

Report Medical Imaging Project

Group 2: Salvatore Ceglia, Andrea Passarelli, Giovanni Paolo Tobia

Abstract—Auto-immune disorders are diseases related to fault on the immune system, which attacks and destroys healthy tissues in our body instead of protect them. HEp-2 cells are human epithelial cells, they are used for laboratory diagnostics for the detection of autoimmune diseases through indirect immunofluorescence.

The aim of the project is to deal with medical images and Multi-Task Learning, in order to improve the performance of autoimmune diseases diagnostic systems. In particular it was focused on the customization of the Multi-Task U-Net architecture, exploiting the EfficientNet-B4 as encoder backbone, performing simultaneously image segmentation and classification, to later compare the model performances, that will be discussed in the report.

I. INTRODUCTION

When some intruder invades the body, the immune system is activated to protect it from worse consequences; his goal is identify, kill and eliminate invaders that may pose a threat to health. Unfortunately, in some cases the immune system works improperly and attacks by mistake and repeatedly healthy cells belonging to the body itself, when it happens we speak generically of autoimmune disease. At least 80 types of autoimmune diseases have been identified, with some evidence suggesting that there may be more than 100 types and nearly any body part can be involved. Common symptoms can be diverse and transient, ranging from mild to severe, and generally include low grade fever and feeling tired [1].

The diagnosis of an autoimmune disease can be difficult to determine, it is based on a specialist medical examination and the execution of tests including:

- Antinuclear antibody test (ANA).
- Complete blood count (CBC).
- Erythrocyte sedimentation rate (ESR).
- Urinalysis.

Specific symptoms combined with specific blood markers may prove the presence of an autoimmune disorder [2].

II. THE PROBLEM

Autoantibodies to intracellular antigens, historically known as antinuclear antibodies (ANA), are serological biomarkers that have a central role in the diagnosis and classification of systemic autoimmune rheumatic diseases. The gold standard for ANA testing is Indirect Immunofluorescence (IIF) on human epithelial cells (HEp-2), due to its high sensitivity and

the large range of antigens that can be detected. However, the method suffers from numerous shortcomings, such as being subjective as well as time and labor intensive.

Computer Aided Diagnostic (CAD) systems have improved the reproducibility of the results and initiated a process of harmonization of this test. These systems are able to acquire, analyze and store the images in a fully automated way, providing quantitative expression of fluorescence intensity and minimizing the subjectivity of the interpretation of the fluorescence patterns [3].

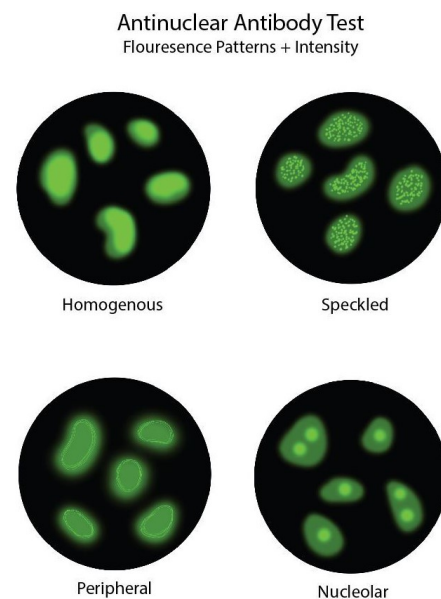


Fig. 1: ANA Patterns [4]

The main steps of the medical workflow [5] followed by the CAD systems that perform ANA testing are:

- 1) Intensity classification: the level of fluorescence can vary from negative to intermediate to positive.
- 2) Specimen segmentation.
- 3) Mitotic cell identification.
- 4) Cell Classification: different patterns can be recognized, the top-level division is showed on Fig. 1.
- 5) Specimen classification: majority voting among specimens cell identified classes.

The purpose of the project lies within these diagnostic systems, in particular, to find a Machine Learning approach to perform simultaneously 3 different tasks:

- 1) **Pattern Classification** among 7 classes: homogeneous, speckled, nucleolar, centromere, golgi, nuclear membrane (numem) and mitotic spindle (mitsp).
- 2) **Image Segmentation** among 2 regions: foreground and background.

Acknowledgement: this template was made starting from the *Template for LNT Seminar Report* available on <https://www.overleaf.com/>

GitHub repository: The source code and the results are available on <https://github.com/pasotech/Medical-Imaging>

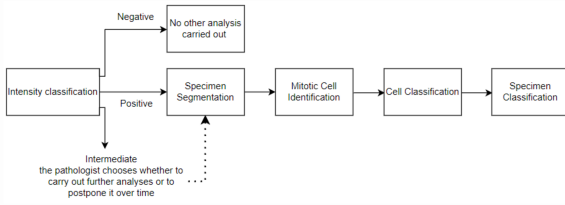


Fig. 2: Medical Workflow [5]

3) Intensity Classification among 2 classes: intermediate and positive.

The HEp-2 cell image dataset provided for the project consists of 252 patients, 4 image specimens for each patient and related masks. From each original specimen image and mask, 15 patches of 384 x 384 pixels have been already extracted, for a total of 15120 samples, as many masks, already labeled with the belonging pattern and intensity level, excluding the samples with negative intensity, because on them no other analysis are required.

From the pattern classification point of view, in the dataset, there is an unbalancing among the patterns to classify, with the following distribution:

- 3180 homogeneous;
- 3120 speckled;
- 3060 centromere;
- 3000 nucleolar;
- 1260 numem;
- 900 mistsp;
- 600 golgi.

The same happens for the intensity classification task: 8520 samples belong to intermediate class, while 6600 to positive one.

III. METHODS

The Neural Network model chosen for the project is based on the U-Net Architecture [6], which has already been widely used for tasks of this nature. From a structural point of view can be distinguished 2 main structures in the network:

The **Encoder** block, that has a constant reduction of image size with the help of the max-pooling layers of strides 2. There are also repeated convolutional layers with an increasing number of filters in the encoder. In this project, the backbone used for the U-Net encoder is the EfficientNet-B4 (with depth 5), implemented in the library from the reference underwritten [7], that is a common model used to reach a good combination of efficiency and accuracy. In particular, starting with input image dimensions of 3, 384, 384, the output dimensions of each block of the network are as follows:

- First Block with dimensions: 48, 192, 192;
- Second Block with dimensions: 32, 96, 96;
- Third Block with dimensions: 56, 48, 48;
- Fourth Block with dimensions: 160, 24, 24;
- Fifth Block with dimensions: 448, 12, 12.

The **Decoder** block, that once it has been reached, the number of filters in the convolutional layers start to decrease along with a gradual upsampling in the following layers all the way to the top. In this project, the dimensions of the Decoder

have been adapted to the dimensions of the Encoder in order to realize the skip connections.

The **skip connections** have a crucial role in this network, to preserve the loss from the previous layers so that they reflect stronger on the overall values and to produce better results and lead to faster model convergence, taking the output of each module of the Encoder as input of the correspondent module of the Decoder.

Starting from the U-Net model for segmentation, the approach used to improve the performances of the network is the Multi-Task Learning (MTL), that allows to solve multiple tasks at the same time while exploiting commonalities across tasks (shared representation), that in addition, grant to have more information on the problem and allows for a generalization of the system. In this case, to perform on the single patches the previously formulated tasks, as showed on Fig. 3, two auxiliary heads, made up of 2 Dense Layers, are added to the embedding layer of the U-Net (the blue block of the figure), each one used to perform a classification task.

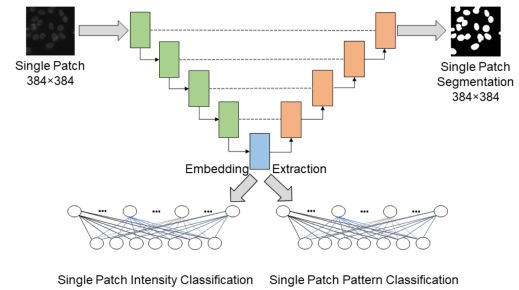


Fig. 3: Architecture of the Multi-Task U-Net

In order to perform these tasks, a specific Loss Function is assigned to each one:

- 1) Dice Loss, based on Dice Coefficient, or Binary Cross Entropy Loss, are used, and then compared, to perform the patch segmentation, since there are only two class to distinguish, the foreground, i.e. the region of interest, and the background.
- 2) Cross Entropy Loss is used to address the multi-class problem, i.e. the pattern classification.
- 3) Binary Cross Entropy Loss is used to address the two-class problem, i.e. the intensity classification.

With the aim of generalizing MTL systems and making them more robust with respect to loss weights in the literature, algorithms are used to train the coefficients of internal losses. In this case, the gradient normalization (GradNorm) [8] has been exploited, that automatically balances training in deep multitask models by dynamically tuning gradient magnitudes finding a common scale. It also matches or surpasses the performance of exhaustive grid search methods, despite only involving a single asymmetry hyperparameter **alpha**. Thus, what was once a tedious search process that incurred exponentially more compute for each task added can now be accomplished within a few training runs, irrespective of the number of tasks. The constant **alpha** sets the strength of the restoring force which pulls tasks back to a common training rate. In cases where tasks have various complexity, leading

to dramatically different learning dynamics between tasks, a higher value of alpha should be used to enforce stronger training rate balancing. When tasks are more symmetric, is not necessary an high alpha.

In order to evaluate the performance of the model in a more accurately, the cross-validation technique was applied, randomly dividing the patients into 5 folds (A, B, C, D, E) and inserting all the samples of the same patient into the chosen fold. In this way, at each training step, the dataset has been splitted into training set (80%), and test set (20%), i.e. 1 fold is used to evaluate the performance of the network trained with the samples of the other 4 folds.

IV. TRAINING AND RESULTS

In order to start the training session it's necessary to define the NN model hyperparameters:

- Batch size : 8
- Image channels : 3
- Dimension of images: 384x384
- The threshold used for binary segmentation and classification: 0.5
- Adam Optimizer with learning rate: 0.0001
- the number of output classes for the segmentation: 1
- Number of epochs: 30 or 35
- Segmentation Loss: Dice Loss or Binary Cross Entropy Loss
- alpha: 0.06 or 0.12

They have been performed several training session, in order to evaluate the Multi-Task U-Net model performances changing some hyperparameters.

A. First Training

- Number of epochs: **30**
- alpha: **0.06**
- Segmentation Loss: **Dice Loss**

Accuracy	FOLD A	FOLD B	FOLD C	FOLD D	FOLD E	Mean Accuracy
Mask	0.968	0.966	0.976	0.975	0.973	0.971
Label	0.844	0.843	0.831	0.817	0.832	0.833
Intensity	0.844	0.842	0.751	0.808	0.863	0.821

Fig. 4: Results of the first training for each fold

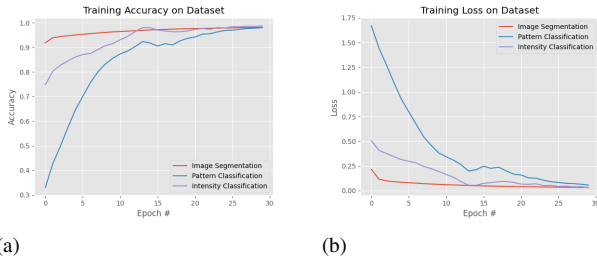


Fig. 5: Plots of the first training session on the accuracy (a), and on the loss (b)

B. Second Training

- Number of epochs: **30**
- alpha: **0.06**
- Segmentation Loss: **Binary Cross Entropy**

Accuracy	FOLD A	FOLD B	FOLD C	FOLD D	FOLD E	Mean Accuracy
Mask	0.968	0.967	0.976	0.975	0.973	0.971
Label	0.835	0.792	0.824	0.817	0.730	0.788
Intensity	0.821	0.851	0.742	0.808	0.779	0.800

Fig. 6: Results of the second training for each fold

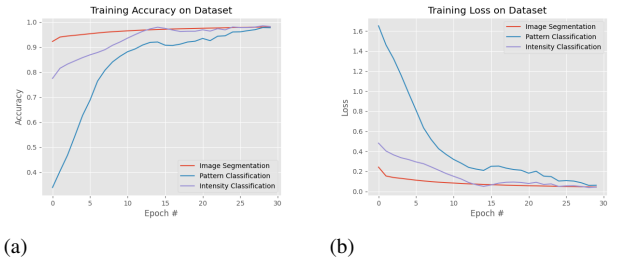


Fig. 7: Plots of the second training session on the accuracy (a), and on the loss (b)

C. Third Training

After the first two trainings, it is noticed that the network continues to learn until the last epochs, so the number of epochs has been raised to 35, to evaluate whether there will be an effective improvement of the network.

- Number of epochs: **35**
- alpha: **0.06**
- Segmentation Loss: **Dice Loss**

Accuracy	FOLD A	FOLD B	FOLD C	FOLD D	FOLD E	Mean Accuracy
Mask	0.968	0.967	0.977	0.975	0.975	0.972
Label	0.833	0.812	0.841	0.812	0.870	0.833
Intensity	0.801	0.832	0.756	0.807	0.910	0.821

Fig. 8: Results of the third training for each fold

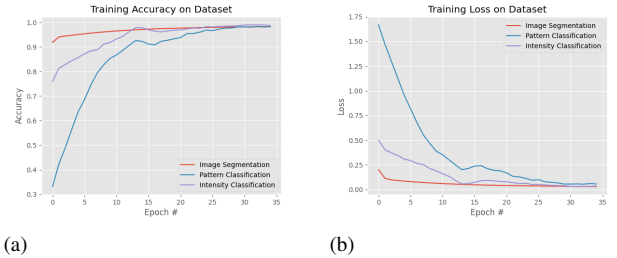


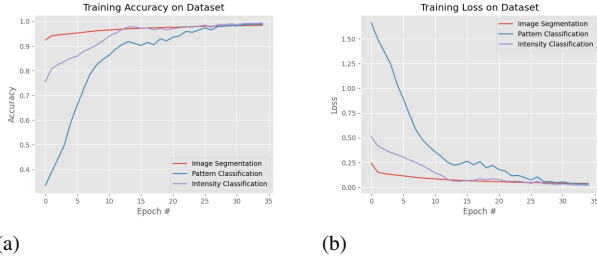
Fig. 9: Plots of the third training session on accuracy (a), and on the loss (b)

D. Fourth Training

- Number of epochs: **35**
- alpha: **0.12**
- Segmentation Loss: **Binary Cross Entropy**

Accuracy	FOLD A	FOLD B	FOLD C	FOLD D	FOLD E	Mean Accuracy
Mask	0.968	0.968	0.977	0.975	0.974	0.972
Label	0.854	0.790	0.822	0.809	0.837	0.822
Intensity	0.841	0.861	0.754	0.794	0.922	0.834

Fig. 10: Results of the fourth training for each fold



(a)

(b)

Fig. 11: Plots of the fourth training session on accuracy (a), and on the loss (b)

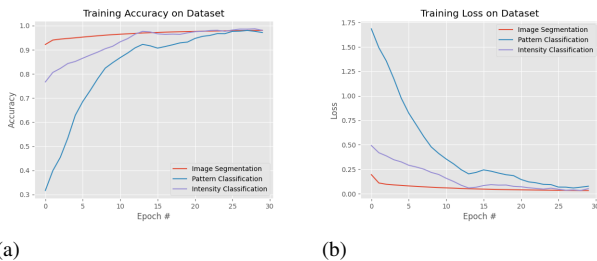
E. Fifth Training

After these first four trainings, it has been tried to slightly change the alpha parameter of the GradNorm, doubling it from 0.06 to 0.12, bringing the number of epochs back to 30, to evaluate if network performance improves.

- Number of epochs: **30**
- alpha: **0.12**
- Segmentation Loss: **Dice Loss**

Accuracy	FOLD A	FOLD B	FOLD C	FOLD D	FOLD E	Mean Accuracy
Mask	0.968	0.967	0.977	0.975	0.974	0.972
Label	0.831	0.820	0.833	0.805	0.864	0.833
Intensity	0.819	0.842	0.745	0.811	0.883	0.820

Fig. 12: Results of the fifth training for each fold



(a)

(b)

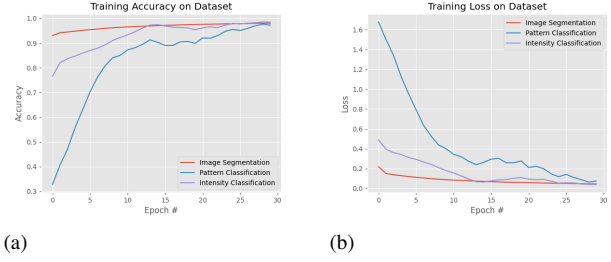
Fig. 13: Plots of the fifth training session on accuracy (a), and on the loss (b)

F. Sixth Training

- Number of epochs: **30**
- alpha: **0.12**
- Segmentation Loss: **Binary Cross Entropy**

Accuracy	FOLD A	FOLD B	FOLD C	FOLD D	FOLD E	Mean Accuracy
Mask	0.968	0.963	0.975	0.975	0.974	0.971
Label	0.844	0.767	0.844	0.809	0.858	0.824
Intensity	0.849	0.855	0.765	0.814	0.892	0.835

Fig. 14: Results of the sixth training for each fold



(a)

(b)

Fig. 15: Plots of the sixth training session on accuracy (a), and on the loss (b)

All the plots shown refer to the results of the second step of the cross-validation training, in which Folds A, C, D and E were used as train set and Fold B as test set.

V. CONCLUSION

In conclusion, from all the trainings, we can analyze that the performances on the segmentation task, are pretty the same in all the tried configurations. Instead, for the classification tasks, the performance are much lower, in all the approaches used. In particular, the behavior of the newtwork with Dice Loss is quite similar in all the configurations used, it does not change even if the parameters are changed. While the behavior of the newtwork with BCE Loss has a quite improvement (about 3.5% in both pattern and intensity classification) changing the number of epochs or the alpha parameter. It's possible to see the mean results of all the configurations in the table below.

Accuracy	Dice Epochs: 30 α : 0.06	Dice Epochs: 35 α : 0.06	Dice Epochs: 30 α : 0.12	BCE Epochs: 30 α : 0.06	BCE Epochs: 35 α : 0.06	BCE Epochs: 30 α : 0.12
Mask	0.971	0.972	0.972	0.971	0.972	0.971
Label	0.833	0.833	0.833	0.788	0.822	0.824
Intensity	0.821	0.821	0.820	0.800	0.834	0.835

Fig. 16: Compared performances of all trainings

The best performances can be observed in the newtwork with BCE Loss, 30 epochs and alpha 0.12, even if slightly compared to the network with Dice Loss, 30 epochs and alpha 0.06.

REFERENCES

- [1] Wikipedia. Autoimmune disease. https://en.wikipedia.org/wiki/Autoimmune_disease, Dec 2022.
- [2] Cleveland Clinic. Autoimmune Diseases. <https://my.clevelandclinic.org/health/diseases/21624-autoimmune-diseases>, Jul 2021.
- [3] Frontiers. Standardization and Quality Assessment Under the Perspective of Automated Computer-Assisted HEP-2 Immunofluorescence Assay Systems. <https://www.frontiersin.org/articles/10.3389/fimmu.2021.638863/full>, Feb 2021.
- [4] National Library of Medicine. Biochemistry. Antinuclear Antibodies (ANA). <https://www.ncbi.nlm.nih.gov/books/NBK537071/>, Nov 2022.
- [5] B.C. Lovell, G. Percannella, M. Vento, and A. Wiliem. Performance evaluation of indirect immunofluorescence image analysis systems. International Conference on Pattern Recognition (ICPR), 2014.
- [6] Wikipedia. U-Net Architecture For Image Segmentation <https://blog.paperspace.com/unet-architecture-image-segmentation/>, 2021.
- [7] Pavel Iakubovskii, Segmentation Models Pytorch https://github.com/qubvel/segmentation_models.pytorch#encoders, 2019.
- [8] Z. Chen, V. Badrinarayanan, C. Lee, A. Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. <https://arxiv.org/pdf/1711.02257.pdf>, June 2018.