Challenge IMA205

Pierre Marchetti

April-May 2024

1 Introduction

A skin lesion is defined as a superficial growth or patch of the skin that is visually different and/or has a different texture than its surrounding area. Skin lesions, such as moles or birthmarks, can degenerate and become cancer, with melanoma being the deadliest skin cancer. Its incidence has increased during the last decades, especially in the areas mostly populated by white people.

The most effective treatment is an early detection followed by surgical excision. This is why several approaches for skin cancer detection have been proposed in the last years (non-invasive computer-aided diagnosis (CAD)).

The goal of this challenge is to classify dermoscopic images of skin lesions among eight different diagnostic classes:

1. Melanoma
2. Melanocytic nevus
3. Basal cell carcinoma
4. Actinic keratosis
5. Benign keratosis
6. Dermatofibroma
7. Vascular lesion
8. Squamous cell carcinoma

The method involved manually extracting the ABCD features—namely asymmetry, border structure, color, and dimension features—of skin lesions for diagnostic purposes. These features were extracted according to criterias that someone would use when evaluating a skin lesion in real life. These features were inputted into a Convolutional Neural Network (CNN).

2 Data

2.1 Description of the data-set

We use a data-set of 25331 dermoscopic images of skin lesions with, when available, their relative segmentation and metadata (age, sex and anatomical position). Data has already been randomly split into a training-validation set (75 percent) and a test set (25 percent). We only have the classification (made by clinicians) of the training-validation set. The goal of the project is to estimate the correct class of each dermoscopic image in the test set.

2.2 Potential issues with the data-set

This dataset poses multiple challenges for this project:

- The data-set classes are unbalanced. This can be solved by feeding our CNN with the corresponding weights.
- Only a small fraction of the data-set includes ground-truth segmentations that are essential for extracting ABCD features accurately. We will then use different methods to segment the rest of the data-set.
- Since we need to segment the images we should consider the multiple difference between them during preprocessing (size, borders, ...)
- The data-set contains only 25331 labeled entries. While this seems significant, it is relatively small for image classification tasks. We can use data augmentation to increase this number (not used here because too computationally heavy)

3 Segmentation process

3.1 Preprocessing

The `preprocessing` function is crafted to prepare image data for detailed analysis by employing a series of image processing techniques. These techniques are designed to enhance image features, improve quality, and extract regions of interest (ROI) for future segmentation.

Cropping the Image

- **Purpose:** To focus on significant areas of the image by extracting the region of interest (ROI) based on brightness levels, thereby eliminating irrelevant parts like black borders that could hinder segmentation.
- **Method:** Uses a custom `crop_img` function that scans the image diagonally to dynamically set cropping boundaries at points where pixel brightness crosses a predefined threshold.

Inpainting

- **Purpose:** To remove small artifacts or hairs which could potentially interfere with segmentation.
- **Method:** Implements the `dullRazor` function, utilizing morphological transformations and inpainting techniques for spot correction.

Contrast Enhancement

- **Purpose:** To enhance the contrast, making it easier to distinguish lesions.
- **Method:** Applies CLAHE (Contrast Limited Adaptive Histogram Equalization), enhancing local image intensities without amplifying noise.

Median Filtering

- **Purpose:** To reduce noise created by hair removal while preserving the structural integrity of the image's edges.
- **Method:** Uses median blur to remove noise.

3.2 Segmentation

Overview

The `segmentation` function implements image segmentation using Otsu's thresholding method on the region of interest (ROI) of an image.

Methodology

Otsu's method is an automatic thresholding technique that derives an optimal threshold separating pixel values into two distinct classes. By minimizing intra-class variance and maximizing inter-class variance, this method effectively removes hairs and artifacts in grayscale images.

3.3 Postprocessing

The `postprocessing` function refines the binary mask obtained from image segmentation, applying several morphological operations to enhance the segmentation's accuracy and utility.

Remove Small Holes

- **Purpose:** Enhances the mask by filling in small holes within segmented regions, improving their integrity.
- **Method:** Employs `morphology.remove_small_holes` to fill holes below a predefined size.

Remove Small Objects

- **Purpose:** Reduces noise by eliminating significant objects from the mask created by artifacts or black borders that still remain after preprocessing.
- **Method:** Uses `morphology.remove_small_objects` with a threshold of 3500, and adjusts the threshold to 1000 if no objects remain (to prevent big corner objects while keeping small lesions.

Morphological Operations

- **Purpose:** Smooths object edges and separates close objects to clarify segmentation boundaries and enlarges object in the mask slightly to compensate for earlier reductions.
- **Method:** Applies `morphology.opening` and `morphology.dilation` with a disk of size 12.

Convex Hull Transformation

- **Purpose:** Simplifies object shapes by enveloping them in the smallest convex shape to better cover the whole lesion.
- **Method:** Implements `morphology.convex_hull_image` with coordinate adjustments.

Recovering Original Mask Shape

- **Purpose:** Aligns the processed mask with the original image dimensions for proper analysis and display.
- **Method:** Adjusts mask pixels to match the original image's dimensions using the custom function `recover_original_mask_shape`.

3.4 Second method of segmentation using U-Net

U-Net is a convolutional network architecture for detailed image segmentation. Its design is symmetric with a contracting path to capture context and an expanding path to enable precise localization.

Contracting Path

The contracting path follows the typical architecture of a convolutional network:

- It consists of repeated application of two 3x3 convolutions (ReLU activation), followed by a 2x2 max pooling with stride 2 for downsampling.
- At each downsampling step, the number of feature channels is doubled to capture more complex features.

Bottleneck

This is the central part of the network, which includes:

- Two 3x3 convolutions with ReLU activation.
- A dropout layer is added for regularization, enhancing the model's ability to generalize.

Expanding Path

The expanding path includes:

- Upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels.
- Skip connections from the downsampling path are concatenated with the upsampled output to help the network localize, followed by two 3x3 convolutions (ReLU activation).

Final Layer

The final layer of the network consists of a 1x1 convolution, which maps the features learned by the network to the desired number of classes.

Model Compilation and Training

- **Optimizer:** The Adam optimizer with a learning rate of 1×10^{-4} is employed.
- **Loss Function:** Binary crossentropy is used, suitable for binary segmentation tasks.
- **Metrics:** The model uses accuracy to measure its performance.

Conclusion

The U-Net model's architecture allows for efficient segmentation of images, making it ideal for applications requiring detailed and precise segmentation, such as this task (although this method was not used in the rest of the project due to bad time management).

4 Feature extraction

Feature extraction was principally computed with the previous segmentations and according to:

- *Zortea et al. 'Performance of a dermoscopy-based computer vision system for the diagnosis of pigmented skin lesions compared with visual evaluation by experienced dermatologists', Artificial Intelligence in Medicine, 2014* [1]
- *N. C. Lynn and Z. M. Kyu. 'Segmentation and Classification of Skin Cancer Melanoma from Skin Lesion Images', 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Taipei, Taiwan, 2017* [2]
- *Messadi, M., Cherifi, H. et Bessaid, A. 'Segmentation and ABCD rule extraction for skin tumors classification', 2021* [3]

The features used are the following:

- **Asymmetry:** Asymmetry of shape (2 features) [1]
- **Border:** Index compact and fractal dimension (2 features) [3]
- **Color:** Typical hues counters divided by the lesion area A to white, red, light and dark brown, blue gray and black. (6 features) [2]
- **Dimension:** Diameter of the lesion (1 feature)
- **Other:** Geometric features (3 features) [1]

Many features could not be implemented in this project and some were too computationally heavy to use (cf un-used functions).

5 Classification

In order to predict the class of each dermoscopic image of the test set. These features were inputted into a Convolutional Neural Network (CNN). (these features could have been used to fine-tune a pre-trained CNN for better results)

5.1 Data Preparation

5.1.1 Reshaping Input Data

Input data variables `X_train` and `X_test` are reshaped to the form `[-1, 1, 15, 1]`, accommodating the requirements for a CNN with one channel, sequence length of 15, and one feature per sequence.

5.1.2 One-Hot Encoding of Labels

Labels `y_train` are one-hot encoded to transform them into a binary matrix representation, suitable for categorical classification across 8 distinct classes.

5.2 CNN Model Definition

5.2.1 Layer Configuration

- **Conv2D Layers:** The network includes two convolutional layers with ReLU activation. The first layer uses 32 filters, the second uses 64 filters, each with a 3x3 kernel size.
- **MaxPooling2D Layers:** Each Conv2D layer is followed by a 2x2 max pooling layer, reducing feature map dimensionality and computation.
- **Flatten and Dropout:** The feature maps are flattened into a vector and passed through a dropout layer with a dropout rate of 0.5 to prevent overfitting.
- **Dense Layers:** A dense layer with 64 neurons is followed by an output layer with 8 neurons, employing a softmax activation function to output class probabilities.

5.2.2 Model Compilation

The model is compiled with the Adam optimizer (learning rate = 0.001), using categorical crossentropy as the loss function, and accuracy as the performance metric.

5.3 Training the Model

The model is trained on the processed training data for 5 epochs, with a batch size of 16. Class weights are used to adjust for imbalanced classes, and a validation split of 0.2 is employed to monitor model performance on unseen data.

6 Conclusion

While the current phase of the project highlighted several key areas for improvement (data augmentation, more robust segmentation, feature extraction, CNN fine-tuning) it also set a solid foundation for future enhancements.

7 Personal Note

I would have liked to have made better use of my time on this project, as I feel I did not take as much care as I should have on certain parts due to bad time management.

References

[1] Maciel Zortea and Thomas R. Schopf and Kevin Thon and Marc Geilhufe and Kristian Hindberg and Herbert Kirchesch and Kajsa Møllersen and Jörn Schulz and Stein Olav Skrøvseth and Fred Godtliebsen (2014) *Performance of a dermoscopy-based computer vision system for the diagnosis of pigmented skin lesions compared with visual evaluation by experienced dermatologists*, Artificial Intelligence in Medicine.

[2] Lynn, Nay Chi and Kyu, Zin Mar (2017) *Segmentation and Classification of Skin Cancer Melanoma from Skin Lesion Images*, 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT).

[3] Messadi, Mahammed and Cherifi, Hocine and Bessaid, Abdelhafid (2021) *Segmentation and ABCD rule extraction for skin tumors classification.*