DeepCINAC: a deep-learning-based Python toolbox for inferring calcium imaging neuronal activity based on movie visualization.

Julien Denis[a,1,*], Robin F. Dard[a,1], Eleonora Quiroli[a], Rosa Cossart[a] and Michel A. Picardo[a]

[a] Aix Marseille University, INSERM, Institut de Neurobiologie de la Méditerranée (INMED), Turing Center for Living Systems, 13007 Marseille, France
[1] These authors contributed equally to this work
[*] To whom correspondence may be addressed. Email: julien.denis@inserm.fr

Keywords: calcium imaging, neuronal activity, deep learning, CNN, LSTM, CA1, development

Abstract

Two-photon calcium imaging is now widely used to indirectly infer multi neuronal dynamics from changes in fluorescence of an indicator. However, state of the art computational tools are not optimized for the analysis of highly active neurons in densely packed regions such as the CA1 pyramidal layer of the hippocampus during early postnatal stages of development. Indeed, the reliable inference of single cell activity is not achieved by the latest analytical tools that often lack proper benchmark measurements. To meet this challenge, we first developed a graphical user interface allowing for a precise manual detection of all calcium transients from detected neurons based on the visualization of the calcium imaging movie. Then, we analyzed our movies using a convolutional neural network with an attention process and a bidirectional long-short term memory network. This method reaches human performance and offers a better F1 score than CaImAn to infer neural activity in the developing CA1 without any user intervention. Overall, DeepCINAC offers a simple, fast and flexible open-source toolbox for processing a wide variety of calcium imaging datasets while providing the tools to evaluate its performance.

Highlights:

- A user-friendly GUI allows experimentalists to label easily cell activity from the visualization of calcium imaging movies
- A neural network trained on labeled calcium imaging data can faithfully infer neuronal activity of few hundreds of neurons.
- The toolbox offers a simple, fast and flexible method for processing a wide variety of calcium imaging data.

1. Introduction

In vivo calcium imaging is widely used to study neuronal microcircuits. Advances in imaging now allows for the simultaneous recording of increasingly large populations of neurons such as 10,000 neurons[1]. One difficulty resides in how to infer single neuron dynamics from changes in fluorescence of a calcium indicator. A challenge is therefore to offer an analytical tool that would be scalable to the wide variety of calcium imaging datasets while providing reliable analysis.

State of the art computational tools to infer neuronal activity (such as CaImAn[2,3]) are based on the deconvolution and demixing of fluorescence traces from segmented ROIs. However, an analysis based on the fluorescence traces even after a demixing process can still be biased by overlapping ROIs[4]. In this recent study from Gauthier and collaborators[4] analyzing calcium imaging data recorded in CA1 in adult rodents[5], 66% of the cells were reported as having at least one false transient and overall, among 33090 transients (from 1325 sources), 67% were considered as true, 13% as false and 20% were unclassified. Those contaminations increase the risk of misinterpretation of the data. Inferring neuronal activity from the developing hippocampus *in-vivo* is even more challenging due to several factors: 1- recurring network synchronizations are a hallmark of developing neuronal networks[6–10], which results in frequent cell co-activations, 2- the somata of pyramidal neurons are densely packed which results in spatial overlap, 3- Different calcium kinetics are observed in the same field of view (due to different cell types and different stages of neuronal maturation[11]). As a result, accurate inference of activity in the developing CA1 is not achieved by the latest analytical tools such as CaImAn. To circumvent this limitation, the use of the correlation between source and transient profiles was proposed[2]. However, we found out that in our dataset, the correlation score itself was not always a sufficient indicator to correctly infer calcium activity. To meet this challenge, we have developed a method based on deep-learning. Even though, several deep-learning based methods to infer neuronal activity from fluorescence traces already exist[12], none of them proposes a method directly using two-photon recordings.

Action recognition from videos has seen recent important progress thanks to deep learning[5,6]. Using a similar approach, we trained a binary classifier on calcium imaging movies (allowing us

to explore both the forward and backward temporal information among the whole sequence of video frames) to capture the fluorescence dynamics in our field of view and then predict the activity of all identified cells. It gave us the opportunity to take full advantage of the information contained in the movie in terms of dynamics and potential overlaps or other sources of contamination that might not be accessible when working only on fluorescence time course.

To train our classifier a ground truth was needed. To our knowledge, no datasets of calcium movies in the hippocampus during development with ground truth are available. The most accurate ground truth would be obtained from simultaneous targeted patch-clamp recordings and two-photon imaging on all the different hippocampal cell types with different calcium dynamics. This is technically difficult, time consuming and even more during development as the ground truth must be obtained from cells at various stages of maturation. As a result, we decided to base our ground truth on the visual inspection of raw movies using a custom-made graphical user interface (GUI). It gives the advantages to work on any kind of calcium imaging data and to offer an easy tool to benchmark methods that infer neuronal activity.
Our GUI offers a tool to precisely and manually detect all calcium transients (from onset to peak, which we think is the most reliable way to consider a cell as active without patch clamp based ground truth). We collected and combined a corpus of manual annotations from four human experts' representing 36.5 hours of two-photon calcium imaging recording on 11 mouse pups from 5 to 16 days old on CA1 region using GCaMP6s. Almost 80 % of the labeled data was used to train our model, while the rest was kept to benchmark the performance. Then, we processed our movies using a convolutional neural network with an attention process and a bidirectional long-short term memory network[13–15].

To evaluate our method, we used our ground truth as a benchmark. We found that our method reached human level performance and offers a better sensitivity and F1 score than CaImAn to infer neural activity in the developing hippocampus without any user intervention. Overall, DeepCINAC (Calcium Imaging Neuronal Activity Classifier) offers a simple, ergonomic, fast and flexible open-source toolbox for processing a wide variety of calcium imaging data while providing the tools to evaluate its performance.

2. Methods

In this section, we will describe all the necessary steps to build our deep learning neural network "DeepCINAC". This toolbox was developed to analyze our in vivo two-photon calcium imaging data acquired in the developing hippocampus (See § **Experimental procedure and data acquisition**). As a first step, we needed to set a ground truth that was established on the visualization of the recorded movie by three to four human experts (§ **Ground truth**). Then data are pre-processed (§ **Data pre-processing and feature engineering and model description**) and used to train the network (§ **Computational performance**). As a final step, we used labelled data to evaluate the performance of DeepCINAC (§ **Performance evaluation**). Tutorials and the source code are freely available online (§ **Toolbox and data availability**).

2.1 - Experimental procedure and data acquisition

All experiments were performed under the guidelines of the French National Ethic Committee for Sciences and Health report on "Ethical Principles for Animal Experimentation" in agreement with the European Community Directive 86/609/EEC (Apafis#81185-2018122110204650v3). To express the calcium indicator GCaMP6s in hippocampal neurons we intraventricularly injected a viral solution (Addgene, pAAV.Syn.GCaMP6s.WPRE.SV40, #100843-AAV1) at P0 in mouse pups (Figure 1A-B). Surgery to implant a cranial window above corpus callosum as well as acute two-photon calcium imaging were performed on the same day (Figure 1C-D). Imaging experiments was performed at least one hour after surgery on head fixed mouse pups aged from 5 to 16 days. 12500-frame-long image series from a 400x400 µm field of view with a resolution of 200x200 pixels were acquired at a frame rate of 10.6 Hz (Figure 1D).

We then motion-corrected the acquired images by finding the center of mass of the correlations across frames relative to a set of reference frames[16]. To detect cell contours, we used the segmentation method implemented in toolbox suite2p[17], that offers the best results for our data.

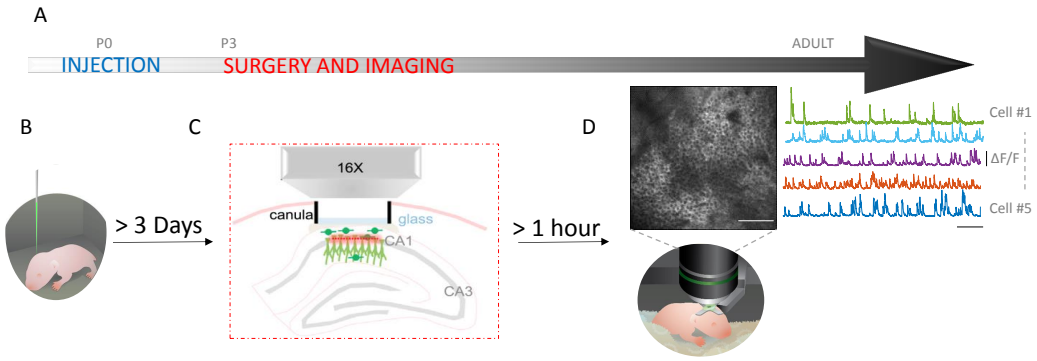

Figure 1: *Experimental paradigm.*

1A*: Experimental timeline.*

1B*: Intraventricular injection of GCaMP6s on pups (drawing) done at P0.*

1C*: Schematic representing the cranial window surgery.*

1D*: Top left: Imaged field of view. Scale bar: 100 µm. Top right: Activity of 5 random neurons in the field of view (variation of fluorescence is expressed as* Δf/f*). Scale bar 50 s. Bottom: Drawing of a head fixed pup under the microscope.*

2.2- Ground truth

To overcome the lack of ground truth based on patch-clamp recordings in the developing hippocampus, we designed a graphical user interface (GUI) that provides a visual inspection of each cell's activity. The GUI was developed using Python and Tkinter package (Figure 2). The GUI offers a set of functionalities allowing, for each cell, the combined visualization of the trace, raw movie and live calcium activity (Figure 2A), the source and transient profiles (as developed by Gauthier and collaborators[4], Figure 2B) as well as the correlation for any given transient profile with all overlapping sources. Through the GUI, we can also display the classifier results with the probability for a cell to be active at any frame. (see Figure 2C)

The ground truth was established based on two-photon calcium imaging from pups from 5 to 16 days old (see Table 1). They were labeled at least by two independent human experts. We then combined those labels and a final agreement was decided by three to four human experts. In addition, we trained an another classifier for interneurons using transgenic pups in which only interneurons express the indicator[18]. As previously described, interneurons' activity was labeled by three or four human experts and used to train a classifier specific to interneurons (see Table

1). After training our network on a first set of cells, we used the predictions obtained on new data to establish additional ground truth based on the mistakes made on those data. At least two human experts labeled segments of 200 frames containing the wrong predictions.

Table 1: *Data used to train and test both classifiers.*

	General classifier		Interneurons classifier	
	training/validation	testing	training/validation	testing
n FOV	13*	5	3	3
n animals	11*	4	3	3
n cells	103*	21	15	5
n transients	4939*	1516	2136	600
n frames	1051772*	26250	187500	62500
recorded time (hours)	29.2	7.3	5.2	1.7

*FOV: Field Of View, n: number of, *: including 2 simulated movies, representing 32 cells, 427 transients and 80000 frames.*

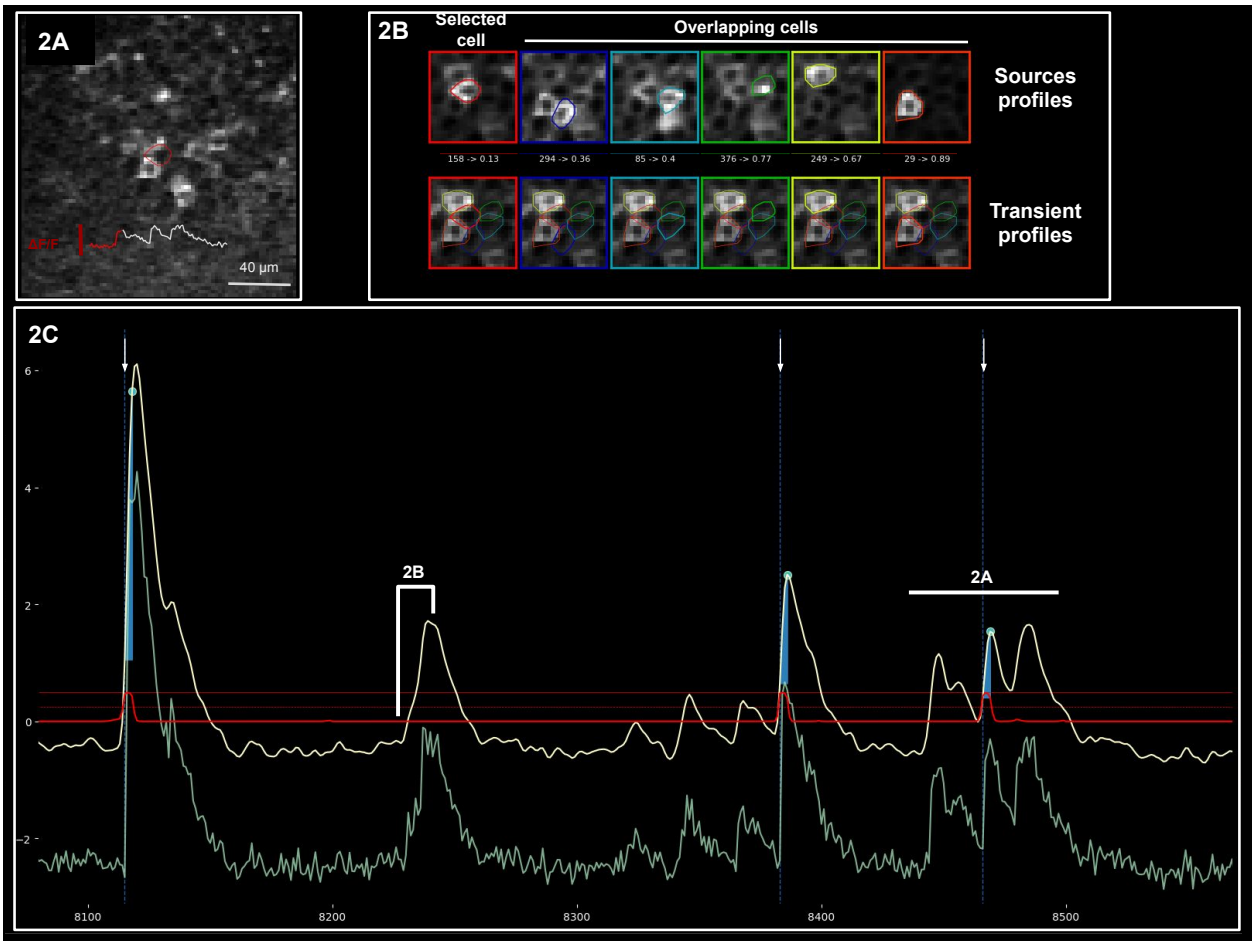

Figure 2: *Screenshot of the graphical user interface used to define the ground truth and visualize the network predictions.*

2A: *The image represents a single frame (extracted from a movie, 160 by 160 µm) centered on the cell of interest (red circle). The traces below represent the live variation of fluorescence of this cell between two chosen timestamps depicted by the white line labeled 2A in figure 2C*

2B: *Source profile / transient profile (such as in*[4]*) of selected region of interest (top left in red) and all overlapping regions of interest (from blue to orange). The transient displayed in this example is represented by the white line labeled 2B in figure 2C.*

2C: *Display of the raw fluorescence signal (in green) and a smoothed version (in yellow) over a certain range of frames. The dashed blue lines (white arrow) represent the transients' onset and the green circles the transients' peak; those can be easily added or removed by the user. We use the GUI to visualize prediction results. The plain bold red line represents the predictions of the classifier. The top plain red line indicates the value 1 on the y-axis, while the zero corresponds to the zero indicated on the y-axis. The dashed red line indicates the 0.5 value. The threshold value*

to set the cell activity can be selected by the user (default is 0.5). The blue area represents the frames during which the cell is considered active according to the threshold set.

2.3 - Data pre-processing, feature engineering and model description

Data pre-processing and feature engineering

Calcium movies in tiff format were split into individual tiff frames to be efficiently loaded in real time during the data generation for each batch of data feed to our classifier. For any given cell, a batch was composed of a sequence of 100 frames of 25x25 pixels window centered on the cell body. The length of the batch was chosen to fit for interneurons activity (rise and decay time). The window size was adapted to capture the activity of cells overlapping our target cell.

In a recording of 12500 frames, the number of transients ranges from 10 to 200 approximately. Thus the frames during which the cell is active (from onset to peak), represents a low percentage of the total data. Because manual labeling is time consuming, the data used as ground truth was limited in size. To overcome the issue of the imbalanced data and to enlarge our dataset, we used the following three approaches:

#1: Data augmentation[19]: temporal and spatial data augmentation were used. Temporally such as each block of 100 frames was overlapping with each other using a sliding window of 10 frames of length, and spatially using transformations such as flip, rotation or translations of the images.

#2: Simulated data: To overcome the low percentage of frames with a fake transient due to overlap, we have simulated calcium imaging movies with a higher rate of overlapping activity than our dataset (an example of artificial movie is available online on our gitlab page, alongside the source code: https://gitlab.com/cossartlab/deepcinac)

#3: Data stratification: In order to balance our data, we used data augmentation on selected movie segments (underrepresented segments), and excluded others (overrepresented segments) from the training data set. After data stratification we obtained approximately 60% of our movie segments containing at least one real transient, 30% at least one fake transient without real ones and 10% without transients. We were then able to be more precise over the proportion of segments with multiple transients or cropped transients. We gave higher weights to segments containing fake transients in order for the network to adjust the accuracy accordingly.

The data augmentation was done online, meaning that the transformations were done on the mini-batches that our model was feeding with. This allowed avoiding memory consumption and generating a dataset on multiple cores in real time.
The data augmentation and stratification were used only to produce the training dataset and not the validation dataset.

Model description

We designed a joint model combining a forward-pass long short term memory (LSTM), a backward-pass LSTM and convolutional neural network (CNN) features. In order for our bi-directional LSTM to focus on relevant information, we reinforced it by an attention process at the stage of encoding similar to previous work[20,21]. Our model was designed using Python and Keras library[22].

Our model takes three inputs each representing the same sequence of 100 frames (around 10 seconds of activity). Each frame had dimensions of 25x25 pixels, centered around the cell of interest, whose activity we want to classify. The first input has all its pixels set to zero except for the mask of the cell of interest (cell activity). The second input has all its pixels set to zero except for the mask of the cells that intersect the cell of interest (overlapping activity). The final input has the cell of interest and the one intersecting it pixels set to zeros (neuropil activity). That way, the model has all the information necessary to learn to classify the cell's activity according to its fluorescence variation.

We used dropout[23] to avoid overfitting, but no batch normalization. The activation function was swish[24]. The output of the model was a vector of length 100 with values between 0 and 1 representing the probability for the cell to be active at a given frame of the sequence. The loss function was binary cross-entropy and the optimizer was RMSprop.

2.4 - Computational performance

Classifier training

We trained our final general classifier version over 22 epochs and kept epoch 19 that gave the best performance (based on our benchmark). We trained it on Linux-based HPC cluster where 10 CPUs (Intel(R) Xeon(R) CPU E5-2680 v3), 320 Go of RAM and 2 bi-GPU NVIDIA Tesla K80 were allocating for the processing task. It lasted seven days (seven hours by epoch). We trained our interneurons specific classifier over 25 epochs and kept epoch 15 (based on our benchmark). We trained it on Linux-based workstation with one GPU (NVIDIA® GeForce GTX 1080), 12 CPUs (Intel Xeon CPU W-2135 at 3.70 GHz), and 64 GB of RAM. It lasted six days (six hours by epoch).

Classifier prediction

Using Linux-based workstation with one GPU (NVIDIA® GeForce GTX 1080), 12 CPUs (Intel Xeon CPU W-2135 at 3.70 GHz), and 64 GB of RAM, the time to predict the cell activity on a movie of 12500 frames was on average 13 sec, so around 3.5 hours for a 1000 cells. Similar performance was achieved using google colab.

2.5 - Performance evaluation

Descriptive metrics: sensitivity, precision, F1 score

We evaluated the performance of our classifiers, which predict for each frame if a cell is active or not. We chose to measure the recall (also called sensitivity) and precision (also called positive predictive value) values, as well as the F1 score that combine precision and recall into a single metric defined as the harmonic mean of precision and recall[25]. Because we have a skewed dataset (cells being mostly inactive), we choose not to use the accuracy that would not be a suitable measure. To define putative transients, we use the change of derivative on a smooth fluorescence time-course to detect all onsets and peaks. We considered as a putative transient all segments between an onset and the immediately following peak. Since the output of our binary classifier is the probability for a cell to be active at a given frame, we considered that a transient was predicted as true if at least one of its frame was predicted as active. On this basis we were then able to compute the recall (defined as the proportion of real transients that were detected) and the precision (defined as the proportion of detected transients that are real transients).

Detection of overlap activity

We specifically questioned whether our classifier was able to predict as false a transient that is due to the activity of an overlapping cell (as seen in Figure 2A and 2B). To do so, for all pairs of overlapping cells (with an intersected area of at least 15% of the highest area of the two cells), we computed their transient profiles (Figure 2B) over all putative activations (all rise time over the full recording) and then calculated the Pearson correlation with their respective cell source profile. If the correlation was superior to 0.7 for the first cell while inferior to 0.2 in the second one, we considered that the transient was a true activation of the first cell leading to a false transient in the second one. Finally, we evaluated whether the classifier could classify the putative transient of the second cell as false (with a prediction < 0.5).

Comparison with CaImAn

We compared our classifier performance against state of the art computational tool CaImAn. A transient was considered as detected by CaImAn, if at least one spike was inferred during the rise time of the transient.

2.6 - Toolbox and data availability

The source code is available on gitlab (https://gitlab.com/cossartlab/deepcinac). The page includes full description of the method, a user manual, tutorials and test data, as well as the settings used. A notebook configured to work on google colab is also provided, allowing for the classifier to run online, thus avoiding installing the necessary environment and providing a free GPU.

Our toolbox has been tested on windows (v7 Pro), Mac Os X (MacOS Mojave) and Linux Ubuntu (v.18.04.1)

3. Results

3.1 Benchmarks :

We first evaluated our general classifier on putative pyramidal neurons (Figure 3A) and putative interneurons (Figure 3B). On 18 putative pyramidal neurons, the median recall was 0.85 (interquartile range 0.804–0.957) (Figure 3A1), the median precision was 0.915 (interquartile range 0.845-0.962) (Figure 3A2) and the median F1 score was 0.894 (interquartile range 0.84-0.917) (Figure 3A3). On 5 putative interneurons, the median recall was 0.826 (interquartile range 0.75–0.94) (Figure 3B1), the median precision was 0.88 (interquartile range 0.806-0.924) (Figure 3B2) and the median F1 score was 0.844 (interquartile range 0.768-0.915) (Figure 3B3).

We asked whether our interneuron specific classifier would reach higher performance than our general classifier to predict interneurons activity. To do so, we evaluated our interneurons classifier (Figure S1). From the 5 putative interneurons (Figure S1B), the median recall was 0.954 (interquartile range 0.915–0.962), the median precision was 0.824 (interquartile range 0.734-0.83) and the median F1 score was 0.884 (interquartile range 0.831-0.897). We then tested this classifier on pyramidal cells. From the 18 putative pyramidal neurons (Figure S1A), the median recall was 0.825 (interquartile range 0.692–0.928), the median precision was 0.814 (interquartile range 0.657-0.909) and the median F1 score was 0.749 (interquartile range 0.659-0.855.

Finally, we evaluated CaImAn on the same cells and with the same metrics against the general classifier. From 18 putative pyramidal neurons (Figure 3A), the median recall was 0.535 (interquartile range 0.409–0.737), the median precision was 1 (interquartile range 0.967-1) and the median F1 score was 0.68 (interquartile range 0.58-0.806). On the 5 interneurons (Figure 3B), the median recall was 0.446 (interquartile range 0.365–0.532), the median precision was 0.977 (interquartile range 0.968-1) and the median F1 score was 0.535 (interquartile range 0.495-0.689).

Overall we built a general classifier that was able to infer neuronal activity from calcium imaging data with better performance than CaImAn. Additionally, we were able to build an interneuron-specific classifier that was more performant that our general classifier at inferring interneurons activity. We next asked whether our classifier was able to reach human level. To do so we

computed the evaluation metrics for each single human expert against the ground truth. We noticed some variability among the three human experts, as for the 8 putative pyramidal neurons labeled, the median recall values were 0.945, 0.882 and 0.916; the median precision values were 1, 0.829 and 0.989 and the median F1 scores were 0.958, 0.856, 0.921. On these 8 cells, the general classifier performance was 0.844, 0.943, 0.909 for recall, precision and F1 score respectively. This is the first proof of the ability of our classifier to detect cell activation at human level (Figure 3A).

Since we aimed at predicting as active all the frames included in the full rise time of the calcium transient (from onset to peak), we looked at the proportion of frames predicted as active in real transients. Using the general classifier, the median ratio of frames predicted among each real transient was 88.889 % (interquartile range 75-100) and 85.714% (interquartile range 66.667-100) for the 18 putative pyramidal cells (Figure S2A) and the 5 putative interneurons (Figure S2B) respectively. We demonstrated that DeepCINAC allows the detection of cell activation all along the rise time, giving us both the onset of cell activation and the duration of the rise time.

We tried to evaluate the extent to which our network was far from the ground truth and whether it was still possible to obtain better predictions. To do so, we plotted the distribution of the prediction scores for True and False Positive, and True and False Negative transients (see Figure S3). We observed that the median prediction value was 0.993 **(**interquartile range 0.961-0.998) and 0.002 **(**interquartile range 0-0.024) for true positives and true negatives respectively. For false positives, the median prediction value was 0.818 **(**interquartile range 0.699-0.935) and for false negatives 0.087 **(**interquartile range 0.014-0.244). Even though the majority of wrongly predicted transients is far from the 0.5 threshold, the predictive values are respectively higher and lower than the true negatives and true positives.

Since it was possible to improve the prediction of the classifier, we decided to correct its wrong predictions and feed them in the new training data set (see Methods). As a result, we improved the performance as illustrated by the increase of the F1 score from 0.81 to 0.894 between the first (v1) and the second version of the classifier (v2) (Figure S1).

3.2 Specific handling of overlap:

The overlap between cells leading to false transients was pointed out as a specific issue due to the analysis calcium traces from a demixing[4]. We asked whether our neural network would be able to distinguish real transients from increase of fluorescence due to the activity of an overlapping cell. Based on the visual inspection of imaged fields of view with numerous overlaps, we chose to specifically test the algorithm on calcium imaging data containing 391 cells segmented using CaImAn. Among those cells, we detected a total of 333 transients (fluorescence rise time) from 22 cells that were likely due to overlapping activity from a neighboring cell (see method for overlap activity detection). Among those transients, 96.40% were correctly classified as false by the general classifier, 74.47% were correctly classified as false by the interneuron specific classifier and 90.99% were correctly classified as false by CaImAn. We next asked if our results could be improved by the use of another segmentation method. To do so, we performed the same analysis on the exact same field of view using our classifier prediction on the segmented cells obtained from suite2p[17]. Among a total of 479 cells, a total of 2869 transients from 104 cells were likely due to the activation of an overlapping cell, 98.222% of them were correctly classified as false by the general classifier.

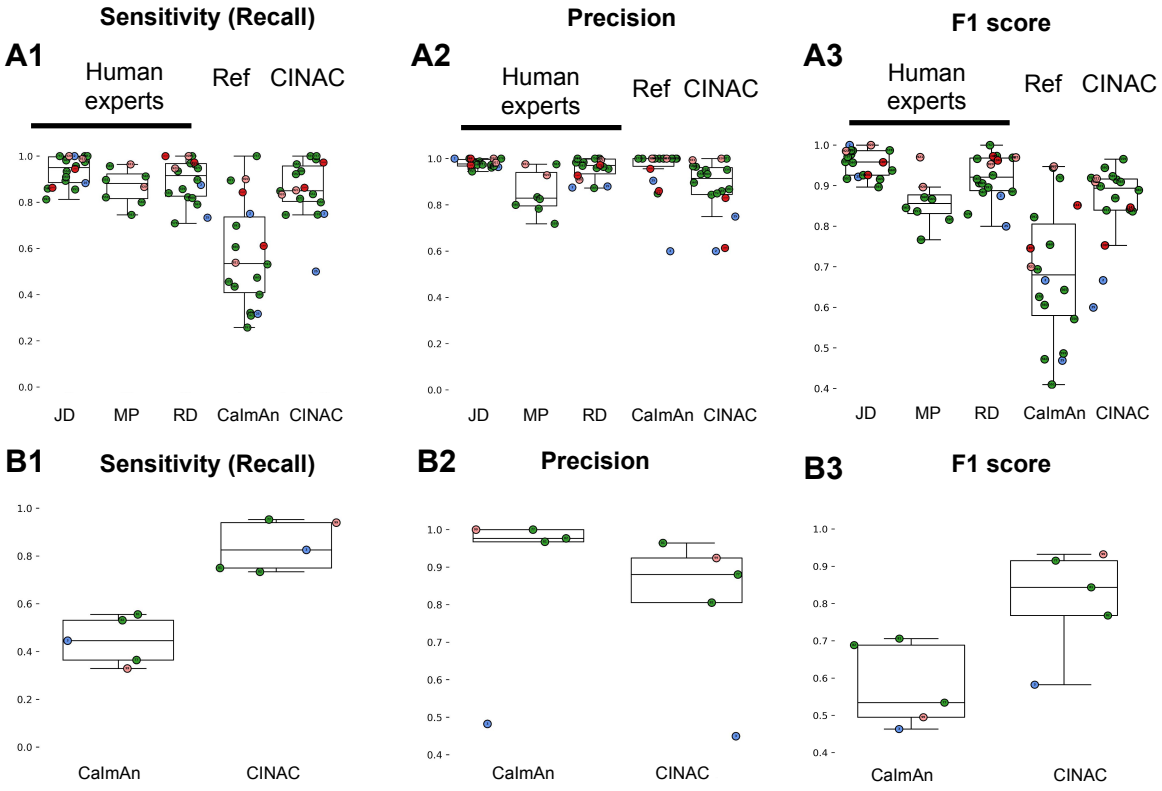

Figure 3: *Recall (sensitivity), precision and F1 score from the general classifier's predictions.*

Top panel: *Each graph represents the performance of three human expert (JD, MP, RD), our gold standard (Ref, CaImAn), as well as our general classifier CINAC on18 putative pyramidal cells.*

Bottom panel: *Each graph represents the performance of our gold standard (CaImAn), as well as our general classifier CINAC on 5 putative interneurons*

A1 and B1: *Represent the sensitivity (Recall)*

A2 and B2: *Represent the precision*

A3 and B3: *Represent the F1 score*

Each circle represents a cell, the number inside indicates the cell id. Each color represents a recording.

4. Discussion

Deep learning based method(s) to infer neuronal activity from 2-photon calcium imaging datasets use cellular fluorescence signals as inputs. Here we propose a method based on the visual inspection of the recordings. We will discuss the advantages and limitations of our approach.

Using the movie dynamics, we benefited from all the information available in the calcium imaging movie. This approach allowed us to not rely on a demixing algorithm to produce the neuron's traces. Instead, by working directly on the raw calcium imaging, our algorithm has learned to identify a transient and distinguish overlap activity from a real transient. DeepCINAC achieves better performance than CaImAn (at inferring neuronal activity in the developing hippocampus) and more importantly is able to achieve human performance level on some field of view and cells. We showed the capacity of our classifier to distinguish activity due to overlap. Visual inspection of the prediction through our GUI suggests that our network is also able to handle fake transients due to X and Y movements or neuropil activation. By avoiding the use of any threshold to select transient over fluorescence time-course, the classifier is able to detect: 1- small amplitude transient, 2- transients occurring during the decay of another one, 3- summations. Moreover, the absence of threshold to detect transient allows the classifier to deal with changes in fluorescence baseline, which can be due to photo-bleaching. Importantly, DeepCINAC still performed well because activity was split into segments of 100 frames (around 10 sec).

Overall, our approach allowed us to create a classifier that generalizes across different developmental stages and different types of neurons. As shown for interneurons, it is possible to train a specific classifier with a relatively small dataset containing specific data. This classifier performs better that our general classifier on interneurons, however it does not generalize well (with poor performance on pyramidal cells and on distinguishing overlaps). Altogether, we offer a flexible method that could be used with other indicators (GCaMP6m and 6f, GECO, GCaMP5...), different cell types, as well as single photon imaging data. Of important note, the performance of the classifier can be incrementally improved using a similar strategy to the one used for example by Tesla[26]. The strategy consists in training a first classifier based on a limited dataset. Then use this classifier on new data, manually identify its errors, correct it and use this new labeled data to feed the next classifier. The quantity of labeled data used to train the classifier is important but the quality is also an important factor to take into consideration, as many specific situations might

not be covered on a limited dataset. Using this philosophy, we could initiate a collaborative work to gather all the corrected errors of the classifiers on all available calcium imaging dataset.

Finally, we explored the range of values of hyperparameters in order to optimize the accuracy of the classifier. Thus, using our approach is simple. The labeling of data is time-consuming but the training does not need any parameters tuning. If a classifier has already been trained, then the prediction is straight forward. Neither tedious manual tuning of parameters is required, nor a GPU on a local device because we provide a notebook to run predictions on google colab (see Methods). The predictions are fast, with a run-time of around 13 seconds by cell for 12500 frames, meaning approximately 3.5 hours for 1000 cells. However, a GPU would be necessary to train the network on a big dataset.

Our approach is still limited by the need of a ground truth that remains challenging to obtain. Indeed, our ground truth is based on the visualization of the calcium movie whereas patch-clamp recordings would be necessary to validate this approach.

Already widely used by many calcium imaging labs[5,27–29], CaImAn offers a performing and functional analysis pipeline. Even though the complex fine tuning of CaImAn parameters on our dataset leads to a suboptimal spike inference from the model, we decided to compare CaImAn against our benchmarks.

Our benchmarks remain limited to a small number of cells for which we established our ground truth and may be extended to more cells. Notably, a future approach could be to use more realistic simulated data such as done in a recent work[30].

In the model we used, each cell was represented by a segment of the field of view, in our case a 25 by 25 pixels (50 μm by 50 μm) window that allows to cover the cell fluorescence and potential overlapping cells. Consequently, our network is able to generalize its prediction to recordings acquired with this resolution (2 μm / pixel). However, to be efficient on another calcium imaging dataset with a different resolution it would be necessary to train a new classifier adjusting the window size accordingly. Importantly, we trained our model on a selection of cell with valid segmentations; meaning that a cell is not represented by several contours. The inference performance of our classifier decreases on cells whose segmentation was not properly achieved. Since precise spike inference cannot be experimentally assessed on our data, we chose to infer the activity of the cell defined by the fluorescence rise time instead of inferring the spikes. However, with a ground truth based on patch-clamp recordings, we could adapt our method to

switch from a binary classification task to a regression task, predicting the firing rate at each frame.

Conclusion

We built DeepCINAC basing our ground truth on movie visualization and training the classifier with movie segments. DeepCINAC offers a flexible, fast and easy-to-use toolbox to infer neuronal activity from any kind of calcium imaging dataset, reaching human level. It provides the tools to measure its performance based on human evaluation. Currently, DeepCINAC provides two trained classifiers on CA1 two-photon calcium imaging at early postnatal stages; its performance can still be improved with more labeled data. In the future, we believe that a variety of classifiers trained for specific datasets should be available to open access.

Acknowledgments:
This work was granted access to the HPC resources of Aix-Marseille Université financed by the project Equip@Meso (ANR-10-EQPX-29-01) of the program « Investissements d'Avenir » supervised by the Agence Nationale de la Recherche.
This work was supported by the European Research Council under the European Union's FP7 and Horizon 2020 research and innovation program (grant no. 242842 and 646925). J.D. was supported by the Fondation pour la Recherche Médicale (grant no. FDM20170638339). M.P was supported by the Fondation pour la Recherche Médicale (grant no. ARF20160936186).

Author Contributions

Performed surgeries: RD
Recorded calcium imaging movies: RD
Concept & design: RD JD MP
Wrote code: JD
Labeled data: MP JD RD EQ
Wrote and edited the manuscript: RD, JD, MP, RC

Competing Interests statement

None

Bibliography:

1. Stringer C, Pachitariu M, Steinmetz N, et al. Spontaneous behaviors drive multidimensional, brainwide activity. Science. 2019; 364(6437):255-255.

2. Pnevmatikakis EA, Soudry D, Gao Y, et al. Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. Neuron. 2016; 89(2):285-99.

3. Giovannucci A, Friedrich J, Gunn P, et al. CaImAn an open source tool for scalable calcium imaging data analysis. Elife. 2019; 8:e38173.

4. Gauthier JL, Koay SA, Nieh EH, et al. Detecting and Correcting False Transients in Calcium Imaging. bioRxiv. 2018; :473470.

5. Gauthier JL, Tank DW. A dedicated population for reward coding in the hippocampus. Neuron. 2018; 99(1):179–193.

6. Provine RR. Ontogeny of bioelectric activity in the spinal cord of the chick embryo and its behavioral implications. Brain Res. 1972; 41(2):365-78.

7. O'Donovan MJ. Motor activity in the isolated spinal cord of the chick embryo: synaptic drive and firing pattern of single motoneurons. J Neurosci. 1989; 9(3):943–958.

8. Hanganu IL, Ben-Ari Y, Khazipov R. Retinal Waves Trigger Spindle Bursts in the Neonatal Rat Visual Cortex. J Neurosci. 2006; 26(25):6728-36.

9. Galli L, Maffei L. Spontaneous impulse activity of rat retinal ganglion cells in prenatal life. Science. 1988; 242(4875):90-1.

10. Ben-Ari Y, Cherubini E, Corradetti R, et al. Giant synaptic potentials in immature rat CA3 hippocampal neurones. J Physiol. 1989; 416:303-25.

11. Allene C, Picardo MA, Becq H, et al. Dynamic Changes in Interneuron Morphophysiological Properties Mark the Maturation of Hippocampal Network Activity. J Neurosci. 2012; 32(19):6688-98.

12. Berens P, Freeman J, Deneux T, et al. Community-based benchmarking improves spike rate inference from two-photon calcium imaging data. PLOS Comput Biol. 2018; 14(5):e1006157.

13. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. Handb Brain Theory Neural Netw. 1995; 3361(10):1995.

14. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997; 9(8):1735–1780.

15. Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need. ArXiv170603762 Cs

[Internet]. 2017 [cité 2018]; . Disponible sur: http://arxiv.org/abs/1706.03762

16. Miri A, Daie K, Arrenberg AB, et al. Spatial gradients and multidimensional dynamics in a neural integrator circuit. Nat Neurosci. 2011; 14(9):1150-9.

17. Pachitariu M, Stringer C, Dipoppa M, et al. Suite2p: beyond 10,000 neurons with standard two-photon microscopy. bioRxiv. 2017; :061507.

18. Melzer S, Michael M, Caputi A, et al. Long-Range–Projecting GABAergic Neurons Modulate Inhibition in Hippocampus and Entorhinal Cortex. Science. 2012; 335(6075):1506-10.

19. Perez L, Wang J. The effectiveness of data augmentation in image classification using deep learning. ArXiv Prepr ArXiv171204621. 2017; .

20. Bin Y, Yang Y, Shen F, et al. Describing Video With Attention-Based Bidirectional LSTM. IEEE Trans Cybern. 2018; :1-11.

21. Rémy P. philipperemy/keras-attention-mechanism [Internet]. 2019 [cité 2019]. Disponible sur: https://github.com/philipperemy/keras-attention-mechanism

22. Chollet F. Keras. 2015.

23. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014; 15(1):1929–1958.

24. Ramachandran P, Zoph B, Le QV. Searching for Activation Functions. ArXiv171005941 Cs [Internet]. 2017 [cité 2019]; . Disponible sur: http://arxiv.org/abs/1710.05941

25. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media; 2019.

26. Tesla Autonomy Day [Internet]. [cité 2019]. Disponible sur: https://www.youtube.com/watch?v=Ucp0TTmvqOE&feature=youtu.be&t=7517

27. Andalman AS, Burns VM, Lovett-Barron M, et al. Neuronal Dynamics Regulating Brain and Behavioral State Transitions. Cell. 2019; 177(4):970-985.e20.

28. Driscoll LN, Pettit NL, Minderer M, et al. Dynamic Reorganization of Neuronal Activity Patterns in Parietal Cortex. Cell. 2017; 170(5):986-999.e16.

29. Katlowitz KA, Picardo MA, Long MA. Stable Sequential Activity Underlying the Maintenance of a Precisely Executed Skilled Behavior. Neuron. 2018; 98(6):1133-1140.e3.

30. Charles AS, Song A, Gauthier JL, et al. Neural Anatomy and Optical Microscopy (NAOMi) Simulation for evaluating calcium imaging methods. bioRxiv. 2019; :726174.

SUPPLEMENTARY INFORMATION

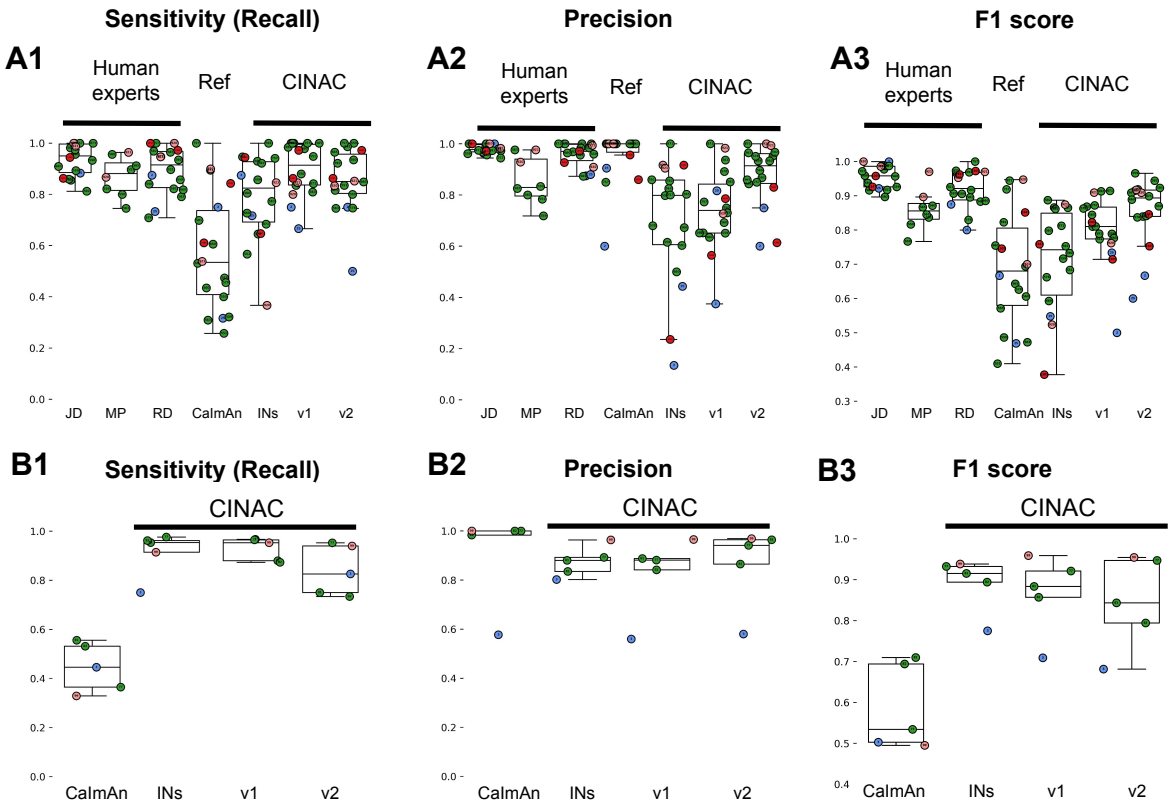

Figure S1: *Recall (sensitivity), precision and F1 score from all classifier's predictions.*

Top panel: *Each graph represents the performance of three human expert (JD, MP, RD), our gold standard (Ref, CaImAn), as well as all classifiers (IN: Interneuron specific classifier; V1: First version of general classifier and V2: Second version of the classifier) on 18 putative pyramidal cells.*

Bottom panel: *Each graph represents the performance of our gold standard (CaImAn), as well as all classifiers (IN: Interneuron specific classifier; V1: First version of general classifier and V2: Second version of the classifier) on 5 putative interneurons*

A1 and B1: *Represent the sensitivity (Recall)*

A2 and B2: *Represent the precision*

A3 and B3: *Represent the F1 score*

Each circle represents a cell, the number inside indicates the cell id. Each color represents a recording.

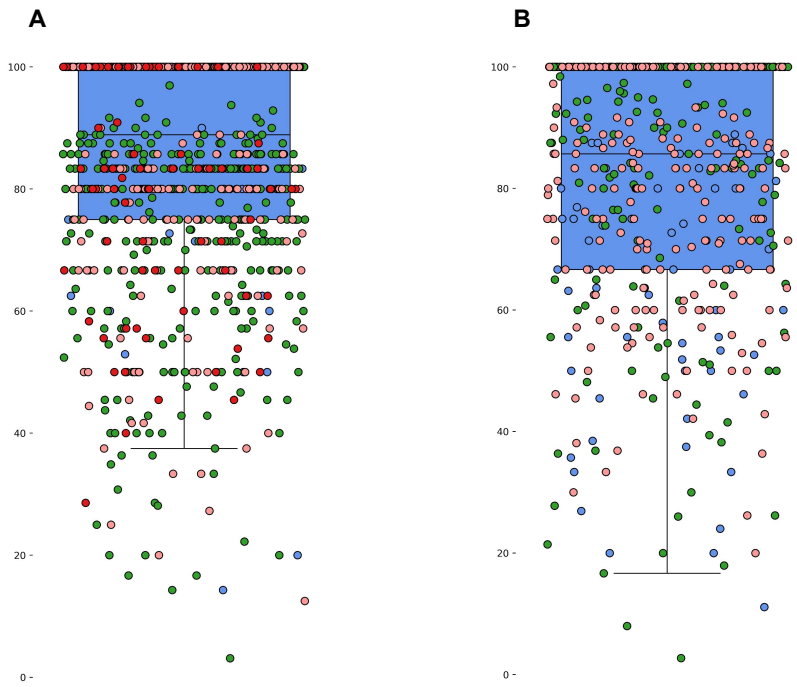

Figure S2: *Distribution of the ratio of frames predicted as active among real transients.*

S2A: *Using the general classifier on 18 putative pyramidal neurons.*

S2B: *Using the general classifier on 5 putative interneurons.*

Each circle represents a transient and each color represents a different recording.

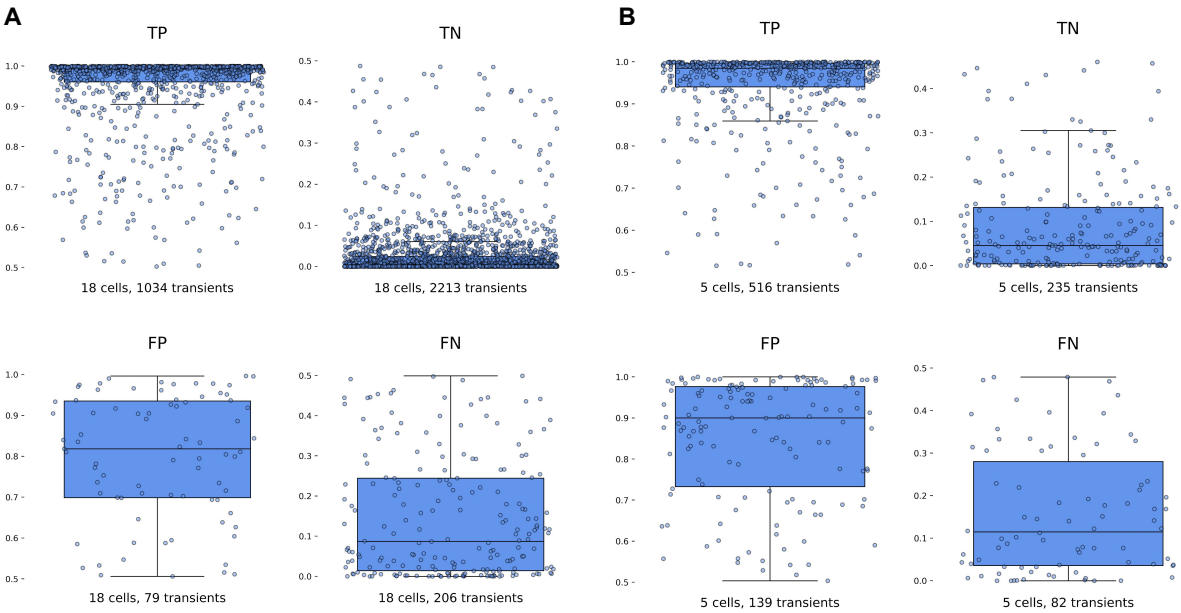

Figure S3: *Distribution of the prediction values from the general classifier for each transient depending on its classification.*

S3A: *18 putative pyramidal cells*

S3B: *5 putative interneurons*

Each circle represents a transient.

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative