**Noname manuscript No.**
(will be inserted by the editor)

# Semantic Segmentation for Skin Melanoma Detection

## A convolutional neural network implementation.

**Mario Alberto Flores Hernández**

Received: date / Accepted: date

**Abstract** The approach described in this paper for the detection of the skin cancer melanoma is known as semantic segmentation. The semantic segmentation is a computer vision task where the different regions of an image are classified according to a specific category. The method used for the implementation of a semantic segmentation software was using the convolutional neural network, specifically a feature pyramid network architecture (`FPN`), to train a model which performs the task. With the convolutional neural networks is possible to train models that replicate the necessary transformations to obtain a targeted output, the only requirement is a rich enough database of samples of the input data and the target or the expected output in relation with that input, which in this case was a segmentation mask of the dermoscopic image. With the experiments described in this paper, the dice loss and Jaccard equations, we obtained a performance profile of the neural network in the training and validation processes.

**Keywords** Neural Network · Convolution · Semantic Segmentation · Classification

## 1 Introduction

Skin is considered one of the largest organs in the human body, its function is to protect the internal organs and structures from the harsh environmental conditions such as temperature, radiation and bacteria. The skin also functions as a large sensorial interface that let us feel the environment conditions such as the temperature in the ambience and the texture of the objects. The skin cancer known as melanoma is one of the deadliest types of cancer if not detected in its early stages, because of the quick spread to other organs caused by the metastasis effect. One way to reduce its mortality rate is by using automatic detection with computer vision technologies, such as the deep learning technology. With deep learning is possible to train models that recognize the presence of the melanoma tissue using a configuration of fine-tuned parameters during a process known as *training* which compares the input of the model and the known output to that input, and then update the parameters to get the computed output closer to the known output for a later use in unknown truth data.

The *general objective* is to implement a convolutional neural network technology with the purpose of creating a dermoscopic computer vision tool to help medical staff to detect skin melanoma tissue in a automated or assisted way, and to contribute the cause of reducing skin cancer mortality rates.

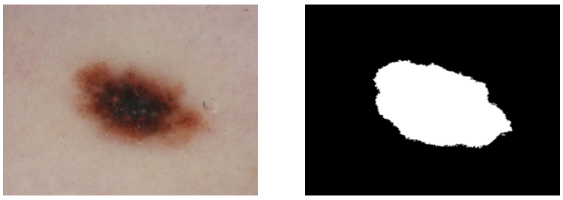
The *specific objective* is determine the process involved to implement said technology, the requirements and considerations implicit in the deployment of a convolutional neural network architecture and the required transformations of the data to make that possible.

In *section 2*, we introduce the basic concepts required to understand the following proposals. In *section 3*, we review the literature to determine the opportunity area in comparison with similar works. In *section 4*, we discuss the proposed solution design and implementation. In *section 5*, we report the results obtained from the evaluation metrics and in *section 6* we discuss the results obtained.

## 2 Background

In this section the basic concepts an theories employed in our proposed solution are discussed. Starting with

Address(es) of author(s) should be given



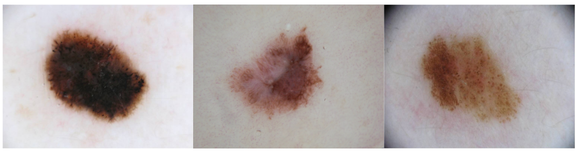
(a) Input (b) Ground Truth

Fig. 1: Example of data input/output.

some basic background aspects about the melanoma cancer, then continuing with the loss functions and the optimizer functions.

### 2.1 Melanoma

The skin cancer is a type of cell mutation that usually is related to the spontaneous appearance of asymmetric moles or patches of dark/red skin, bulges or scales on the skin surface. This mutation of the melanocytes (the cells responsible for the skin pigmentation) can affect any skin tone, mostly exposed parts of the body to ultra-violet light like the arms and the neck, but not exclusively [2].



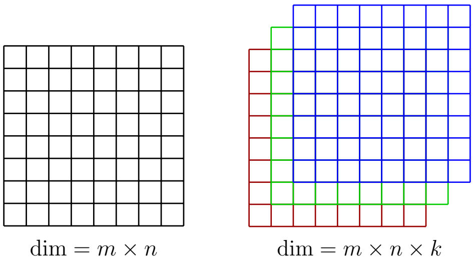
(a) (b) (c)

Fig. 2: Examples of skin melanoma obtained from the ISIC [4] database.

There are two main factors involved in the mutation of the melanocytes cells: intrinsic aging and extrinsic factors. The *intrinsic* aging occurs inevitably as a consequence of the time, while the *extrinsic* factors are rather external like the sunlight, nicotine abuse unhealthy lifestyles [5].

### 2.2 Image Dimensions

An important aspect to consider is the dimension of the images before introducing them to the neural network architecture. Images in a gray-scale configuration do not share the same dimensions of an image of the same resolution in a *rgb* configuration. A gray-scale or mono-channel image resolution can be described as $m \times n$, where $m$ and $n$ are the width and height respectively. This however is not the case for the rgb configuration which dimensions are described as $m \times n \times k$, where $k$ is the number of color channels. That means that in color images there are a specific number of layers where each has an independent set of pixel intensity values, as described in Figure 3.



(a) Grayscale. (b) Color.

Fig. 3: Dimensional comparison between images.

### 2.3 Convolutional Neural Networks

A convolutional neural network consists of two main phases: *coder* and *decoder*. In the *coder* phase the input image gets down-sampled to a smaller resolution object with more dimensions to perform the prediction. Then in the *decoder* phase, the image gets up-sampled or reconstructed from the prediction in the coding, creating a probabilistic map of the different regions in the image.

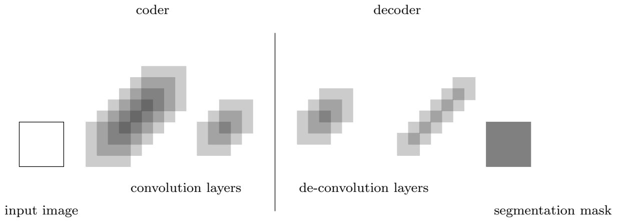


Fig. 4: Basic coder-decoder neural network architecture for segmentation tasks.

## 3 Related Work

In this section the reviewed literature is summarized to make a comparison between different methods to approach the melanoma detection problem and same approaches for the solution of different problems to get a scope of the opportunity areas, the advantages and disadvantages of the proposed solution.

### 3.1 Comparative Analysis

These are the characteristics used to compare the implementation in this paper with the work from the other authors.

**Classification.** Has the capacity of classify between two or more categories.

**Segmentation.** Can recognize the pixel-wise region of the different categories.

**Supervised.** Requires a curated database to perform the training process.

**Pre-trained.** Can initialize the model with pre-trained weights.

**Evaluation.** Can determine the precision of the computed output in comparison with the ground truth.

Table 1: Similitude and difference in the literature reviewed; the available features are shown as ✓, and the not available are represented with ✕.

| **Work** | **Model** | **Classification** | **Segmentation** | **Supervised** | **Pre-trained** | **Evaluation** | **Output** |
|---|---|---|---|---|---|---|---|
| Badrinarayanan *et al.* [1] | `SegNet` | ✓ | ✓ | ✓ | ✓ | ✓ | label |
| Ronneberger *et al.* [11] | `U-net` | ✓ | ✓ | ✓ | ✕ | ✓ | label map |
| Chen *et al.* [3] | `DeepLab` | ✓ | ✓ | ✓ | ✕ | ✓ | label map |
| Teichmann *et al.* [12] | `MultiNet` | ✓ | ✓ | ✓ | ✕ | ✓ | label map |
| Kroner *et al.* [8] | `VGG16` | ✓ | ✓ | ✓ | ✓ | ✓ | heat map |
| Kadampur y Al Riyaee [7] | `CNN` | ✓ | ✕ | ✓ | ✕ | ✓ | label map |
| Zhou *et al.* [14] | `ML / SVM` | ✓ | ✕ | ✓ | ✕ | ✓ | label |
| Luc *et al.* [10] | `CNN/GAN` | ✓ | ✓ | ✓ | ✓ | ✓ | label map |
| Jain *et al.* [6] | `A.B.C.D` | ✕ | ✓ | ✕ | ✕ | ✕ | label |
| Proposal | `FPN` | ✓ | ✓ | ✓ | ✓ | ✓ | label map |

### 3.2 Opportunity Area

For the semantic segmentation task is indispensable to count with a coder and decoder which dimensions and output coincide with the requirements of the input data and the desired output. The neural networks in contrast with other technologies offer that versatility, they can adjust the parameters during the transformation process to obtain the desired output in the dimensional aspect.

## 4 Proposed Solution

The approach proposed in this paper for the automated detection of the skin melanoma is the training of a model using a `FPN` convolutional neural network architecture [9], which output would be the probabilistic map of the regions inside of the dermoscopic input image. The language selected for the implementation of said neural network architecture was `Python`, because is a powerful, dynamic and quick to deploy language which also contains the `Torch` library which is the core tool of the implementation.

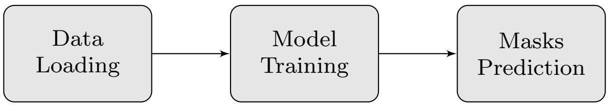


Fig. 5: Implementation process flowchart.

### 4.1 Implementation of solution.

The proposed implementation has four principal modules as shown in Figure 6. The `load_dataset.py` module has the function of loading, filtering and correcting the images.

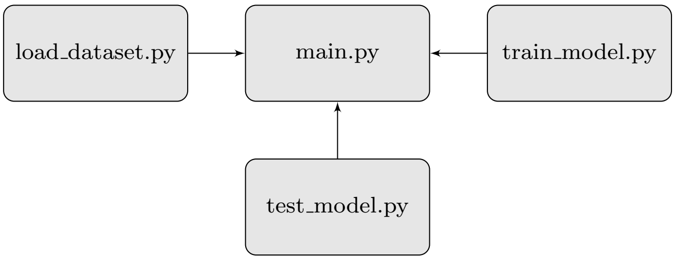


Fig. 6: Principal modules of the implementation.

The `train_model.py` module performs the model training using the `Torch` sampling functions. Then after obtaining a model from the training process, the `test_model.py` module reads a random image from the test images folder and computes a segmentation mask.

### 4.2 Pre-Processing

The *pre-processing* pipeline required to adapt the input image dimensions to the required dimensions by the architecture is described in the Figure 7. First a re-size is necessary to prevent `VRAM` memory saturation, the higher the resolution the higher the `VRAM` requirement. Another important aspect about the pre-processing is transforming the input resolution to a permitted resolution

$$\text{MP} = 2^N \quad (1)$$

where MP is the multiplier and $N$ is the number of layers of the coder phase. If the input resolution can be divided by the multiplier and the remainder is zero, then the resolution is compatible with the architecture. Then the data is transformed to a *tensor* which is an algebraic object that describes the relationship between objects.

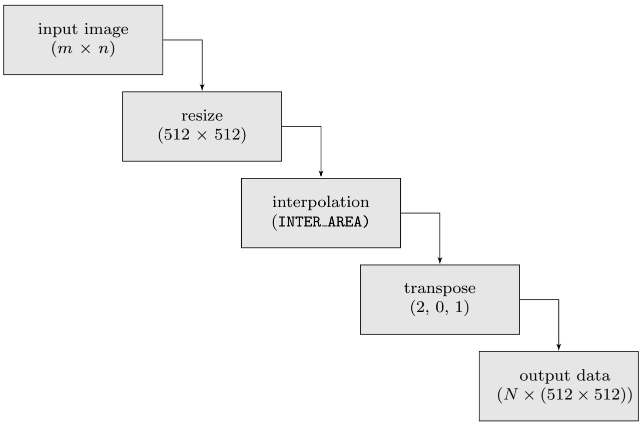


Fig. 7: Pre-processing pipeline.

### 4.3 Solution Design

In this section the code utilized for the implementation is discussed. The Listing 1 has the task of training the model with the data provided from the data loader module. The number of epochs can be specified and the model which gets the best score per epoch gets saved.

```python
# train_model.py
max_score = 0
for i in range(0,40):
    print('\n Epoch: {}'.format(i))
    train_logs = train_epoch.run(train_loader)
    valid_logs = valid_epoch.run(valid_loader)
    if max_score < valid_logs['iou_score']:
        max_score = valid_logs['iou_score']
        torch.save(model,('Model/'+encoder+'.pth'))
        print('Highest Score Model Saved: {}'.
  format(max_score))
    if i == 25:
        optimizer.param_groups[0]['lr'] = 1e-5
        print('decreased decoder learning rate to 1
  e-5')
 lr= 0.0001),])
```

Listing 1: Búcle de entrenamiento.

## 5 Experiments

The methods used to evaluate the performance of the training process and also gauge the probabilistic map predictions are described in this section. Also the analysis of the threshold is included here.

### 5.1 Threshold Analysis

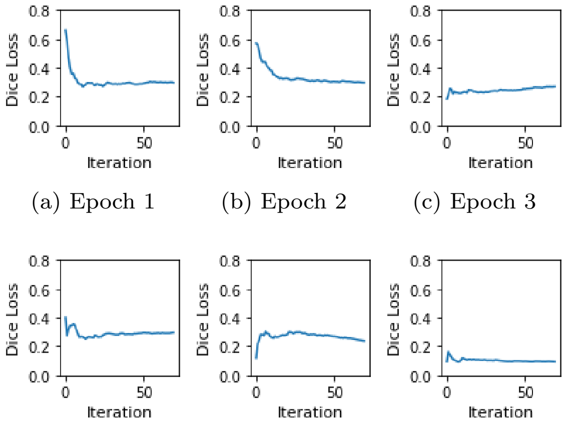
The threshold analysis consists of the visualization of the intensity bands where the pixels tend to cluster and that can affect the complexity level of the computation in the model.

### 5.2 Dice Loss

To determine the probability that two pixel intensity maps are similar is required to use an evaluation function which determines that similitude. The dice loss function compares the margin of equal values along two matrices and then divides it by the number of total pixels between both maps. The result obtained is a coefficient between zero and one, which represents the error.

$$\mathrm{DL} = \frac{2|A \cap B|}{|A| + |B|} \tag{2}$$

where $A$ and $B$ are the ground truth and the computed mask respectively.



(a) Epoch 1 (b) Epoch 2 (c) Epoch 3

(d) Epoch 4 (e) Epoch 5 (f) Epoch 6

Fig. 8: Error reduction in the training process by iteration.

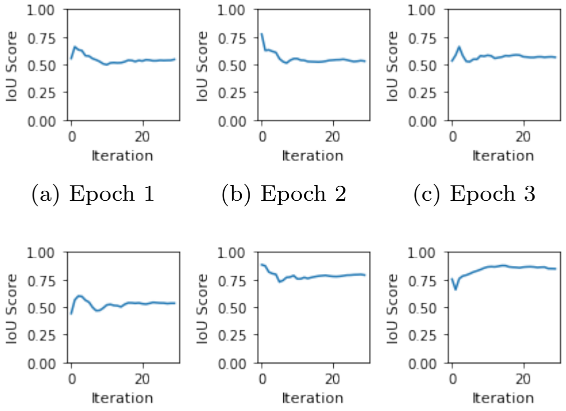
The *results* are visible in the Figure 8. The graphs show how the *error* fluctuates between iterations but decreases over the epochs. Meaning that there's improvement in the model parameters configuration.

### 5.3 Intersection Over Union

Similar to the *dice loss* function, the *intersection over union* or *Jaccard equation* compares the similar pixel intensities between two binary maps with the difference that the error has more penalization, which means that the result obtained is more robust. This is useful in the *validation* process where the error has more significance, the equation of Jaccard is

$$\mathrm{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \tag{3}$$

where $A$ and $B$ are the known mask and the computed mask respectively.



(a) Epoch 1 (b) Epoch 2 (c) Epoch 3

(d) Epoch 4 (e) Epoch 5 (f) Epoch 6

Fig. 9: Similitude between ground truth and computed output according to the IoU score by iteration.

The *results* for this experiment are the following. In the Figure 9 is visible that the similitude of the computed masks increases between iterations and epochs. In comparison with the *dice loss* equation this improvement has more weight since the *Jaccard* equation penalizes more the error.

### 5.4 Experiment Discussion

The process of error reduction is not linear, the Figure 8 and 9 show that exists some fluctuation between iterations, this however does not represent a problem because the code automatically saves the highest score model on each epoch. In the table 2, the highest score by epoch was registered.

Table 2: Summarized results by epoch.

| Epoch | DL | IoU |
|---|---|---|
| 1 | 0.2968 | 0.5616 |
| 2 | 0.2972 | 0.5546 |
| 3 | 0.2690 | 0.5879 |
| 4 | 0.2960 | 0.6389 |
| 5 | 0.2371 | 0.8415 |
| 6 | 0.0929 | 0.8626 |
| 7 | 0.0792 | 0.8626 |
| 8 | 0.0692 | 0.8798 |
| 9 | 0.0556 | 0.9017 |
| 10 | 0.0542 | 0.9039 |

## 6 Conclusion

In this document was described the process of the implementation of a convolutional neural network for the semantic segmentation task, by which is possible to classify between healthy tissue and melanoma tissue regions inside the dermoscopic images. A `feature pyramid network` was implemented and as a result of the convolutional layers and the associated weights(parameters) were adjust in relationship with the evaluation criteria from the dice loss and Jaccard functions. The result was a probabilistic map of the different categories inside the image.

### 6.1 Discussion

The main objective of this implementation was to obtain a tool able to perform perform the segmentation task in the dermoscopic images that hopefully contributes to the cause of the early detection of the melanoma disease. Thanks to the pre-build architectures of convolutional neural networks from the library provided by Yakubovskiy [13], it was possible to quickly deploy the FPN architecture and configure the network for the pre-processing and the training.

### 6.2 Future Work

This implementation of the semantic segmentation task has one major limitation, which is that it can distinguish between different categories inside the regions of the dermoscopic images, but it cannot distinguish between two objects of the same category which means that for example this same two objects are considered one single object inside the same category. This limitation in particular doesn't affect the purpose of this implementation because in this case the melanoma will always represent one single instance in the person. There are a lot of papers which talk about instanced segmentation where this is not the case, and the different instances of the different objects under the same category are recognized as different instances. I would like to experiment with instanciation and multi-class segmentation in the future.

## 7 References

1. Badrinarayanan, V., A. Kendall y R. Cipolla (2015), ≪SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation≫, *CoRR*, **abs/1511.00561**, `1511.00561[cs.CV]`.
2. Boni, R., C. Schuster, B. Nehrhoff y G. Burg (2002), ≪Epidemiology of skin cancer≫, *Neuroendocrinology Letters*, **23**, págs. 48–51. pmid: 12163848.
3. Chen, L., G. Papandreou, I. Kokkinos, K. Murphy y A. L. Yuille (2016), ≪DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs≫, *CoRR*, **abs/1606.00915**, `1606.00915[cs.CV]`.
4. Codella, N. C. F., V. Rotemberg, P. Tschandl, M. E. Celebi, S. W. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. A. Marchetti, H. Kittler y A. Halpern (2019), ≪Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)≫, *CoRR*, **abs/1902.03368**, `1902.03368[cs.CV]`.
5. Farage, M., K. Miller, P. Elsner y H. Maibach (2008), ≪Intrinsic and extrinsic factors in skin ageing: a review≫, *International Journal of Cosmetic Science*, **30**(2), págs. 87–95.
6. Jain, S., V. jagtap y N. Pise (2015), ≪Computer Aided Melanoma Skin Cancer Detection Using Image Processing≫, *Procedia Computer Science*, **48**, págs. 735–740. doi: 10.1016/j.procs.2015.04.209.
7. Kadampur, M. A. y S. Al Riyaee (2020), ≪Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images≫, *Informatics in Medicine Unlocked*, **18**, pág. 100 282. doi: 10.1016/j.imu.2019.100282.
8. Kroner, A., M. Senden, K. Driessens y R. Goebel (2020), ≪Contextual encoder–decoder network for visual saliency prediction≫, *Neural Networks*, **129**, págs. 261–270. doi: 10.1016/j.neunet.2020.05.004.
9. Lin, T., P. Dollár, R. B. Girshick, K. He, B. Hariharan y S. J. Belongie (2016), ≪Feature Pyramid Networks for Object Detection≫, *CoRR*, **abs/1612.03144**, `1612.03144`.
10. Luc, P., C. Couprie, S. Chintala y J. Verbeek (2016), ≪Semantic Segmentation using Adversarial Networks≫, *CoRR*, **abs/1611.08408**, `1611.08408`.
11. Ronneberger, O., P. Fischer y T. Brox (2015), ≪U-Net: Convolutional Networks for Biomedical Image Segmentation≫, *CoRR*, **abs/1505.04597**, `1505.04597[cs.CV]`.
12. Teichmann, M., M. Weber, J. M. Zöllner, R. Cipolla y R. Urtasun (2016), ≪MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving≫, *CoRR*, **abs/1612.07695**, `1612.07695[cs.CV]`.
13. Yakubovskiy, P. (2020), ≪Segmentation Models Pytorch≫, `https://github.com/qubvel/segmentation_models.pytorch`.
14. Zhou, J., B. Huang, Z. Yan y J.-C. G. Bünzli (2019), ≪Emerging role of machine learning in light-matter interaction≫, *Light: Science & Applications*, **8**(1), págs. 1–7.