can contain examples of the three classes, and therefore removing the bias introduced by a batch with only one class. Additionally, for the max pooling layer the baseline has pool size of \( 2 \times 2 \) with stride 2, and the fully connected layer has a size of 1024. Table VIII sums up the baseline configuration.

**TABLE VIII**

<table>
<thead>
<tr>
<th>Pool</th>
<th>Fully Connected</th>
<th>Dropout</th>
<th>Batch</th>
<th>DA</th>
<th>BACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2 \times 2 )</td>
<td>1024</td>
<td>False</td>
<td>8</td>
<td>True</td>
<td>71.8%</td>
</tr>
</tbody>
</table>

**B. Network Optimization**

The best classification model was achieved by tuning the size of the fully connected layer, and the pool and stride size of the max pooling layer.

Figure 7 to figure 9 shows the model performance, with the input image, the model classification output and the true class. There are two lesion input which obtain a correct classification. Figure 8 shows a common mistake for this network, to consider seborrheic keratosis lesions as melanomas. As can be seen with the following lesions, both can be very similar.

**TABLE IX**

<table>
<thead>
<tr>
<th>Set</th>
<th>BACC</th>
<th>Sensitivity (SE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>75.4</td>
<td>70.0 70.5 85.7</td>
</tr>
<tr>
<td>Test</td>
<td>64.1</td>
<td>58.1 55.2 78.9</td>
</tr>
</tbody>
</table>

**C. Classification Test Set Evaluation**

The test set performance results differ from the validation set. Table IX shows the performance values on both sets. Both sets differ on more than 10%, and can mean that the model did not learn to generalize enough. Comparing with the state of the art, our model achieved a lower melanoma sensitivity.

**TABLE X**

<table>
<thead>
<tr>
<th>Depth</th>
<th>Parameters</th>
<th>Activation</th>
<th>Batch</th>
<th>Dropout</th>
<th>Loss weight (( \alpha ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>50M</td>
<td>ReLU</td>
<td>8</td>
<td>True</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**B. Network Optimization**

The multi-task model was optimized using the \( \alpha \) value. This value assigns the importance between the two tasks. A value closer to 1 assigns more importance to the segmentation task, and a value closer to 0 prefers the classification task. The target was to find a value where both tasks reach admissible performance values that match or improve the separate networks.

With the multi-task and the independent models, two hypothesis were tested:

- Transfer Learning from segmentation model to multi-task model:
  - Use the segmentation weights and train only the layers exclusive to classification, with \( \alpha \) value set to 0. This
test can show if the weights used for segmentation can be used as well for classification.

- Multi-task model trained from scratch:
  Train the whole network from scratch for simultaneous segmentation and classification. Weights are trained and features are extracted to favor both outputs of the network. Here, the $\alpha$ value is optimized to favor both tasks equally.

Table XI shows the tests performed over the $\alpha$ value. The value of 0.8, which prioritizes the segmentation task, obtained the best value of mean performance. However, this yield performance values lower than the independent models (classification task worse by 1.4% and segmentation task worse by more than 5%).

TABLE XI
VALUES TESTED FOR $\alpha$ HYPER-PARAMETER

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Epoch</th>
<th>Mean Performance</th>
<th>JAC</th>
<th>BACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>73</td>
<td>58.5</td>
<td>58.0</td>
<td>59.0</td>
</tr>
<tr>
<td>0.3</td>
<td>90</td>
<td>67.7</td>
<td>65.7</td>
<td>69.6</td>
</tr>
<tr>
<td>0.5</td>
<td>55</td>
<td>67.8</td>
<td>64.4</td>
<td>71.1</td>
</tr>
<tr>
<td>0.7</td>
<td>32</td>
<td>69.0</td>
<td>65.3</td>
<td>72.2</td>
</tr>
<tr>
<td>0.8</td>
<td>22</td>
<td>70.8</td>
<td>67.3</td>
<td>74.0</td>
</tr>
<tr>
<td>0.9</td>
<td>40</td>
<td>69.1</td>
<td>67.4</td>
<td>70.9</td>
</tr>
</tbody>
</table>

C. Overview of the Results for the Multi-task Network

The best model configuration led to performance below the expected (seen in table XII).

TABLE XII
BEST MULTI-TASK MODEL

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Balanced Accuracy</th>
<th>JAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>74.0%</td>
<td>67.5%</td>
</tr>
</tbody>
</table>

Figure 10 to figure 12 show three examples of the model output. Example 3 (seen in figure 12) shows a segmentation error where almost all of the ground truth image is considered a lesion. The proposed model fails to realize this and segments only a small part inside the lesion.

However, despite the drop in balanced accuracy, this model has performed better in melanoma lesions sensitivity (seen in table XII). The value increased in 6.7%, meaning that more malignant lesions (melanoma) will be detected. However, this was at the cost of a lower specificity, meaning more false positives will be generated for melanoma lesions. Further statistical analysis can be seen in table XIII.

TABLE XIII
EVALUATION OF SENSITIVITY AND SPECIFICITY FOR THE BEST CONFIGURATION

<table>
<thead>
<tr>
<th></th>
<th>Melanoma</th>
<th>Seborrheic Keratosis</th>
<th>Nevus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>76.7%</td>
<td>66.7%</td>
<td>78.6%</td>
</tr>
<tr>
<td>Specificity</td>
<td>80.0%</td>
<td>91.7%</td>
<td>88.9%</td>
</tr>
</tbody>
</table>
The two proposed hypothesis rendered:
Transfer Learning from segmentation model to multi-task model: This test achieved a balanced accuracy of 70.2% for the classification model, while maintaining the segmentation performance. This result proves that segmentation learned features on the encoder are successful for image classification.
Multi-task model trained from scratch: As seen in the optimized model, the performance values were close to the independent models, but didn’t attain any performance improvement.

D. Multi-task Network Test Set Evaluation

The multi-task model obtained equivalent performance values for the segmentation task. However, for the classification task the model was unable to maintain the performance values. Table XIV shows the performance values for the validation and test sets. The mean performance is worse by 7% and the balanced accuracy falls to 59.6%. For classification, the melanoma sensitivity is 10% below the state of the art.

<table>
<thead>
<tr>
<th>Set</th>
<th>Post-p. JAC</th>
<th>BACC</th>
<th>Mean Performance</th>
<th>Sensitivity M</th>
<th>Sensitivity SK</th>
<th>Sensitivity N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>67.5</td>
<td>74.0</td>
<td>70.8</td>
<td>76.7</td>
<td>66.1</td>
<td>78.6</td>
</tr>
<tr>
<td>Test</td>
<td>66.3</td>
<td>59.6</td>
<td>63.0</td>
<td>63.2</td>
<td>44.5</td>
<td>71.1</td>
</tr>
</tbody>
</table>

XVII. Conclusions and Future Work

A. Conclusions

The main goal of this thesis was to create an automatic tool for simultaneous recognition and segmentation of different skin lesions in dermoscopy images. The algorithm for this tool comprises a pre-processing step, the deep learning architecture which segments and classifies, and a post-processing step to enhance the segmentation output.

The pre-processing step adapts the images for the architecture in use by artificially enlarging and down scaling. This method doesn’t distort the original image. The post-processing step enhances the images by removing excessive noise and selecting the most promising lesion in each image. Important to note that the model Jaccard Index improved between 2% and 5% with this automatic post processing technique.

Each task was enhanced independently, using a wide known deep neural network for medical images (U-Net). Each task achieved good scores, with segmentation (JAC = 70.6%), and classification (Balanced Accuracy = 64.1%). The combination of these two tasks yielded results in the same range of the separate tasks, and for half the computational cost. Unfortunately, the 2017 ISIC challenge leaderboard is no longer available online, making it impossible to position this work among the rest of the participants. For this implementation, there is no performance gain in joining both tasks. However, the savings done in computational cost must be recognized. Moreover, the independent trained network for segmentation can be of use for the classification task with additional training. These results prove that features extracted from a CNN encoder can favor multiple tasks. The full implementation built for this thesis is available at https://github.com/franciscosenae/Tese.

B. Future Work

The joint method proposed in this thesis achieves performance values on par with the independent models for segmentation and classification, while using half the computational time. However, these conclusions are made for a specific CNN network with a extensive but basic parameter optimization.

This work could be improved in different ways:
- **Pre-processing:**
  - The idea of re-scaling and resizing an image to a size that can be 10 times smaller implies that a lot of information is lost. A sliding window (with the input size of the model) that could segment and classify the image, and posteriorly joined the information, would likely achieve better performance values.
- **Classification architecture:**
  - The classification architecture could benefit with an earlier branching from the segmentation architecture, or even with dedicated convolutional layers.
- **Hyper-parameter optimization:**
  - Methods such as Bayesian optimization can be successful in discovering better parameter configurations.
- **Post-processing:**
  - The post processing step was carefully chosen and optimized. However, the large amplitude in additional performance between different models hindered the model optimization.

**References**


