Comments to the Author:

Thank you for submitting your manuscript on AI-Track-tive to GChron. The three reviewers all agree that this is a potentially important contribution to the field, which deserves to be published in GChron pending changes to both the software and the paper. In addition to the reviewers' comments, I would like to share some thoughts of my own:

1. In your response to reviewer Chew, you wrote that you will share the source code for AI-Track-tive on GitHub (https://github.com/SimonNachtergaele/AI-Track-tive). I think that this is an excellent idea. At this moment the GitHub page is a mostly empty box, but I look forward to seeing the code appear here when you submit the revised manuscript. Note that you can link the software to the paper via a DOI, as instructed in the Copernicus editorial system.

The GitHub repository contains all necessary files (.py source code) and the Jupyter Notebook. Because GitHub does not allow large files to be distributed through GitHub, we need to distribute the executable application (.exe), the deep neural networks (.weight) and their configuration file (.cfg) through my website (https://users.ugent.be/~smanacht/download_aitracktivev2.php). I will try to link the software to the paper using a DOI in the Copernicus system.

2. Because AI-Track-tive is written in Python, it should be possible to run the software not only on Windows, but on OS-X and GNU/Linux as well. Do you have any plans to make this happen yourself?

Unfortunately, we did not find a way to create executable files for Mac-OS and Linux users. However, using the downloadable Python source code it is possible to successfully run the software in a Python IDE such as Spyder, Pycharm or Thonny (Linux). The software has been successfully tested by one Mac-OS user. Specific installation instructions are specified in the readme file in GitHub. The software has also been successfully tested by the authors on a Linux-based Raspberry Pi 4B. Again, specific install instructions are specified in the readme file available in the GitHub repository.

3. I couldn't really figure out how the software worked until I saw your YouTube video. Note that GChron allows videos to be included with the paper as a supplementary data item. After watching the video, I found the user interface reasonably intuitive. However I did find it a bit annoying that polygons or rectangles are not visible until after the user clicks the ESC button (note that this behaviour also confused reviewer Chew). On several occasions, I accidentally clicked ESC twice, causing the program to terminate. I would suggest that you use a different keyboard shortcut (or key combination) to enter a selection and to terminate the program. I also want to point out that, in its current form, AI-Track-tive cannot be used on a laptop without an external mouse, because laptop touchpads don't have the middle button that is required to terminate the track counting process.

We have made a new video (https://youtu.be/CRr7B4TweHU) that will be linked to the paper as a supplementary data item. Concerning the keyboard shortcuts we have made the changes you asked us to do. Firstly, we enabled the live visualisation of the segments of the polygon. Secondly, we replaced all functionalities of the Escape button by the Space button. The middle mouse button functionality is replaced by the combination Ctrl + mouse move or Ctrl + left mouse click. It is possible to use AI-Track-tive on a laptop using the touchpad.

3. Reviewer Gleadow remarks that the default neural network is based on a small set of only 15 apatite images. In your response, you wrote that:

"Feeding a larger image dataset into the Deep Neural Network would (1) increase DNN training times, (2) require better (and more expensive) GPU's and, last but not least, (3) could improve track recognition success rate." Am I correct that (1) and (2) only affect the training computer? Having run AI-Track-tive on an underpowered Windows laptop, it appears to me that the computational demands on the client's computer are modest. If it is only the training computer that needs to have a powerful GPU, then there is no real reason to use a small set of training data.

You are definitely correct to assume that (1) and (2) only affect the training computer. In order to increase the precision and recall of the training dataset, we increased the training dataset from 15 to 50 images. Our current training dataset consists now of 4734 instead of 624 tracks in apatite and 6212 instead of 1520 tracks for muscovite mica.

Also, I found that the program runs smoothly on a (€60) Raspberry Pi 4B with 4GB RAM. Hence, client's computational demands are really low. The software can be run on any operating system (Windows, Linux and Mac-OS). There is also a logging file that stores the typical output that has been printed in the console.

4. Installing the software generates a folder that is 1.3 Gb in size. This folder contains another folder called INPUT, which is 470Mb in size. This is mostly caused by two files (yolov3_training_*.weights), which I think contain the neural networks for the apatites and micas. Does the size of these files double when you double the number of training images? I don't think so because otherwise the two .weights files have the same size. It may be a good idea to provide the software and the training network as two separate downloads. That way the program will always be the same size (800Mb), and you can slot in different neural networks for different microscopes and different minerals.

You are right in assuming that the .weights files represent the neural networks. The size of the neural networks does not change when the training dataset is changed. I have split up the neural networks in a separate folder the .exe file that I distributed on https://users.ugent.be/~smanacht/download_aitracktivev2.php. Indeed, it makes sense to give the neural networks a useful name so that we can train neural networks for every mineral and microscope set-up.

5. Following up on reviewer Chew's suggestion, I agree that it would be very useful if AI-Track-tive could be used to train a new neural network. If this requires a strong GPU, then that is fine as long as this is made clear to the user.

For the moment we believe that it is a better idea to train a deep neural network using Google's Colab environment, because it is free and is much cheaper than buying a personal computer with the same expensive GPU's. Users who would want to train new neural networks are of course free to do so.

We have provided a full step-by-step protocol in the deep neural network training process in the text and described all necessary thresholds. The neural network training process includes executing lines in an online Jupyter notebook file.

Also, we have programmed one new feature that gives the locations of the identified fission tracks in an image in the "yolov3 .txt format". This file can be loaded in the LabelImg software that is used for the annotation of tracks in training images. It is now possible to annotate the tracks using the deep neural networks or without using them. If one discovers mistakes or ameliorations in the track annotation process, the given dataset from SN can be changed in the LabelImg.

6. For the sake of reproducibility and traceability, it would be very useful if you gave the user access to the training images. You can either include the images with the paper as a supplementary data item, or you can store them on your GitHub page and link them to the paper via a DOI (just like the software itself as mentioned in my first comment).

I have uploaded the training dataset for apatite and mica on GitHub each consisting of 50 .jpg images and 50 .txt files. I will link them through a DOI number.

7. I assume that the 15 apatite images in the training data were also Durango? In that case, the 70% success rate for your test data (which also use Durango apatite) is likely to be overly optimistic. Have you tested the algorithm on different samples and/or age standards?

I have added more information on the training and testing dataset in Figure 2 and Figure 4.

8. I think that the topic of this paper is important and interesting enough for a full sized research paper, rather than a short Technical Note. Expanding the text would allow you to add further details about the AI algorithm, as requested by reviewer Gleadow. Expanding the paper would also allow you to discuss the workflow in more detail, and to provide additional statistics on the training and test data.

We appreciate the positive feedback and interest in our work. Further details about the AI algorithm, a full description of the AI training process, a copy of the training dataset and other details are now specified in the revised version of our paper. Additional statistics (precision and recall) are also added and illustrated Figure 4.

I would be happy to give you up to 8 weeks to update the software and prepare a suitably updated manuscript.

Non-public comments to the author:

Contacteer me gerust direct per email als je vragen hebt.