Class Project 2 : Cell segmentation using time-sequence data

Machine Learning, CS-433

GARNIER Virginie, JAMAA Yassine, ROCHEPEAU Romain

ABSTRACT

Saccharomyces cerevisiae, also known as budding yeast is a very commonly used model in Biology research labs. In many applications of microscopy movie analysis, defining the borders of individual cells, also known as segmentation, is a critical step. Segmentation reveals itself to be quite difficult to automate, as you need to clearly identify the border of each cell. One of the current faced challenges is the detection of newly born cells, also know as buds Although this task is quite difficult to perform using standard image processing techniques, deep-learning methods have been shown to tackle it better. The state-of-the-art method, YeaZ, is very accurate on large cells, and only present errors on buds. We try addressing this problem by applying Temporal Attention mechanism. We hope that using time-series rather than a single image helps in the detection of buds. We base our model on the paper *Panoptic segmentation of satellite image time series with convolutional temporal attention networks* by Garnot, Vivien Sainte Fare, and Loic Landrieu (2021) [1].

With our method, the best result we obtained for the Intersection Over Union (IoU) was $\sim$ 92.5% when grouping our images at 7 consecutive times and predicting the last image of the group.

1 INTRODUCTION

Context and Motivation - Research Problem. As presented in the Abstract, segmentation is critical for a variety of Biological research. In the case of budding yeast, even the state-of-the-art algorithm still struggle with detecting buds. Although buds represent a whole cell whose identification is as important as the one of a big cell, they occupy only a fraction of the pixels of the whole colony, making the training of deep-learning methods harder. Overall, the goal is to produce a binary mask, where a pixel corresponding to a cell has value 1, and the background 0. Now, to address the bud prediction,

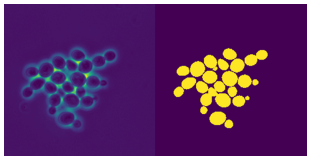

Figure 1: Input image in RGB and target segmentation mask.

we try using LTAE. As you can see in Figure 2, the buds growing significantly throughout the time frame. We hope that including temporal attention mechanism to a typical U-net will results in additional attention to buds.

Class 2 Project, CS-433, EPFL
2022.

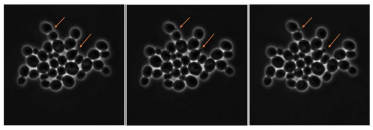

Figure 2: Time-series images of a colony. T0=0, T1=5min, T2=10 min

Contribution Summary. In this project, we adapted the code of [1] to work with our time-serie images. A major modification is that we group the images by different intervals, and apply U-TAE. Also, as we only have 2 classes (cell or background), we use a Binary Loss function, rather than a Cross Entropy.

2 BACKGROUND

Since its release, the paper "Attention is all you need"[2] has revealed itself to be a turning point in Deep-Learning. The Attention mechanism is applied in a model called the Transformer. This model enables a lower computational cost, due to averaging attention-weighted positions. This is counterbalanced by the introduction of the Attention mechanism, which takes into account position and context. The U-TAE model we apply to our data is a modified version of the Transformer.In [1], they tried to lighten the computational cost by different modification of the original algorithm, resulting in Lightweight Temporal Attention.

In this report, we tried applying the U-TAE to time-series of Yeast-cells images. The goal here is to see if this modified Transformer could be a good solution to detect individual cells. Another goal of ours it to manage detecting small budding yeast. We hope that the LTAE application enables a better detection of buds.

The U-TAE model is based on the typical Unet model, with Encoder and Decoder blocks. In addition to the spacial encoding and decoding, we also apply a temporal encoding and decoding with LTAE.

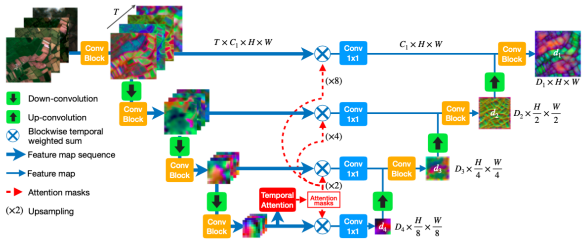

Figure 3: Schematic representation of the U-TAE model we used for segmentation [1].

The spacial encoder is composed of 4 identical layers. The Lightweight Temporal Attention is applied at the lowest resolution, at

the bottom of the encoder, after positional encoding of the images. The temporal encoding results is a single attention mask, that will be applied to all resolutions. The spatial decoder uses this mask and the lower resolution, to give rise to a single segmentation mask.

The Lightweight temporal Attention is a modified version of Multi-head Attention, first described in [2]. This mechanism gives our model the ability to learn which pixel is more important, depending on the time frame. For instance, it will learn that pixel [x1,y1] of the input is more important four our prediction at time Ti than at time Tj.

3 OUR METHOD

3.1 Design

In their work, Garnot et al. obtained better results thanks to the addition of temporal attention to their 3D U-Net. They thus showed that the time dimension was important for input that evolved over days and even months in their case.
The principle is the same for budding yeasts colonies, that develop a lot in just a few hours. Our method tries to implement this same time attention to budding yeast cells to be able to observe how important time really is when we try to segment cells.

The aim of our project was thus the adaptation of the existing U-TAE model to semantic cell segmentation. The model itself is truly complex and modifying it deeply was not our primary focus. We rather aimed to heavily base ourselves on the model and adapt it to cell images input, as well as trying to see what small modifications we could do to improve the prediction of the model for purposes of cell semantic segmentation, regarding the loss computation or the neural network parameters.

3.2 Optimization

Moreover, studying how time attention behaves in the case of cell segmentation was one of the aims of this project. In particular, we wanted to discover more about the number of consecutive images necessary and/or optimal to make the best predictions possible. To this end, we slightly modified the model to be able to obtain results for different time-grouping values as well as, within these groups, observe which image was predicted the best.

3.3 Implementation

To handle correctly the time dimension while loading our cell images correctly, we based ourselves on the names of our image files. Rather than converting our image to our *.json*s beforehand, we named our *.png* files with the set they belong to and the timestamp they correspond to. For instance, the fourth image of our third sequence of cell images will be named *"set_3_time_4_input.png"* and the corresponding mask (which is the ground truth) would be *"set_3_time_4_mask.png"*. By doing so, we were able to firstly build the images groups based on the temporal dimension of our images. We then use these pre-built time sets to retrieve the corresponding images where they are stored on our machine and feed them to the U-TAE model.

We then adapted some parts of the code to improve the predictions on a binary classification problem that we have here. The most important modification we applied to the model is its loss function. In the original case of 20 classes (or at least when we have several classes), the cross entropy loss gave good results, as described in the paper[1]. In our case however, we chose to replace it with the Binary Cross Entropy (BCE) with logits loss function after trying out several options. This loss function combines a sigmoid layer with the BCE loss, which is a more stable option than doing both seperately and is well adapted to our segmentation problem.
We took Adam as optimizer which is a popular algorithm in the field of deep learning. The benefits of using Adam on non-convex optimizations problems are mainly for its computational efficiency and its little memory requirements.

Finally, we created two new parameters, *Groupby* and *Mask_pos*, to respectively handle the number of images to take for temporal attention and the position of the ground truth mask for which we want to observe the predictions of our model, within a certain *Groupby*. For instance, if the *Groupby* is equal to 5, images are taken at times $[t-2, t-1, t, t+1, t+2]$ and *Mask_pos* can thus take a value $\in [0, 1, 2, 3, 4]$ in our code.

4 EXPERIMENTAL EVALUATION

4.1 Dataset

Our original dataset was comprised of 3 sequences of 256×256 images and an additional set of images with 450×256 px. Each sequence contained 121 different observations, taken 5 minutes apart from each other. As the last set of images contain two independent colonies that are clearly differentiable in each images, we decided to resize it to 256×256 px as well, by cropping the images in two as such manner that we generate two additional set of images from the set of oversized images.

To strengthen our model and to make our predictions better and more robust, we also performed data augmentation on the sequences of images. Indeed, the dataset we were given was small, so we aimed to prevent overfitting as well. On each set, we performed different types of modifications to produce new sequences to add to our dataset :

- The first set was rotated (90 degrees to right) and up-and-down flipped
- We adjusted the contrast of the second dataset (by a 0.3 factor) and its gamma (by a 0.5 factor)
- The third set was centered-crop and resized to a 256×256 shape, as all the other images.

4.2 Technical setup

UTAE is a deep and complex model which requires solid components, a GPU being mandatory to train efficiently. Unfortunately, there was some miscommunication and we asked too late for the access to the clusters, not letting us enough time to train our model on it. We therefore had to train the model on our own machines or on Google Collab for two of our group members, while the third one ran the model using a NVIDIA GeForce GTX 1660 Ti with Max-Q Design. With it, training our model for 5 *mask_pos* values, a *groupby* = 11 and 30 epochs took approximately 18 hours, while

Batch size	3
Learning rate	1e-4
Encoder width	[64, 64, 64, 128]
Decoder width	[32, 32, 64, 128]
Final out convolution	[32,1]
Input dimension for LTAE	256
Temporal encoder dropout	0.1
Scaled Dot Product Attention dropout 0.2	0.2
Number of heads for LTAE	16

Table 1: Parameters used for our U-TAE model.

Google Colab's GPU allowed the time taken to be 10hours approximately, but still not enough to train our model for 100+ epochs and several *groupby* values for instance.

4.3 Choosing parameters

We tried to improve our predictions by changing the batch_sizes, dropout or learning rate, starting from the parameters present in [1]. The optimal parameters for our model are summed in Table 1.

To obtain some exploitable results, we thus limited ourselves to 30 epochs for training. We first decided to focus on the *Groupby* and on the Mask position. We trained our model to find the best among various possibilities. We found optimal parameters to be [*Groupby* = 7, Mask position = 6]. We then only tuned the learning rate of the Adam optimizer while the *Groupby* and Mask position values stayed constant and equal to the optimal values we found for them. We found the best value to be $1e-4$.

If many models use accuracy to effectively choose the best parameters, we trained ours looking at the Intersection over Union (IoU), a primary metric to measure the accuracy of our model in image segmentation. IoU essentially measures the degree of overlap between our prediction and the ground truth, in our case the masks we were given. This quantity takes into account the spatial information of our images and maximizing this quantity allows to ensure that a prediction mask is reasonably close to the ground truth, which is the aim of our cell segmentation experiment. This thus allows to predict that our budding cells are located in the right place, as a shift in position of a cell, even if it is detected, would lower the IoU. On the other hand, as the cells are generally not taking up the entire spatial plan of the image, even a bad prediction would lead to rather high accuracy results. The changes in IoU are thus more significant in our case and we assumed that it was a better metric than accuracy, hence our choice to base our model tuning on IoU.
Unfortunately, we could not perform cross-validation, whether it was K-fold or LOOCV, due to the limitations detailed earlier. The optimal parameters were thus chosen based on the Iou of a unique validation set, composed of 120 images.

4.4 Results

We first studied the results for a groupby = 1 (See Figure 4.), hence when taking a single image as input and not a time-sequence. We obtained validation IoU values around 91% that did not seem to improve with the epochs number, staying constant throughout the experiment (besides outliers values). The accuracy values reached $\sim$ 99%.

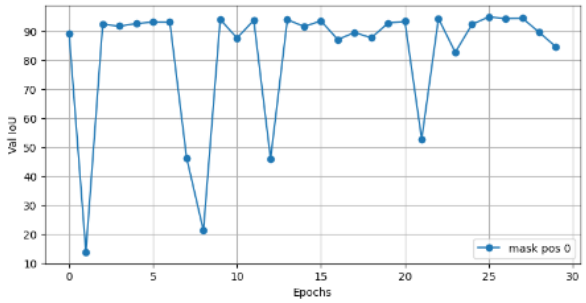

Figure 4: Validation IoU values for *groupby* = 1, for our 30 training epochs.

For *Groupby* values of 3,5,7,9 and 11, we observe that accuracy remains high for all our *groupby*s and *mask_pos*, above 98% to 99% after 30 epochs in most cases (See Figure 5.). However, our validation IoU values vary depending on the *groupby*. More specifically, *groupby* values of 5,7 and 11 seem to be reaching approximately 90% after 30 epochs, while 3 and 11 showed worse results, reaching validation IoUs of 85% after 30 epochs at most. Either way, the *mask_pos* parameter within a *groupby* generates noise, as some outliers can be seen in accuracy & IoU graphs.

Looking more specifically into the *mask_pos* parameter, we focused on *groupby* values of 7 and 9 and observe that the mean validation IoUs (on our 30 epochs) do not seem to follow any trend according to the *mask_pos* value. (see Figure 7)

For a *groupby* = 7, the final value we chose, the *mask_pos* that gave us the best results was 6. We thus ran a 100 epochs training with these parameters and obtained an increase in validation IoU, reaching values up to 92.5%. We can also see that the IoU reaches a plateau after 60 epochs approximately.

By looking at the prediction images (Figure 9), we can indeed observe that not all our cells are nicely predicted. The overall result is good, but some of the small cells specifically are not perfectly rendered in our predictions.

4.5 Analysis

First, the results of Figure 4. seem to corroborate our choice of IoU as the main metric to determine the performance of our model. For instance, while a *groupby* value of 9 shows accuracy values above 99% after 30 epochs, its IoU values are worse than for other groupby values such as 5 by 5 to 10%, when their accuracy is pretty similar. Changes in IoU are thus more observable and significant in our case than changes in accuracy, that are moreover very small.

An interesting thing to notice is that, besides outliers, our *groupby* = 1 seem to yield better predictions than larger *groupby* values for the first epochs. After our 30 epochs, some of our *groupby* values allow us to get similar predictions (most likely 5, 7 or 11, depending on the mask position) than for our single image input. Indeed the

Figure 5:
Left column : Validation accuracy of varying *mask_pos* within *groupby* values of 3, 5, 7, 9, 11**, for our 30 training epochs.**
Right column : Validation IoU of varying *mask_pos* within *groupby* values of 3, 5, 7, 9, 11**, for our 30 training epochs.**

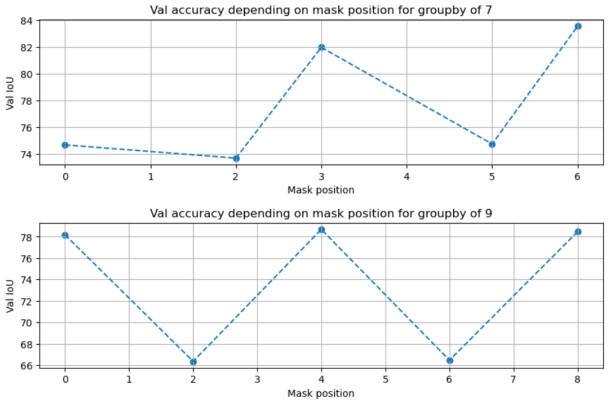

Figure 6: Validation IoU values averaged on our 30 training epochs, for several *mask_pos* within *groupby* values 7 and 9

increasing behaviour we'd expect from running on many epochs is not seen for a *groupby* value of 1, while the larger *groupby* values display this increase in their IoU values with the epochs. Our model

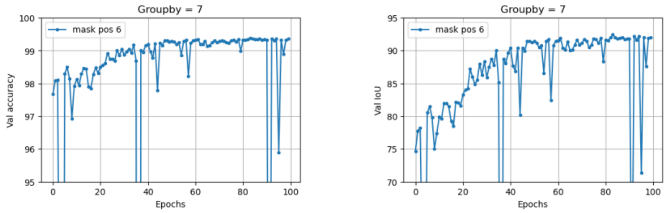

Figure 7: Accuracy and validation IoU values on 100 epochs, for a *groupby* = 7 and *mask_pos* = 6

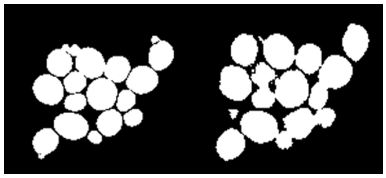

Figure 8: Ground truth mask (left) and the corresponding predictions of our model(right)

thus seem to need an adaptation period to this time dimension in order to be able to yield decent predictions for our budding yeast time-sequence data.
Regarding the *mask_pos* parameter, the data generated when varying it is noisy and while the general trend follows an increase, a lot of outliers can be observed in the validation IoU values. The behaviour of *mask_pos* parameter on *groupby* 7 and 9 (Figure 6.) tend to indicate that these might not be significant and more likely fluctuations due to the stochasticity of the training.

The predictions obtained when running more epochs (see Figure 7.), for our best parameters selection, indeed display an increase in both accuracy and most importantly Validation IoU. However, the plateau reached around IoU values of 92.5% seems to be indicating the limitations of temporal attention in our model. Even if the *groupby* = 7 improves the predictions, it is only by $\sim$ 2% in IoU compared to the *groupby* = 1 values. Time attention in our model is therefore not a great improvement to the use of a single image. Nonetheless, the rather smooth increasing behaviour of our values with a *groupby* = 7 indicates that taking time into account makes our model more robust than in the case of a *groupby* = 1, where the model does not seem to improve with the epochs.

If time attention has been useful to predict accurately the small and middle cells of a colony, (see Figure 8.) the really early-budding cells are not determined yet correctly determined. We can see that our predictions tend to indicate that a cell is developing in these zones, with some white pixels, but the full developing cell is not detected.

5 CONCLUSION

The use of temporal attention in our model did not give as much as an improvement we thought it would be. However, the results we obtained were still pretty good and temporal attention seemed to still have a level of importance in semantic cell segmentation. The combination of U-TAE model with time attention for the sole purpose of a binary semantic cell segmentation was maybe a bit too complex, hence the lack of improvement we obtained.

REFERENCES

[1] Vivien Sainte Fare Garnot and Loic Landrieu. 2021. Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks. *ICCV* (2021).

[2] Shazeer N. Parmar N. Uszkoreit J. Jones L. Gomez A.N. Kaiser L. Polosukhin I. Vaswani, A. 2017. Attention is all you need! *NeurIPS* (2017).