Automatic Gleason Grading

L. Pagano, E. Natuzzi, D. K. Russica
Dipartimento di Elettronica, Informatica e Bioingegneria
Politecnico di Milano, Milan, Italy
{luca1.pagano, ernesto.natuzzi, danielekota.russica}@mail.polimi.it

***Abstract*— Prostate Cancer (PCa) is the sixth most common and second deadliest cancer among men worldwide [1]. The aggressiveness of prostate cancer is measured by Gleason grading, a system based on the appearance of cancer cells [2]. The grading is usually performed via visual inspection (with a microscope) of the prostate tissue sample by expert pathologists. However, this is a time-consuming task and suffers from very high inter-observer variability. Automatic computer-aided methods have the potential to improve the speed, accuracy, and reproducibility of the grading process. Therefore, this project aims at the automatic Gleason grading of prostate cancer from Hematoxilin and Eosin (H&E) stained histopathology images through the use of a neural network. Additionally, from the the experimental evaluation we reach a IOU Score of 79.81 ± 0.163 and F1 Score of 0.8765 ± 0.124.**

I. Introduction

The Gleason grading system is used to help evaluate the prognosis of men with prostate cancer using samples from a prostate biopsy. A Gleason score is given to prostate cancer based upon its microscopic appearance. Cancers with a higher Gleason score are more aggressive and have a worse prognosis. Pathological scores range from 1 to 5, with higher numbers indicating greater risks and higher mortality. In this context common clinical practice is to microscopically examine the biopsy specimen for certain "Gleason"patterns. In particular, these Gleason patterns are associated with the following features, shown in fig. 1:

- **Pattern 1**: The cancerous prostate closely resembles normal prostate tissue. The glands are small, well-formed, and closely packed. This pattern corresponds to a well differentiated carcinoma.
- **Pattern 2**: The tissue has well-formed glands similarly to "Pattern 1", but with larger and more abundant tissue between them. This pattern also corresponds to a moderately differentiated carcinoma.
- **Pattern 3**: The tissue has recognizable glands, but the cells are darker. At high magnification, some of these cells have left the glands and are beginning to invade the surrounding tissue or having an infiltrative pattern. This pattern corresponds to a moderately differentiated carcinoma.
- **Pattern 4**: The tissue has few recognizable glands. Moreover, many cells are invading the surrounding tissue in neoplastic clumps. This pattern corresponds to a poorly differentiated carcinoma.
- **Pattern 5**: The tissue shows few or no recognizable glands. There are often just sheets of cells throughout the surrounding tissue. This pattern corresponds to an anaplastic carcinoma.

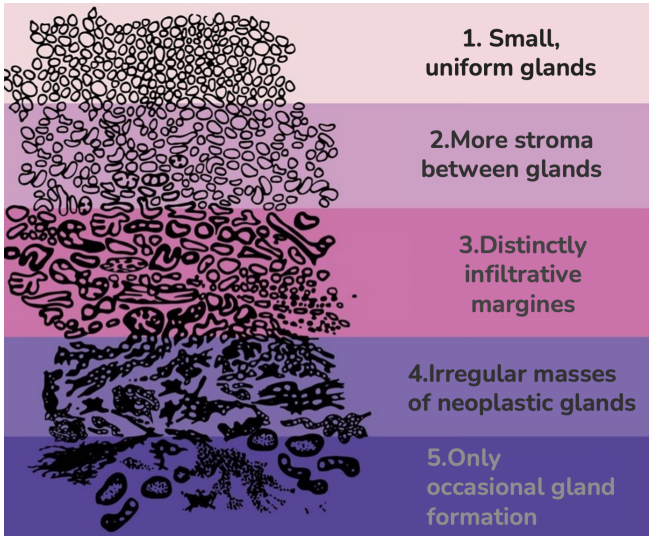

Fig. 1: High-level description of the five Gleason patterns

As mentioned above, the Gleason grading process is usually performed manually by expert pathologists. However, manual characterization of numerous patients is very time expansive, laborious, and prone to mistakes. Therefore, our project aims to develop a Deep Learning method to automate the Gleason grading while freeing pathologists from onerous workloads and results variability.

II. Proposed Solution

This section describes in detail the dataset, the preprocessing needed to adapt it to our needs, and how the model was trained to solve our problem.

A. Dataset Description and Preprocessing

The dataset [3] employed in this project consists of a set of Tissue Micro-Array (TMA) images, divided in two subsets: one having labels, for training purposes, consisting of 244 images, and one without them, for testing, made of 86 images. Each TMA image is annotated pixel-wise by several expert pathologists. Most of the images feature 6 annotations, each created by an experienced doctor. Labels pixels range from 0 (background) to 6, where 1-6 represents the Gleason grades. Then, we proceeded to rename the entire dataset with the following convention:

- maps: `slideXXX_coreYYY_classing_nonconvex.png` → `sXXX_cYYY_MAPn.png`
- images: `slideXXX_coreYYY.jpeg` → `sXXX_cYYY.png`

Subsequently, we checked for a size mismatch between the images and their corresponding map in the dataset. In the end, we came with the following results:

1) The annotations/images vary in size, with the dominant dimension equal to 5120x5120.
2) All the training images have at least one annotation (usually coming from pathologist '5', '4', or '6').
3) The corresponding label of an image has the same resolution.
4) For each label there is a corresponding original image: to exemplify, `s007_c137` and `s007_c145` do not have an image they refer to. Therefore, we decided to delete those annotations.

1) Combining the masks: Since we have 6 different mask for each training image, each one made by an expert pathologist, we have decided to combine them through the use of the STAPLE (Simultaneous Truth and Performance Level Estimation)[4] algorithm, implemented in the SimpleITK library.

STAPLE is an algorithm used in medical image processing, specifically in the field of medical image segmentation. It is commonly used for combining multiple segmentation results obtained from different algorithms or observers to improve the overall segmentation accuracy and reliability.

The primary goal of STAPLE is to estimate the true segmentation of an image while taking into account the performance characteristics of the individual algorithms or observers. It assumes that each algorithm or observer has a certain level of performance, which may vary in terms of accuracy and reliability. fig. 2 shows a comparison between the input image, the six corresponding masks and the STAPLE one.

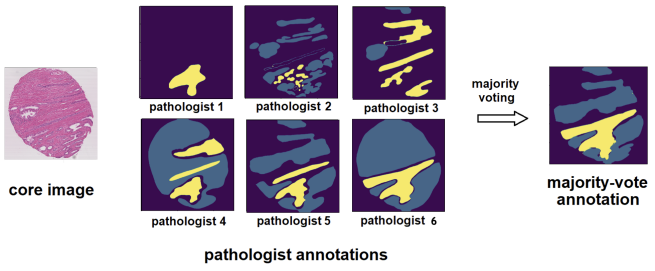

Fig. 2: Comparison between the original maps and STAPLE one

In case of "undecidedLabel", which is a pixel to which the algorithm does not know how to combine the pixel values, our code fills the information by copying the pixel from the *Pathologist1* label directory.

2) Dividing into patches: Given the high dimension of the images, we have decided to downscale them to ease the subsequent model training. Therefore, we first resize the images to 2048×2048 and divide them in 64 patches, with an overlapping percentage of 25%, based on [5]which adoperates the same procedure.
However, we noticed that the model could not properly learn from the labels due to the lack of context caused by the excessive fractioning of the images.

Based on this first attempt results, we decided to adopt a final strategy consisting of:

1) Resizing the images to a 1024×1024.
2) Dividing them into patches of 512×512, with a 50% overlap (fig. 3, fig. 4).

Indeed, having overlapping patches ensures that object boundaries that may not align perfectly with patch boundaries are captured, and they provide additional context, helping the model to better understand spatial relationships between regions and improving the segmentation accuracy.

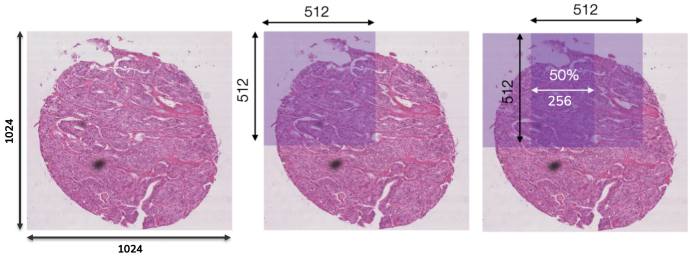

Fig. 3: `s001_c003` patient: 50% overlapping patches

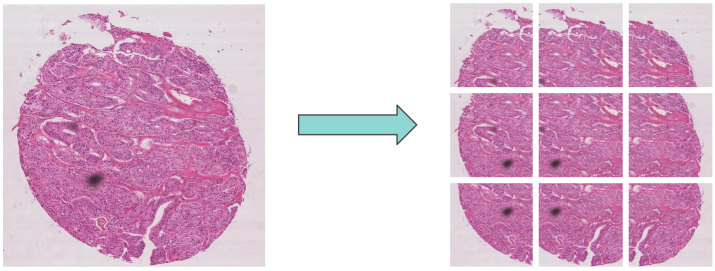

Fig. 4: `s001_c003` patient: from whole-slide image to image patches

B. Model Definition and Training

To implement the proposed solution and train the proposed DL model, we use Colab Free, which offers a NVIDIA T4 GPU with 12 GB RAM. Additionally the model was implemented in Keras (version 2.12.0), extended with Segmentation Model (version 1.0.1), for the evaluation metrics and training losses.

Considering our hardware limitations, for the model architecture we chose a UNet, with an EfficientNetB4 as the encoder (fig. 5). The EfficientNet [6] uses compound coefficient technique to scale up models in a simple, but effective, manner. In particular, instead of randomly scaling up width, depth or resolution, compound scaling uniformly

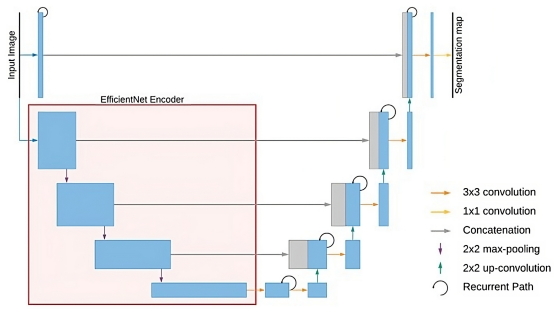

Fig. 5: UNet model architecture with EfficientNet as encoder

scales each dimension with a certain fixed set of scaling coefficients. Using the scaling method and AutoML, the authors of EfficientNet developed seven models of various dimensions, which surpassed the state-of-the-art accuracy of most convolutional neural networks, with better efficiency.

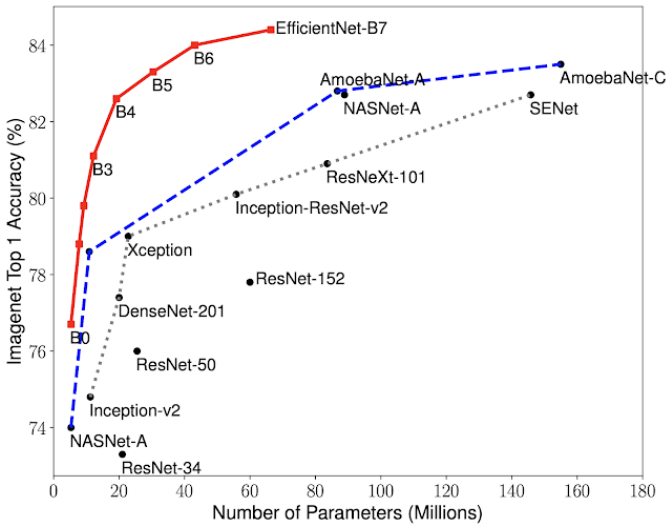

Fig. 6: Comparison between state-of-the-art DL model performances

Figure 6 shows the performance of EfficientNet compared to other network architectures. As shown, EfficientNetB7 obtained state-of-the-art performance on the ImageNet and the CIFAR-100 datasets. Indeed, it obtained around 84.4% top-1/and 97.3% top-5 accuracy on ImageNet. Additionally, the model size was 8.4 times smaller and 6.1 times faster than the previous best CNN model.

After some comparison with other alternatives, and keeping in mind the hardware limitations, we opted for EfficientNetB4 since it is the best compromise between performance and accuracy as showed in Figure 6.

At this point, to train our model we split the training portion of dataset (the one having the labels) with the following convention:

1) 80% Training.
2) 20% Validation.

Since each image consists of 9 patches, the training dataset has a total of 1755 images, and the validation dataset a total of 441.

After the model definition, we have focused on choosing the best possible loss function. From our evaluation, and given that the considered dataset was not considerably unbalanced, the Categorical Cross-Entropy loss achieved the best overall performance. Indeed, Categorical Cross-Entropy loss is traditionally used in multiclass segmentation tasks. It quantifies the degree of uncertainty in the model's predicted value for the variable and is defined as the sum of the entropies of all the probability estimates.

$$Entropy = -p_i log_b(p_i),$$
$$CrossEntropy = -\Sigma_{i=1}^{i=n} Y_i \ log_b(p_i),$$

where Y is the true label and p is the predicted probability.

C. Results

1) Training History: In the end, we trained our proposed model for 30 epochs with batches of 4 patches, and augmented the dataset exploiting the data augmentation of the `ImageDataGenerator`. The data augmentation parameters adopted are reported in algorithm 1

Algorithm 1: Data augmentation

```
data_gen_args = dict(shear_range=0.05,
                     zoom_range=0.2,
                     horizontal_flip=True,
                     vertical_flip=True,
                     fill_mode='reflect')
```

The results achieved are shown in fig. 7. In particular, it is visible how overfitting is reached around 17^{th}.

Fig. 7: Proposed model training history. From the left to the right the training and validation loss, and IOU metrics trends

2) Comparison between original and predicted segmentations: To evaluate the performance of our solution, we reconstructed the predicted images by adopting the algorithm proposed in [7]. It takes advantage of 50% overlapping patches, as well as Gaussian filters (fig. 8). Its benefit is visible from the visual results shown in fig. 9.

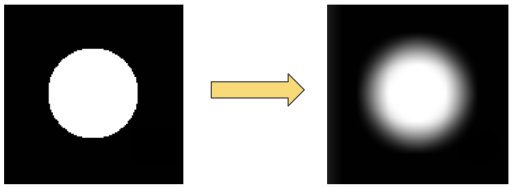

Fig. 8: Example of a Gaussian filter

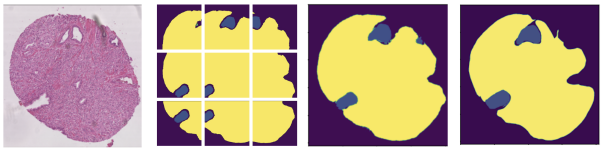

Fig. 9: Reconstruction process of the whole-slide segmentation from the predicted segmentation patches

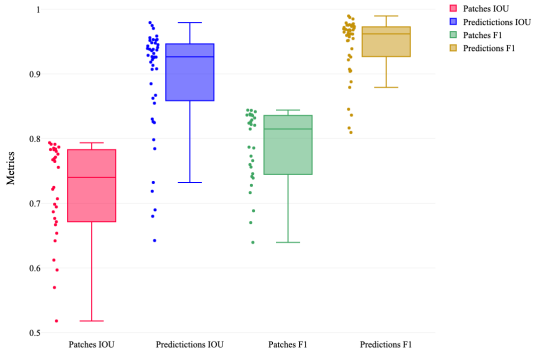

Fig. 10: BoxPlot comparison in terms of IOU and F1 score metrics between patched predictions and reconstructed segmentations

	IOU Score		F1 Score	
Metric	**Patches**	**Reconstructed**	**Patches**	**Reconstructed**
Best	0.8319	0.9792	0.8464	0.9844
Worst	0.5181	0.2069	0.5769	0.5378
Average	0.7211	**0.7981**	0.7313	**0.8765**
Std Dev	0.075	**0.163**	0.083	**0.124**

TABLE I: Comparison in terms of IOU and F1 score metrics between patched predictions and reconstructed segmentations

We can see a strong improvement in average metrics between patches and reconstructed images in fig. 10 and in table I. This can be explained by the fact that the overlap is used in reconstructing images and, therefore, more than one image is used concurrently to construct an image portion. This leads our reconstruction to be more accurate then the one of the single patches simply recombined, without any smoothing. However, from table I a decrease in the worst cases is visible. This is appreciated in fig. 11, where our algorithm infers wrongly the label shown in green". This behaviour is caused by the fact that our algorithm, in the case of indecision due to overlapping areas classified inconsistently by the model, chooses the most frequent and predominant label within the 9 patches of the entire image.

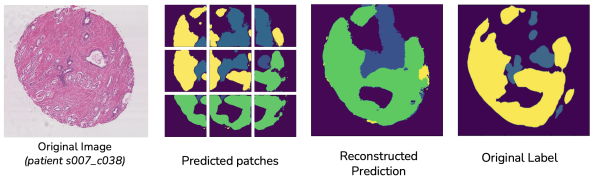

Fig. 11: Example of wrong reconstruction of the image

III. Conclusions

Although there is still room for improvement, we are satisfied with the average performance of our model, as it reached **IOU Score** of **79.81** ± **0.163** and **F1 Score** of **0.8765** ± **0.124**. A big step forward in terms of decreasing the variability of results could come from improved hardware. Specifically, an increase in the memory dedicated to the GPU would allow both the training of the model on the entire images, which would provide more context to the network, and, as well, the use of a more accurate model architecture as an EfficientNetB7.

References

[1] Wietske I Luining, Matthijs C F Cysouw, Dennie Meijer, N Harry Hendrikse, Ronald Boellaard, André N Vis, and Daniela E Oprea-Lager. Targeting psma revolutionizes the role of nuclear medicine in diagnosis and treatment of prostate cancer. *Cancers (Basel)*, 14(5), February 2022. Funding Information: Funding: This research was partially financed by Cancer Center Amsterdam, Amsterdam, the Netherlands. Publisher Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland.

[2] The Internet Pathology Laboratory for Medical Education. "male genital pathology", 2009.

[3] Danielle Walker. Gleason challenge 2019.

[4] Wells WM Warfield SK, Zou KH. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. 2004 July.

[5] Aashis Khanal and Rolando Estrada. Dynamic deep networks for retinal vessel segmentation, 2019.

[6] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[7] Dr. Sreenivas Bhattiprolu.