Principle of TEM alignment using convolutional neural networks: Case study on condenser aperture alignment

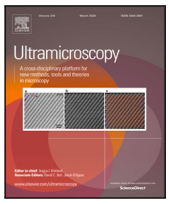
Loïc Grossetête, Cécile Marcelot, Christophe Gatel, Sylvain Pauchet, Martin Hytch

Ultramicroscopy 267 (2024) 114047

Contents lists available at ScienceDirect

Ultramicroscopy

journal homepage: www.elsevier.com/locate/ultramic



Principle of TEM alignment using convolutional neural networks: Case study on condenser aperture alignment

Loïc Grossetête [a,b,*], Cécile Marcelot [a], Christophe Gatel [a], Sylvain Pauchet [b], Martin Hytch [a]

[a] CEMES-CNRS, 29 rue Jeanne Marvig, Toulouse, 31055, France
[b] Fédération ENAC ISAE-SUPAERO ONERA, 7 Avenue Edouard Belin, Toulouse, 31055, France

ARTICLE INFO

Keywords:
Artificial intelligence
Convolutional neural networks
Automatic alignment

ABSTRACT

The possibility of automatically aligning the transmission electron microscope (TEM) is explored using an approach based on artificial intelligence (AI). After presenting the general concept, we test the method on the first step of the alignment process which involves centering the condenser aperture. We propose using a convolutional neural network (CNN) that learns to predict the x and y-shifts needed to realign the aperture in one step. The learning data sets were acquired automatically on the microscope by using a simplified digital twin. Different models were tested and analysed to choose the optimal design. We have developed a human-level estimator and intend to use it safely on all apertures. A similar process could be used for most steps of the alignment process with minimal changes, allowing microscopists to reduce the time and training required to perform this task. The method is also compatible with continuous correction of alignment drift during lengthy experiments or to ensure uniformity of illumination conditions during data acquisition.

1. Introduction

A transmission electron microscope (TEM) is a powerful but complex instrument. Before observations can be carried out, the microscope needs to be aligned correctly. This alignment can take dozens of minutes and is highly dependent on the user's expertise and desired level of performance. Sometimes alignment fails and a more experienced operator is required for assistance. Even for experienced users, many different modes of operation are now available, each of which has its particular requirements. In fact, particularly complex elements, such as imaging spectrometers and aberration correctors, have their own alignment procedures. Another issue is that for lengthy experiments, alignments drift. The microscope configuration should therefore be adjusted continually to maintain comparable observation conditions across a whole series of data acquisitions. All these reasons have led to research over the years to introduce automated alignment procedures by computer control; and only through alignment automation can fully automated workflows be realized.

Early work focused on correcting the aberrations associated with the objective lens [1–3]. Further developments only truly became viable with the introduction of high-performance digital cameras (now considered an essential part of automation) in the form of slow-scan CCD cameras [4,5]. In fact, without some kind of automation, aberration correction in TEM would have been impossible due to the complexity of aligning the multiple elements that make up a corrector [6]. Autofocus and astigmatism correction are now integrated into automated image acquisition workflows, notably in cryo-electron microscopy (for a review, see [7]). Despite these advances, the available automation solutions remain incomplete and fragmented. The application is often too specific — to the microscope, the particular microscope component, the specimen, or the type of data to be acquired — to be used more generally. We would also like corrections to be made dynamically, on the fly [8], to compensate for optical and specimen drift in real-time through feedback control of the microscope [9]. The processes therefore need to be fast, faster than the drift in the alignments, and computationally light to avoid interference with observations or data acquisition underway. Traditional algorithms have to be custom-made for each alignment step and are hard to adapt from microscope to microscope. Indeed, they are not used outside of some very specific applications. It is therefore interesting to investigate whether an approach based on artificial intelligence (AI) could better address the issues of generality and speed.

New algorithmic solutions to similar issues are emerging from the machine learning community, for example, for image processing. Convolutional neural networks [10–14], in particular, allow image extraction and could be used to predict the modification to perform on the microscope configuration to correct the alignment. Convolutional

* Corresponding author at: CEMES-CNRS, 29 rue Jeanne Marvig, Toulouse, 31055, France.
E-mail address: loic.grossetete@cemes.fr (L. Grossetête).

https://doi.org/10.1016/j.ultramic.2024.114047
Received 14 May 2024; Received in revised form 7 August 2024; Accepted 11 September 2024
Available online 9 October 2024

Neural Networks (CNNs) are a family of deep learning models inspired by the human visual system that is revolutionizing image analysis. CNNs work by hierarchically extracting features. They are composed of several layers, including convolutional layers that identify different characteristics within an image, pooling layers that reduce spatial dimensions of the data, and fully linked layers that extract information.

Deep learning, and more specifically CNNs, are already being used in electron microscopy for various tasks [15,16] such as noise removal [17], segmentation [18] and compressed sensing [17]. By scanning the input image with filters, convolutional layers extract information such as edges, textures, and patterns. Using these traits to generate higher-level representations, the network can then identify complex structures in electron microscope images. Concerning alignment, deep learning has been applied to autofocus in scanning electron microscopy (SEM) [19], but the most significant development is for the configuration of specific components such as an orbital angular electron sorter [20]. There are also reports of use for aberration correction [21, 22].

The main difficulty in applying machine learning methods, and indeed traditional methods, to microscope alignment is that a great number of parameters are involved (as can be seen Fig. 1), including high tension, gun, apertures, lenses, stigmators, deflectors, goniometer(s), detectors and additional elements like biprisms, phase plates and spectrometers. Furthermore, many parameters can take a large range of values leading to a stupendous number of possibilities. A random configuration would most likely produce a beam that does not reach the screen. Another difficulty is that a single image might not contain all the information necessary to correct the configuration, for example, the beam tilt with respect to the detector. Unleashing an AI model blindly on the microscope to learn by itself is, maybe, too ambitious for the moment. We therefore first looked at the way that a microscope is currently aligned by humans.

The alignment of the microscope is decomposed into several steps, in general starting at the gun and working down (see e.g. [23] and Fig. 1). Alignments are achieved by fluctuating the currents in one element (“wobble”) or sweeping a range of values, and observing the effect on the image. The microscopist is left to decide in which direction to adjust the currents to achieve the desired result, for example, the beam pulsing on itself or minimizing image shift. The advantage of this procedure is that very different kinds of alignment can be achieved similarly, from gun tilt and condenser aperture alignment to correcting astigmatism and beam tilt: the alignment procedure is more general and easier to learn. Since AI is supposed to ape humans, then possibly models should learn in the same way.

In this paper, we will focus on the centring of the condenser aperture as a manageable but representative task. We will train deep convolutional models to predict the displacement of the aperture. However, the experimental dataset required for training must, however, be acquired within a reasonable time, automatically, and without damaging the microscope or detector. At all stages, we will attempt to keep the approach as general as possible to imagine aligning the other elements with minimal changes. If the machine can be demonstrated to equal the human operator, even an expert, in speed or accuracy, the methodology is worth pursuing.

Our long-term goal is to use artificial intelligence to automate the TEM alignment process and possibly extend this work to assist microscopists during their experiments in maintaining ideal conditions for sample study.

2. Experimental details

Experiments were carried out with the I2TEM microscope, an HF3300-C (Hitachi HT) [24] (Fig. 1), specifically designed for laboratory needs in in-situ and interferometry. The microscope is equipped with a cold-field emission gun (C-FEG), multiple biprisms (one in the illumination system and two post-specimen), two independent specimen stages (an additional Lorentz stage above the objective lens), a wide field of view imaging aberration corrector (B-COR, CEOS), a direct electron detector (K3, Gatan) and imaging spectrometer (Quantum ER-965, Gatan). A standard carbon grid was inserted in the conventional specimen stage and the microscope was operated at 300 kV.

Communication with the microscope is through TCP (Transmission Control Protocol). Using the details kindly provided by Hitachi High-Technology, we were able to develop our own application programming interface (API) to communicate directly with the microscope server [25]. Corresponding libraries of commands were created to allow the microscope to be controlled through DigitalMicrograph (Gatan) or Python scripting.

The latter option was used to program automatic acquisition of experimental training datasets. We chose to run these scripts outside of DigitalMicrograph, acquiring images through screen capture 1 (on the right) in a region defined by the user. There are several reasons for this. Firstly, the implementation should be directly transferable to microscope setups using any other image acquisition software and runs independently to the normal control of the microscope. Secondly, the microscopist can easily change the acquisition from the image to, for example, a live fast-Fourier transform (FFT) on display. Finally, the information received by the model is identical to that of the human operator. There is about a half-second delay between the capture and the transfer of the acquired image, which is manageable.

Simulated datasets were also produced using a previously developed digital twin of the I2TEM microscope [25,26]. Ray paths are calculated using the paraxial approximation, which is sufficient for our current purposes (Fig. 1). Although simulations are not an accurate representation of experimental images, they are useful to rapidly test the viability of different CNN architectures.

On top of that, we designed simulations to predict the path of the electrons inside a TEM similar to the I2TEM (Fig. 1 on the left) [25]. This greatly helps to produce large quantities of data, to test new solutions fast, and continue working when the microscope is not available. We also hope to use it as a base for transfer learning [27,28] in the future. Transfer learning is the ability to use a model pretrained on the simulation data to make the real training faster with fewer microscope images.

All Python scripts were developed for Python 3.9, the machine learning models are implemented using TensorFlow.

3. Aperture alignment

Before tackling the issue at hand, we developed a proof of concept with the aim of trying to find the configuration used to generate images through deep learning. Although we will not detail the results here, they allowed us to gain key insights regarding the project’s feasibility and to the performances we could expect from CNN’s, especially since predicting the whole configuration shift would probably be too difficult. We, therefore, chose to develop a solution mimicking human behaviour of aligning the column component by component to adjust only a couple of parameters at a time. In this paper, we are showcasing the method we developed on aperture alignment but the process would be very similar for the rest of the column.

The first alignment step is the centring of the condenser aperture. A human can perform this task by first “wobbling” the condenser lens current and determining whether or not the spot on the images remains centred. The aperture is then moved by an x/y shift on the control panel, wobbling after each change, until the beam pulses on itself. The choice of shift direction and final acceptability are subjective and guided by experience.

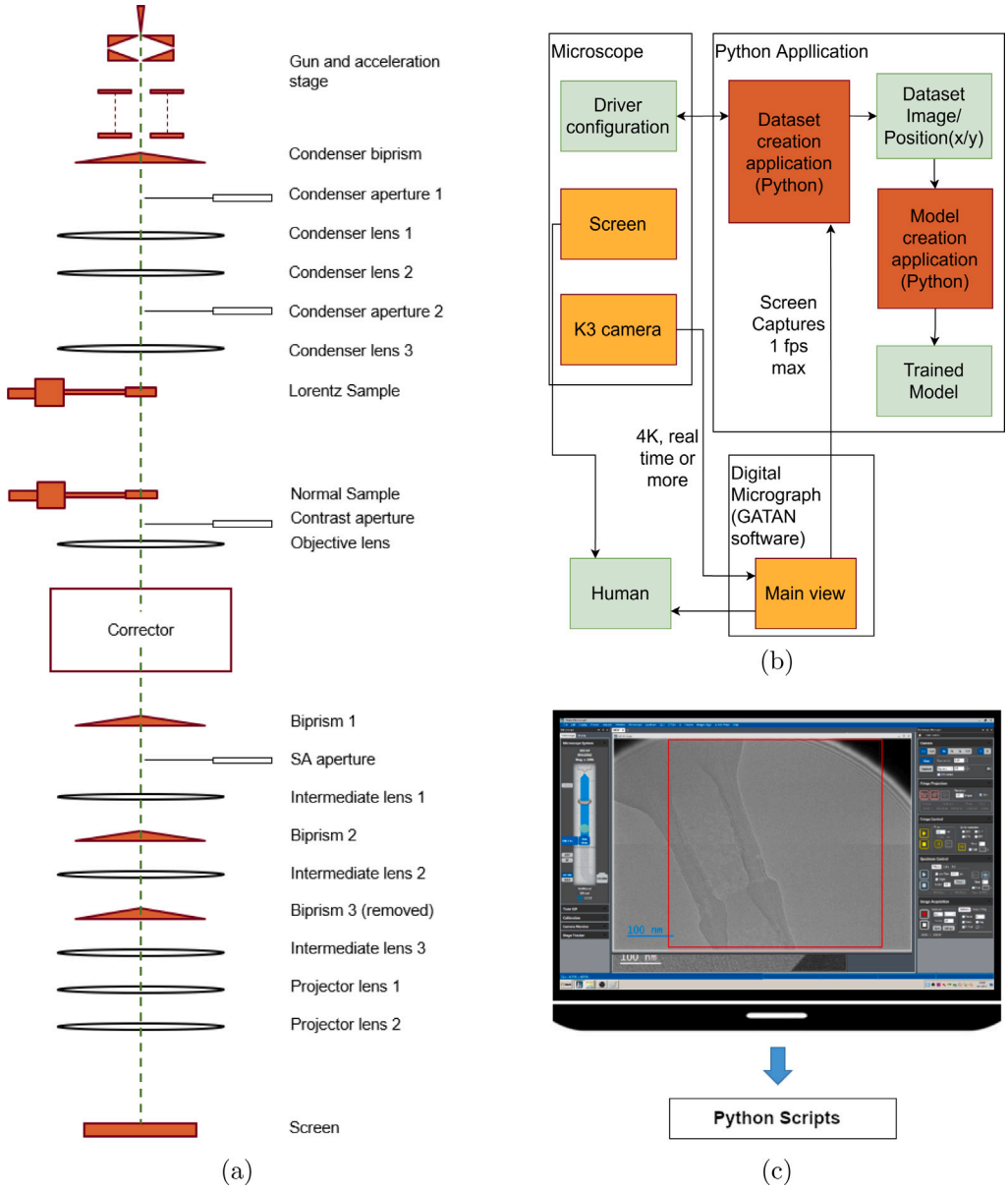


Fig. 1. (a) Simplified schema of the I2TEM, (b) Data Acquisition process with the I2TEM, (c) I2TEM image retrieving.

3.1. Traditional algorithms

Before describing the AI implementation, it is interesting to consider the possibility of using more traditional computing methods, notably for the analysis of a set of images with different values of condenser lens currents. The diameter and position of the beam would first need to be determined. Secondly, the sizes and positions would have to be ordered and analysed to determine the central position of the wobble, where the beam is the smallest. Finally, this position would have to be converted into an x/y displacement of the aperture. Whilst this procedure seems simple, there are a number of technical difficulties associated with each step once we enter into the details. For example, the beam might not always be entirely on the detector, resulting in part of the beam being cut off. Neither is the beam (or aperture) always perfectly circular or indeed clearly delineated if the aperture is out of focus. The determination of beam size and position would have to deal robustly with such cases. Even the determination of the centre of the wobble could become inaccurate because of the need for extrapolation when the recorded images miss the in-focus position. The most delicate part, however, involves the conversion of the in-focus position recorded on the detector into an x and y shift of the aperture. Only extensive calibration could address this issue. The implementation would therefore be highly specific to the task and difficult to generalize. And, finally, to our knowledge, traditional methods have not been implemented for the alignment of the condenser aperture, possibly for these reasons.

On the other hand, machine-learning-based methods do not face the same issues. The prediction of the shift is computed using the same kind of information available to humans. If a model performs well, it means that the difficulties have been overcome without the need to adapt the process for each problem. If after deployment another difficulty is discovered, the model can be retrained using examples showcasing this specific issue and the model should generalize. Training can be accomplished during microscope downtime and performance can be improved continually without changing the basic algorithms, unlike traditional solutions. Furthermore, transfer learning should significantly reduce the data collection time. Most important though is the possibility of generalizing the solution to the other alignment tasks, such as focus, and pivot-point alignment. Some kind of adaptation will certainly be

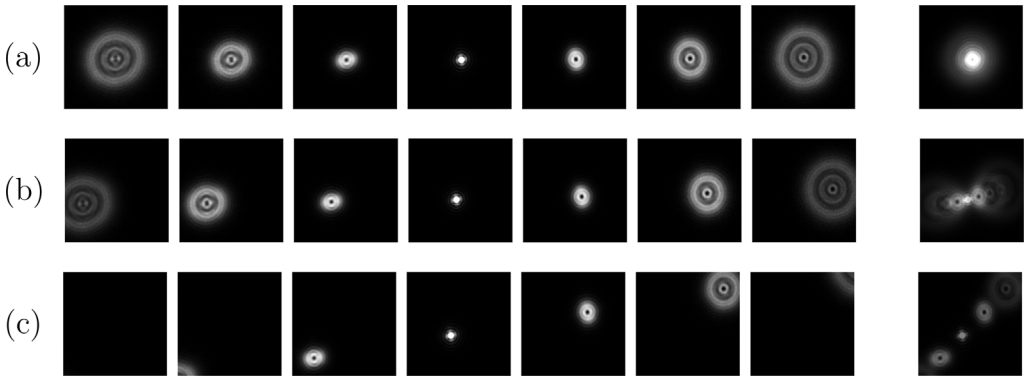


Fig. 2. Example of images in dataset 13. Series (a) is aligned, (b) is slightly misaligned and (c) is very misaligned. For each series, the seven first images correspond to different wobble values and the eighth is the stacked image.

necessary but not on the scale of more traditional solutions. But if AI-based methods fail on the very first step of the alignment process, there is of course no point in hoping for a general solution.

3.2. Datasets

Experimental datasets were acquired corresponding to the task at hand. We started with an aligned aperture and then changed the position of the aperture by a known Δx and Δy resulting in a known misalignment of the aperture. We then acquired 7 images at varying values for the $C1$ (condenser) lens current. The choice in the number of images to be taken is arbitrary but is a compromise between the time cost and the expected gain in accuracy. The maximal displacement values were selected to keep the beam visible in most images. This allowed us to make datasets linking an ideal Δx and Δy to a 7-layer image. Experimental examples are shown in Fig. 2.

The first row (series A) is the decomposition of the seven different images corresponding to the aligned state, with the condenser lens varied linearly about the original value, C1, starting at 0.9 C1 to 1.1 C1. Each image was taken with a two-second delay to allow for the configuration to be stable again after changing the lens current. The last image displayed on the right is a composite image representing the sum of the seven individual images, where we see more clearly the concentric nature of the images. Note that we chose to use the smallest condenser aperture in the initial tests to avoid any chance of damaging the detector. This precaution was, in fact, later found to be unnecessary and explains the strong Fresnel fringes but we do not know if it impacts the model predictions.

Introducing a displacement (such as Series B and C in the second and third row of Fig. 2) means that the circles are no longer concentric.

Since the displacement of the aperture is mechanical, and hence inherently slower than a current change, rather than choosing random positions in a random order, the displacements were chosen in a spiral (see Fig. 3), scaled with respect to the maximum displacement. The algorithm was designed to guarantee a good coverage of the parameter space as well as proximity between successive points:

```
def generate_datapoints(
        perfect_x, perfect_y, delta_max, number_of_points
    ):
    datapoints = array(number_points, 2)
    for i in range number_of_points:
        delta_max = radius * i / number_points
        phase = i/number_points*2*np.pi * number_revolution
        datapoints[i, 0] = perfect_x + delta_max*np.cos(phase)
        datapoints[i, 1] = perfect_y + delta_max*np.sin(phase)
    return datapoints
```

The different experimental dataset are described in Table 1, most had 127 positions of the aperture, including the aligned position, each associated with a set of 7 wobble images. The maximum Δx and Δy used for the aperture position is given for the outermost point on the spiral. The maximum possible displacement for this aperture was ±1000 (arbitrary units) and we chose between 150 and 250 because starting from an aligned position, a greater displacement would have caused the beam spot to be outside of the camera's view when wobbling. The wobble value represents the percentage variation of the condenser lens current around its initial position. (If the C1 value is 1.2 and wobble is 3% then the values for C1 will be [1.2 – 0.036, 1.2 – 0.024, 1.2 – 0.012, 1.2, 1.2 + 0.012, 1.2 + 0.024, 1.2 + 0.036]).

Table 1
Descriptions of the datasets collected on the I2TEM.

Dataset	Number of samples	Delta x/y	Wobble
N° 14	84	240	3%
N° 13	127	240	3%
N° 10	127	240	3%
N° 6	127	200	2%

It should be noted that dataset N° 6 is different than the others on many aspects to allow us to test the model robustness against such changes. In particular, it is wobbled by only 2% of the condenser lens value and contrary to the others, the wobble is not centred around maximal convergence implying a very different set of images. Also, it was taken two days later introducing a potential drift in the ideal configuration. Other datasets were acquired but were deemed of insufficient quality for different reasons (see Supplementary Materials for more information).

3.3. CNN architecture

All the CNN we tested had similar architecture to the proof of concept: a first group of convolution, dropout, and pooling layers followed by a group of dense layers to process the information extracted by the convolutions (Fig. 4). The model shown is, in fact, the best-performing CNN, the selection of which is now described.

3.4. Training and evaluation

Models were trained on 80% of the data and tested on the remaining 20% unless otherwise specified. The RMSE was defined as a percentage with respect to the maximum range of 1000 for the aperture. A human operator is happy to align the aperture to within 20 units of displacement, which represents an RMSE of 2%. As for the proof of concept, we know that the model is training well if the RMSE between the predicted parameters and the ground-truth parameters decreases steadily with epochs.

To find the best model for our needs, we compared the performance of different candidates trained and tested on dataset N° 14. The resulting RMSE after 200 epochs of training are shown in Fig. 5 for a selection of the models. The first column shows the fully connected network,

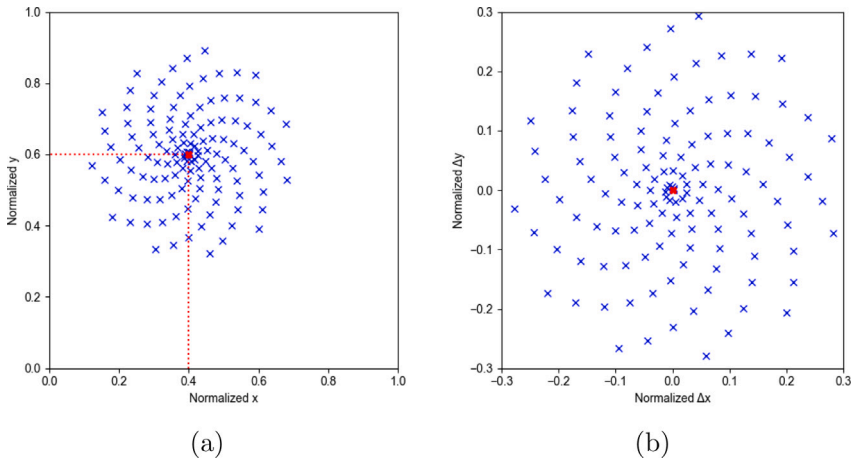


Fig. 3. (a) Example of aperture coordinates (normalized) used for training where the aligned configuration is at (0.4, 0.6); (b) the normalized $\Delta x/y$ to predict.

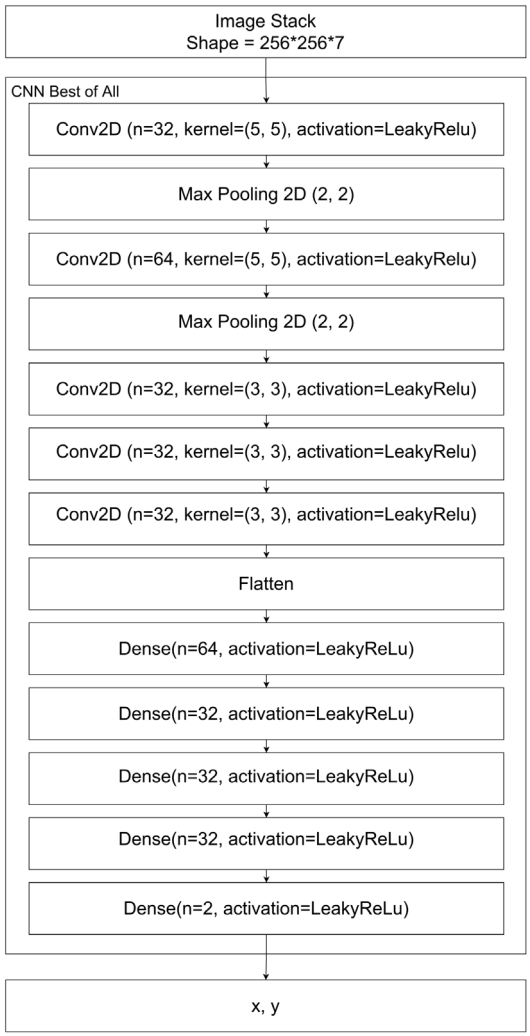


Fig. 4. Architecture of the model used in this paper.

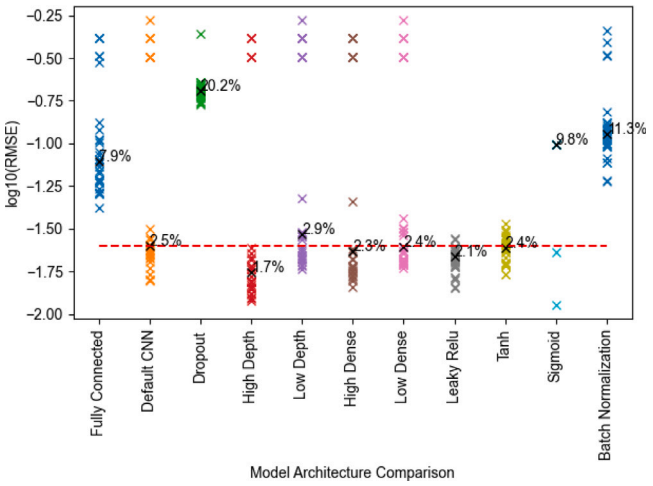


Fig. 5. Performances of different model architectures tested on dataset 14. Each cross represents one training over 200 epochs of the given model and the median performance is high. The dotted line (red) indicates the median performance (2.5% RMSE) of the initial model (CNN). The logarithm of the loss is used here for clarity.

which is not in fact a CNN as per se, but is included for comparison purposes. The next column is the initial CNN we built, which will be the initial model for further modifications. The red dotted line corresponds to its average performance (2.5% RMSE); models above and below this line thus either under- or out-perform the initial CNN. We changed just one feature of this model at a time and tested the performance (results shown in the succeeding columns). For example, we varied the use of dropout layers, number of convolution/max-pooling layers, batch normalization [29], some optimizers (not shown here, see Appendix A for further details), and different activation functions. Full details of the different models can be found in Appendix A. We then combined all the characteristics that enhanced the model architecture into one final model.

We can see that although the performance of the initial model is fairly consistent, there are still 10% to 15% of the training sessions which fail. This is not the case when other activation functions are used such as 'LeakyReLU' or 'tanh' which is called the "dying ReLu" problem [30]. From those we chose to use 'LeakyReLU' for all layers since it performed better. Better results were obtained using a CNN with higher rather than lower depth. Similarly, the model with more dense layers performs consistently better than models with fewer dense layers, although the median performance is only marginally better. Changing the activation function to LeakyReLU or Tanh produces good results with no outliers, whilst Sigmoid did poorly on average despite a few training reaching low losses.

Batch normalization (usually used to stabilize training) only seemed to deteriorate the model performances.

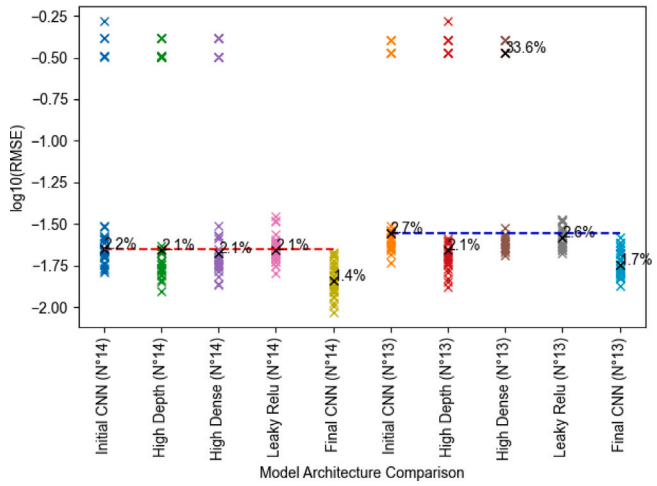


Fig. 6. Performances of different model architectures tested on dataset 14 and 13. The dotted line indicates the average performance of the initial CNN. In red for N° 14: 2.2% RMSE and in blue for N° 13: 2.7% RMSE.

Table 2
Performances of the model on all datasets.

Dataset	N° samples	$\Delta x/y$	Wobble	Median RMSE	Real Δ
N° 14	84	240	3%	1.37%	14
N° 13	127	240	3%	1.94%	19
N° 10	127	240	3%	2.64%	26
N° 6	127	200	2%	1.53%	15
N° 10/13	254	240	3%	1.93%	19
N° 10/13/14	338	240	3%	2.75%	28
N° 10/13/14/6	465	240	2%–3%	2.72%	27

We therefore created a model combining all improvements made on the first model by applying all the changes that improved its performances (more CNN and dense layers as well as LeakyReLu) and did 30 runs to see if it would show an improvement over the initial CNN and the results are shown in Fig. 6.

We can see that the best results are obtained with the final model. Very similar results were obtained when these CNN were tested on dataset N° 13.

3.5. Model performances on each dataset

Having chosen the best model, we trained and tested it on the other datasets. To enlarge their number we also created merged datasets from the initial set (Table 1). The model was trained on each dataset 20 times during 200 epochs and the median performance was determined. The results are summarized in Table 2 and Fig. 7.

The model reaches a performance comparable to that of an experienced operator (2% RMSE) on all of the datasets, better performing on most and underperforming on one of the original datasets (n° 10) and two of the merged datasets (n° 10/13/14 and n° 10/13/14/6). The model is also reliable, with only one case of a bad outlier (n° 6) in thirty training per dataset which is probably explained by bad luck on the data selected for the testing set.

The variation in performances between the different datasets is most likely due to the inherent quality of the dataset which depends on a lot of variables such as the drift of the ideal configuration with time, the contrast, the initial condenser lens value etc... The small sample size can also cause variations, especially for the same dataset which is highlighted by the fact that the models trained and tested on merged datasets seem to perform more consistently. It means that luck plays a bigger part in the recorded performances on single runs.

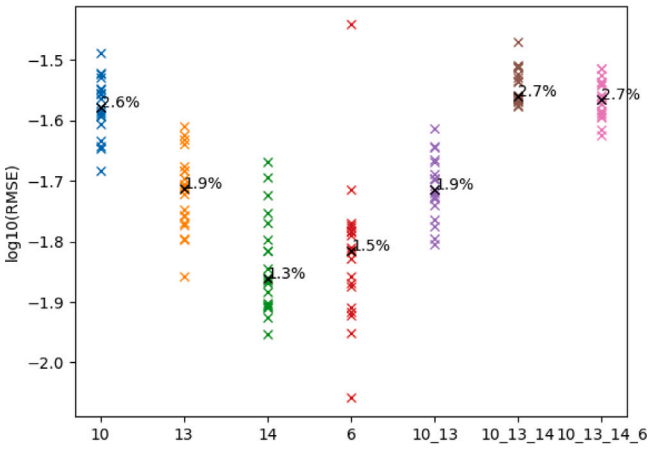


Fig. 7. Performances of the model on all datasets used.

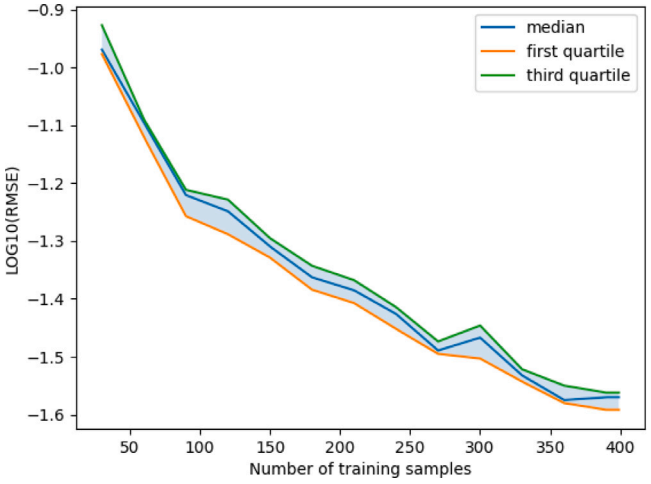


Fig. 8. Performances of the model depending on the number of training samples on the dataset n° 10/13/14.

3.6. Impact of dataset size

To assess the importance of the size of the dataset, we trained the network on one of the largest datasets (n° 10/13/14) but with a fewer and fewer number of configurations. The results are shown in Fig. 8.

We can see that the network probably still has room for improvement if more data could be provided.

3.7. Generalization

The previous tests show that the network can predict the alignment needed in a similar scenario to the one it was trained for. However, the final goal is that the trained model performs well on other days and for other configurations of the lens.

To investigate how well models might generalize we used the model trained on the dataset N° 10/13 and applied it to the dataset N° 14 which is similar (taken just after the datasets N° 10 and 13 and with the same parameters) as well as the N° 6 which was collected two days later with a slightly different configuration and different parameters. The results are shown in Table 3 and Fig. 11.

We see that the model performs remarkably well on a similar dataset (N° 14), better than the training dataset (N° 10/13), even though the cold-field emission gun was flashed between dataset acquisitions. Good performance is obtained for the dataset taken on a different day, multiple flashes, different alignment and under different conditions (N°

Table 3
Test of generalization: performance of the model trained on dataset N° 10/13 applied to datasets N° 10/13, N° 14 and N° 6.

Dataset	RMSE	Real difference
N° 10/13	2.15%	22
N° 14	0.51%	5
N° 6	2.90%	29

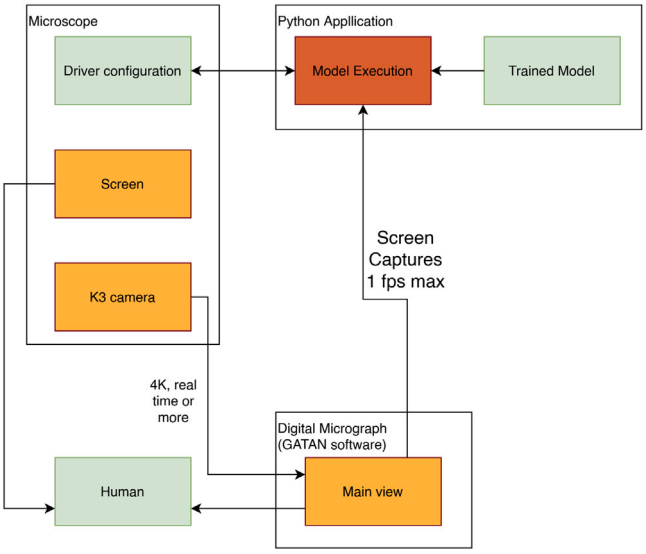


Fig. 9. Schematic representation of AI correction.

6). The amplitude of the wobble was smaller than the training dataset for example despite some outliers. The outliers are a natural occurrence for such models despite causing an issue. If the image is not perfectly stable during acquisition or sometimes due to the imperfect training of the model.

3.7.1. Automatic alignment test

Considering these promising results, we wondered whether a fully automated alignment scheme could function on the microscope. We therefore set up the scheme shown in Fig. 9.

The program automatically acquires a wobbled series of seven images, the model predicts the position of the aperture with respect to an aligned state and the correction is communicated back to the microscope. A further wobbled series of images is automatically acquired to confirm the accuracy of the alignment.

We tested the model trained on dataset N° 14 on a misaligned configuration. The composite image of the wobble before and after automatic alignment is shown in Fig. 10.

A more quantitative idea of the performance can be appreciated in Fig. 11. This shows how the model trained on dataset N° 10/13 performed, comparing the injected misalignment with the results after correction. The AI performs well even when applied on a different initial configuration.

4. Discussion

For the alignment scenario tested here, there is probably no point in improving the model further. Indeed, the initial model performed to a near human level already. The most significant improvement involved the activation functions, making the model more reliable. We also chose to train the model on the minimum of data. The datasets could be easily increased by an order of magnitude by increasing the time spent on the microscope from 30 min to 5 h. Another order of magnitude would be more difficult perhaps, without decreasing the time between acquisitions. If we could accept blurred images (notably during wobble), or not completely stabilized configurations, another order of magnitude could be achievable. This would allow more data for a particular configuration or more configurations to be tested.

The performances shown by the model trained on dataset 10 and 13 on dataset 6 show that the technique is resilient since dataset 6 is very different from the others. It would still be useful to stress-test the model in the future with configurations and therefore images varying considerably from the training data, especially with lower signal-to-noise ratio or with completely misaligned configurations. It would also be interesting to carry out tests using the more commonly used larger aperture size.

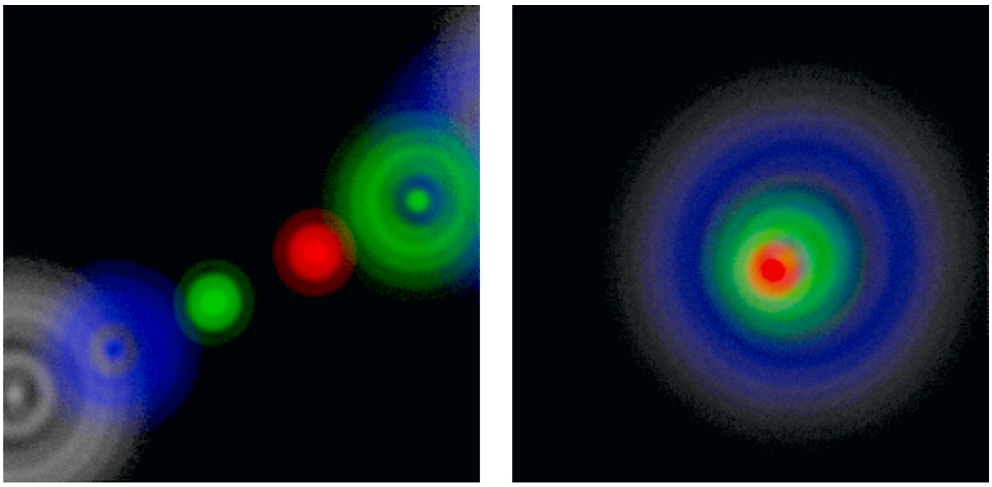
The initial configuration was set by a human operator. The magnification was notably adjusted so that the wobbled images were visible on the screen. A higher level of automation could therefore require a routine that adjusts the initial conditions, or possibly an iterative use of AI. Currently, the model predicts the position of the aperture in one go, so there is plenty of opportunity for exploring a sequence of adjustments. There remains, however, the general but difficult problem that there is no easy evaluation of whether the AI makes a mistake.

The generalization of the current scheme to other elements in the column needs further consideration. One point that should be addressed is the range of certain parameters, some having an intrinsic range of values beyond most uses. For the example studied here the x/y position of the aperture is coded by an integer in the range [−8 388 608, 8 388 607] when for a given hole, the effective range would only be in the thousands. Normalizing on the whole range would therefore lead to a loss in accuracy and we would probably require more data and a dynamic adjustment of the learning rate of the optimizer. Without changing the general approach the ranges of reasonable values for the parameters need to be predefined in some way, ideally automatically.

Still, the results are promising since most elements to align do follow a similar alignment method and most only concern electrical adjustments rather than mechanical ones. There is also the exciting possibility to use real-time training through reinforcement learning (RL) which could allow the model to do trial-and-error to obtain the ideal configuration as well as to explore the configuration space to get more information than what the current model can see. An ensemble of metrics about the output image could be developed, such as magnification, position of the sample, focus etc…, that a reinforcement model [31–33] could use as a goal. This would allow the user to set a goal for the network which would then stabilize the configuration during the experiment to maintain the conditions imposed by the goal.

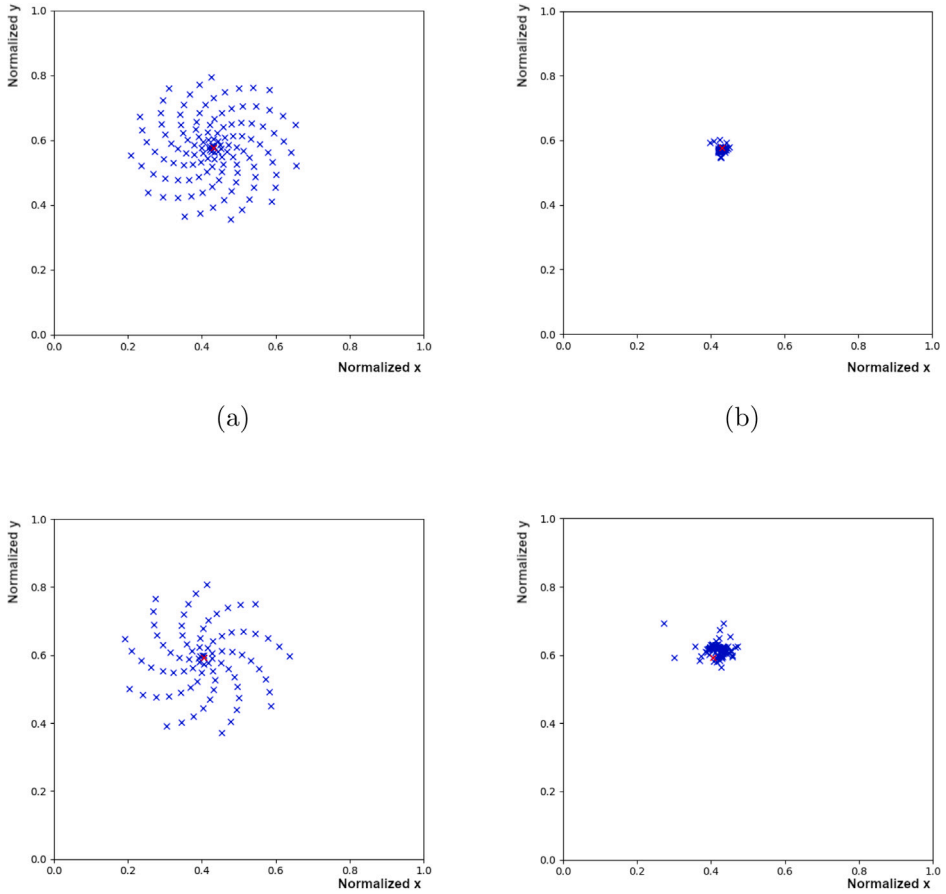
5. Conclusion

We demonstrated the possibility of using machine learning to accomplish part of the microscope alignment, in this case, the condenser aperture. The method was designed to be general and should be extendable to other alignment processes as well as to other microscopes. Although it is still imperfect, the model's performances are similar to a human and should improve using more training data. Further tests are still needed to ensure that the method will continue working as the microscope's optimal parameters shift over time to avoid the need for periodic retraining. However, it still shows a surprising adaptability even when the images used are very different from the images used in the training set. It may then be possible to combine routines to completely align the microscope automatically and stabilize imaging conditions during experiments.



(a) (b)

Fig. 10. Example of alignment provided by a model trained on dataset 14 directly on the microscope after the operator set a misalignment. (a) Before alignment using the model, (b) after alignment. Different colours were used for each image (the model sees them as 7 layers) and the aligned one only has 4 images for readability.



(a) (b) (c) (d)

Fig. 11. Visualization of the correction provided by the model, trained on dataset N° 10–13, on dataset N° 14 (a) and (c) are the normalized x/y positions of the aperture of datasets N° 14 and N° 6 respectively, while (b) and (d) are the aperture normalized x/y positions after correction by the model. The red cross corresponds to the ideal correction.

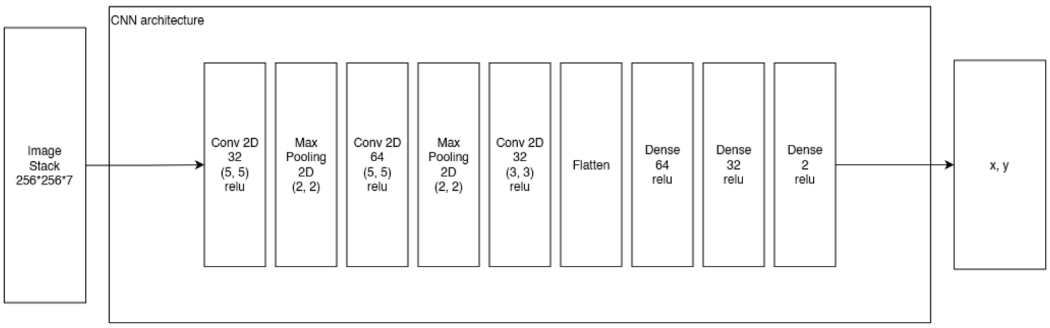


Fig. A.12. Simple CNN model.

CRediT authorship contribution statement

Loïc Grossetête: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Data curation, Conceptualization. **Cécile Marcelot:** Writing – review & editing, Validation, Resources, Investigation. **Christophe Gatel:** Writing – review & editing, Validation, Resources, Investigation. **Sylvain Pauchet:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Martin Hytch:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Martin Hytch reports financial support was provided by EUROPEAN UNION - IMPRESS. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors acknowledge funding from the European Union under grant agreement no. 101094299 (IMPRESS). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

We also extend our gratitude to Hitachi for providing invaluable information enabling communication through TCP with their Microscope. Their assistance has been crucial in advancing our research efforts.

Appendix A. Model architectures

This appendix describes in detail all the models used in the main article.

A.1. Main model used in the paper

The initial model against which we tested all variants is the one described in Fig. A.12. This is a Convolutional Neural Network taking 256×256 pixels images where the channels traditionally associated with colours are replaced with the 7 grayscale images corresponding to the different values of the first condenser lens.

The model is sequential and starts with a combination of convolution layers followed by max pooling layers. The first applies n convolution filter to the image of kernel's shape (x, y) before using the activation function defined (here ReLu) and the second reduces the size of the input data by taking the maximum value for patches of a feature map and, uses it to create a smaller feature map.

Then it follows with a few convolution layers without pooling followed by a flattening layer which transforms the (k, k, d) array by a vector of size $k * k * d$.

Finally, there is a succession of Dense layers which take an input vector of size n and multiply it by a (n, n) matrix before applying the activation function defined (here ReLu).

For this model and unless specified otherwise for the other models, the optimizer used is Adam and the loss is the mean squared error.

The code for the model is the one described in

```
model = Sequential(name=name)
model.add(InputLayer(input_shape=input_shape))
model.add(
    Conv2D(32, (5, 5), activation='relu', padding='same')
)
model.add(MaxPooling2D((2, 2)))
model.add(
    Conv2D(64, (5, 5), activation='relu', padding='same')
)
model.add(MaxPooling2D((2, 2)))
model.add(
    Conv2D(32, (3, 3), activation='relu', padding='same')
)
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(np.prod(output_shape), activation='relu'))
model.add(Reshape(output_shape))
model.compile(
optimizer='adam', loss='mse', metrics=[RootMeanSquaredError()]
)
```

A.2. Other variants

A.2.1. Fully connected

The fully connected variant A.13 directly flattens the image and uses dense layers to predict the output.

A.2.2. Dropout

The dropout variant A.14 adds dropout layers after some convolution/max pooling layers which can help the model with generalization in some cases. The dropout layer hides a fraction of the layer parameters to prevent the model from relying too much on some part of the input and encourages it to fully exploit the data.

A.2.3. Batch normalization

The batch normalization variant A.15 normalizes its input during training through re-centering and re-scaling. The exact reason why it sometimes helps with the training remains still unclear.

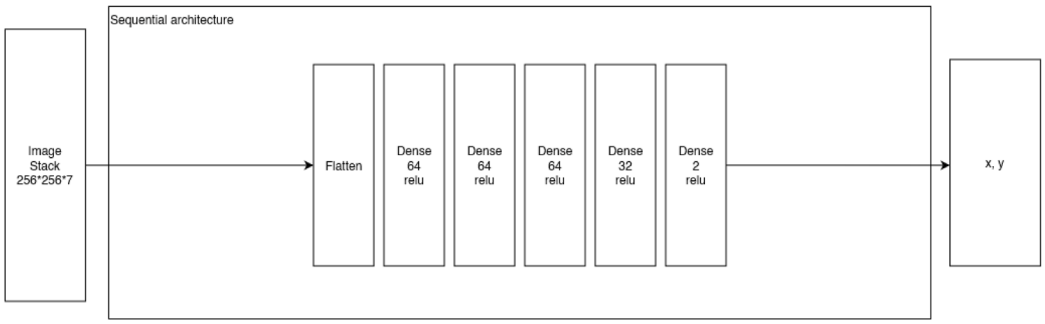


Fig. A.13. Fully connected model.

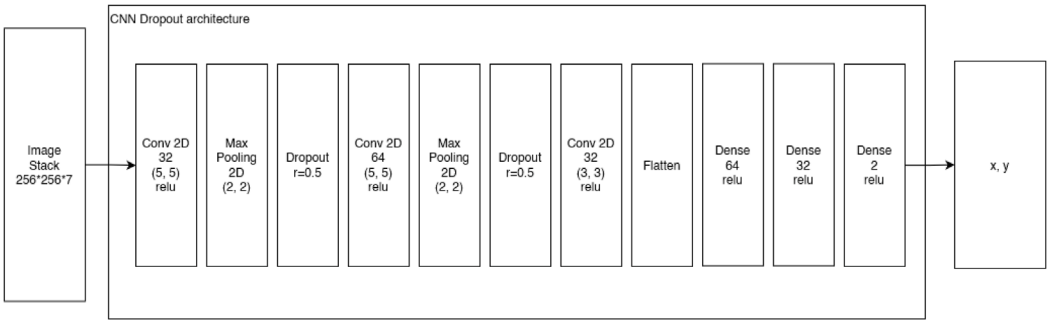


Fig. A.14. Dropout model.

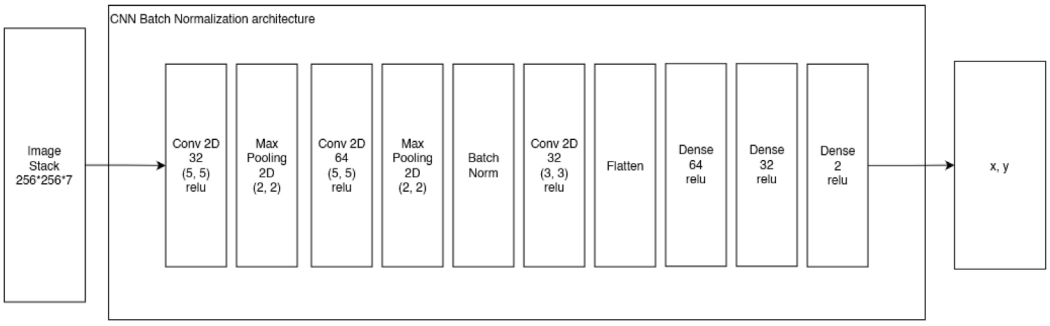


Fig. A.15. Batch normalization model.

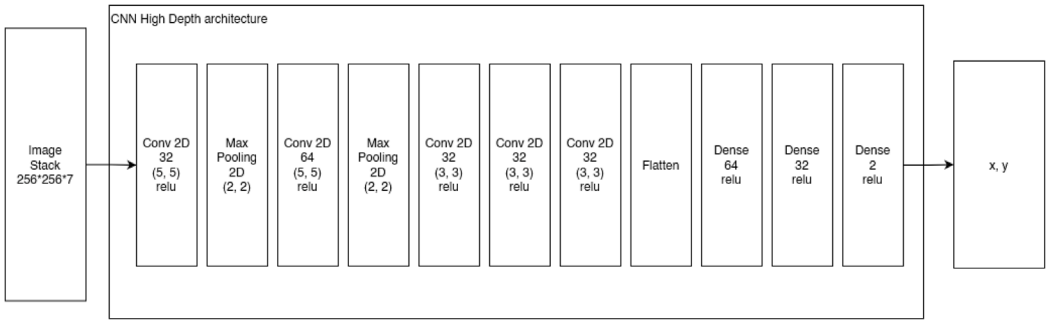


Fig. A.16. High depth convolution model.

A.2.4. Depth in the convolution layer

The exact number of convolution layers is also an important parameter which depends on numerous other parameters such as the size of the dataset, the type of images used, etc... We tried two variants of the initial model, one with one less convolution layer and one with one more as can be seen Figs. A.16 and A.17.

A.2.5. Depth of the dense layers

Another important factor is the number of dense layers used to process the output of the convolution layers. We tried two variants, one with less dense layers and one with more dense layers as can be seen Figs. A.18 and A.19.

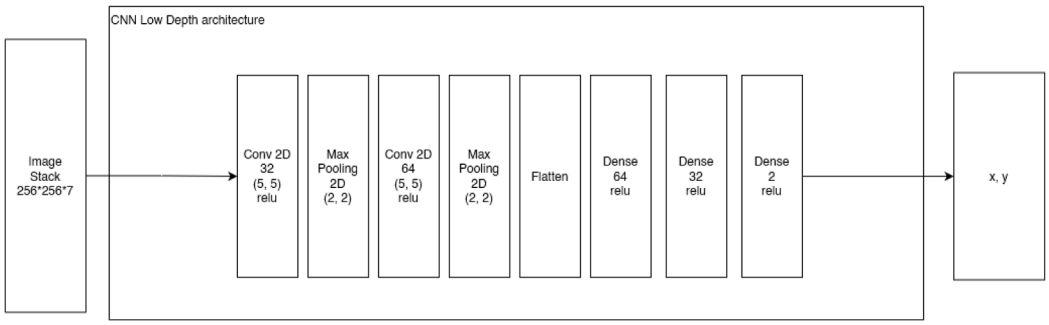


Fig. A.17. Low depth convolution model.

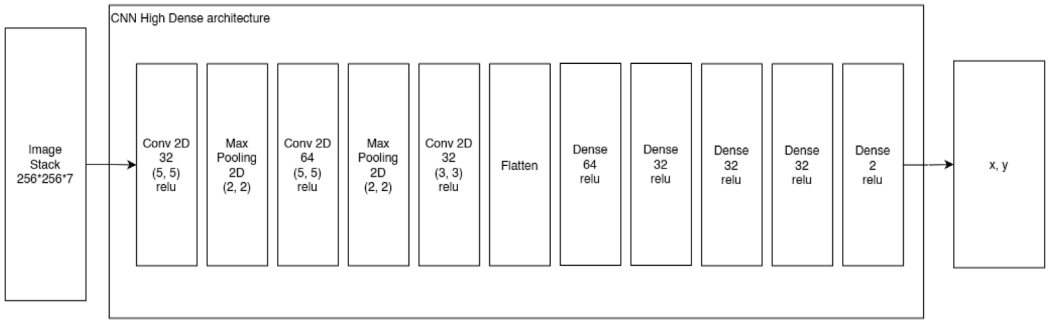


Fig. A.18. High depth dense layers model.

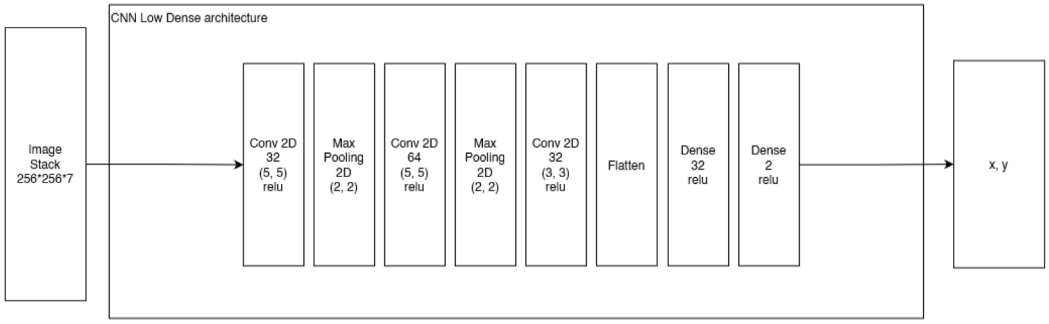


Fig. A.19. Low depth dense layers model.

A.2.6. Activation function

An important factor in the performance of the model is the chosen activation functions. Many layers have a step after they sum the influence of different input values to form an output value where they pass the resulting value through an activation layer which allows control over the range of the output value and helps break linearity. We tested different variants for the different layers involved as can be seen in Figs. A.20, A.21, A.22 to compare with the ReLu activation function used in the default cnn model.

A.2.7. Optimizers

We tried different optimizers for the CNN, optimizers are used to determine how new training samples update the weight inside the model and there are a lot of possibilities. In this article, we tried on the initial CNN:

- Adam(0.01)
- Adam(0.001)
- Adam(0.001)
- Adadelta
- Adamax
- Adagrad
- Sgd
- Nadam
- Rmsprop
- frtl

A.2.8. Final model

Given the results of the first experiment which can be seen in Fig. A.23, we decided to create a model with all the properties which enhanced performances.

The new model presented in Fig. A.24 was then tested against the best models of the previous experiment and the results are presented in Fig. A.25.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ultramic.2024.114047.

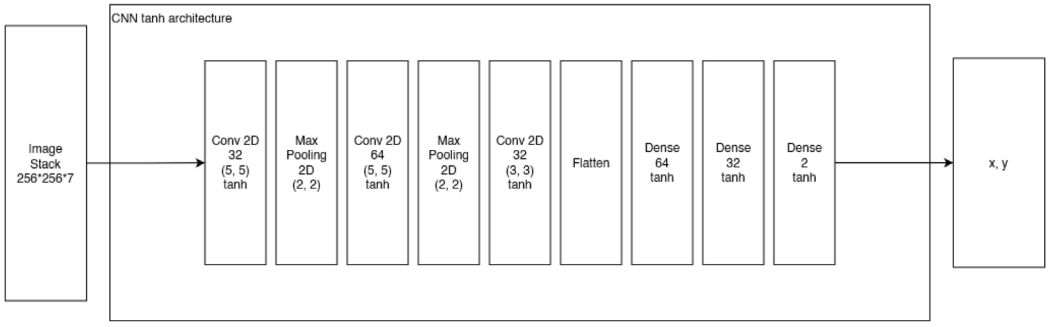


Fig. A.20. Tanh activation model.

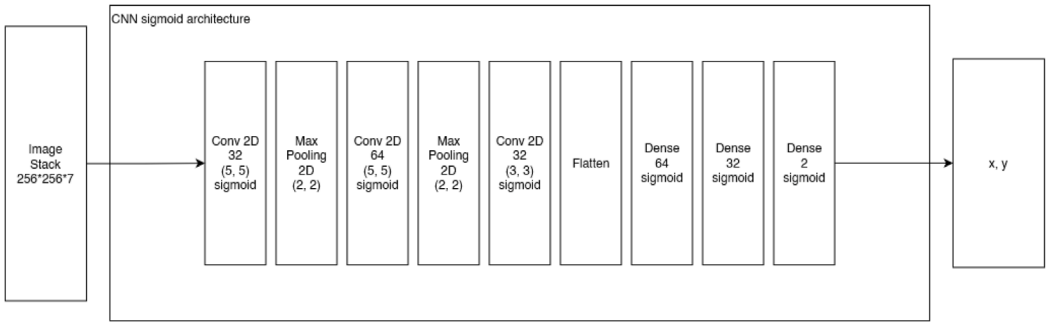


Fig. A.21. Sigmoid activation model.

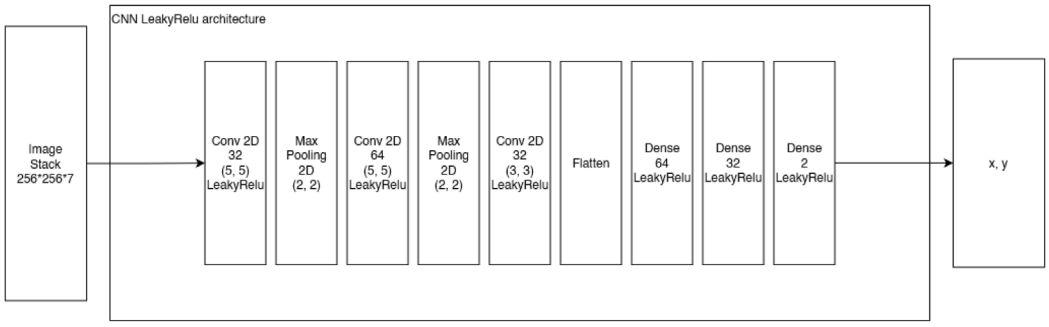


Fig. A.22. LeakyReLu activation model.

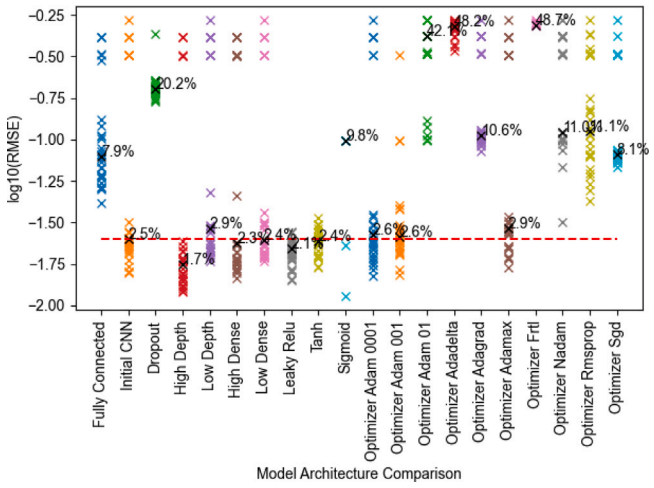


Fig. A.23. Performances of different model architectures on dataset 14.

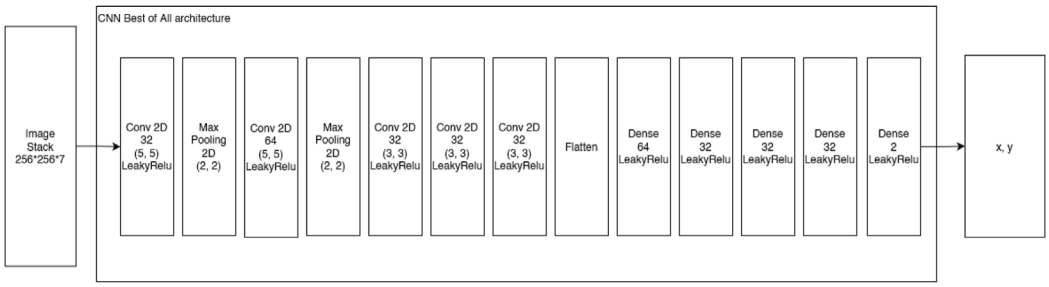


Fig. A.24. Final model.

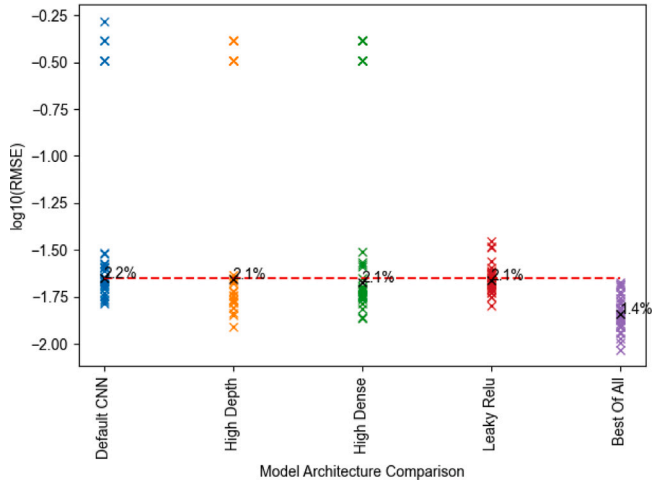


Fig. A.25. Performances of the best model architectures on dataset 14.

References

[1] S.J. Erasmus, K.C.A. Smith, An automatic focusing and astigmatism correction system for the SEM and CTEM, J. Microsc. 127 (2) (1982) 185–199, http://dx.doi.org/10.1111/j.1365-2818.1982.tb00412.x, URL https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2818.1982.tb00412.x.

[2] W.O. Saxton, D.J. Smith, S.J. Erasmus, Procedures for focusing, stigmating and alignment in high resolution electron microscopy, J. Microsc. 130 (2) (1983) 187–201, http://dx.doi.org/10.1111/j.1365-2818.1983.tb04217.x, URL https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2818.1983.tb04217.x.

[3] A. Koster, A. Van den Bos, K. van der Mast, An autofocus method for a TEM, Ultramicroscopy 21 (3) (1987) 209–222, http://dx.doi.org/10.1016/0304-3991(87)90146-X, URL https://linkinghub.elsevier.com/retrieve/pii/030439918790146X.

[4] K. Dierksen, D. Typke, R. Hegerl, A. Koster, W. Baumeister, Towards automatic electron tomography, Ultramicroscopy 40 (1) (1992) 71–87, http://dx.doi.org/10.1016/0304-3991(92)90235-C, URL https://linkinghub.elsevier.com/retrieve/pii/030439919290235C.

[5] O. Krivanek, P. Mooney, Applications of slow-scan CCD cameras in transmission electron microscopy, Ultramicroscopy 49 (1–4) (1993) 95–108, http://dx.doi.org/10.1016/0304-3991(93)90216-K, URL https://linkinghub.elsevier.com/retrieve/pii/030439919390216K.

[6] S. Uhlemann, M. Haider, Residual wave aberrations in the first spherical aberration corrected transmission electron microscope, Ultramicroscopy 72 (3) (1998) 109–119, http://dx.doi.org/10.1016/S0304-3991(97)00102-2, URL https://www.sciencedirect.com/science/article/pii/S0304399197001022.

[7] Y.Z. Tan, A. Cheng, C.S. Potter, B. Carragher, Automated data collection in single particle electron microscopy, Microscopy 65 (1) (2016) 43–56, http://dx.doi.org/10.1093/jmicro/dfv369.

[8] A. Tejada, W. Van den Broek, A.J. den Dekker, Measure-by-wire (MBW), in: Advances in Imaging and Electron Physics, Vol. 179, Elsevier, 2013, pp. 291–346, http://dx.doi.org/10.1016/B978-0-12-407700-3.00005-3, URL https://linkinghub.elsevier.com/retrieve/pii/B9780124077003000053.

[9] C. Gatel, J. Dupuy, F. Houdellier, M.J. Hÿtch, Unlimited acquisition time in electron holography by automated feedback control of transmission electron microscope, Appl. Phys. Lett. 113 (13) (2018) 133102, http://dx.doi.org/10.1063/1.5050906, URL https://pubs.aip.org/apl/article/113/13/133102/35469/Unlimited-acquisition-time-in-electron-holography.

[10] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybernet. 36 (4) (1980) 193–202, http://dx.doi.org/10.1007/BF00344251, URL http://link.springer.com/10.1007/BF00344251.

[11] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, Handwritten digit recognition with a back-propagation network, Neural Inf. Process. Syst. (1989).

[12] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. 61 (2015) 85–117, http://dx.doi.org/10.1016/j.neunet.2014.09.003, arXiv:1404.7828 [cs].

[13] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539, URL https://www.nature.com/articles/nature14539.

[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90, http://dx.doi.org/10.1145/3065386, URL https://dl.acm.org/doi/10.1145/3065386.

[15] J.M. Ede, Review: Deep learning in electron microscopy, 2021, http://dx.doi.org/10.1088/2632-2153/abd614, arXiv:2009.08328 [cond-mat].

[16] K.P. Treder, C. Huang, J.S. Kim, A.I. Kirkland, Applications of deep learning in electron microscopy, Microscopy (Oxford, England) 71 (Supplement_1) (2022) i100–i115, http://dx.doi.org/10.1093/jmicro/dfab043.

[17] A. Suveer, A. Gupta, G. Kylberg, I.-M. Sintorn, Super-resolution reconstruction of transmission electron microscopy images using deep learning, in: 2019 IEEE 16th International Symposium on Biomedical Imaging, ISBI 2019, IEEE, Venice, Italy, 2019, pp. 548–551, http://dx.doi.org/10.1109/ISBI.2019.8759153, URL https://ieeexplore.ieee.org/document/8759153/.

[18] G. Roberts, S. Haile, R. Sainju, D. Edwards, B. Hutchinson, Y. Zhu, Deep Learning for Semantic Segmentation of Defects in Advanced STEM Images of Steels, Sci. Rep. 9 (2019) http://dx.doi.org/10.1038/s41598-019-49105-0.

[19] W. Lee, H.S. Nam, Y.G. Kim, Y.J. Kim, J.H. Lee, H. Yoo, Robust autofocusing for scanning electron microscopy based on a dual deep learning network, Sci. Rep. 11 (1) (2021) 20933, http://dx.doi.org/10.1038/s41598-021-00412-5, URL https://www.nature.com/articles/s41598-021-00412-5.

[20] P. Rosi, A. Clausen, D. Weber, A.H. Tavabi, S. Frabboni, P. Tiemeijer, R.E. Dunin-Borkowski, E. Rotunno, V. Grillo, Automatic alignment of an orbital angular momentum sorter in a transmission electron microscope using a convolution neural network, Microsc. Microanal. 29 (1) (2021) 408–417, http://dx.doi.org/10.1017/S143192762201248X, URL https://arxiv.org/abs/2111.05032.

[21] C. Zhang, Z. Baraissov, C. Duncan, A. Hanuka, A. Edelen, J. Maxson, D. Muller, Aberration corrector tuning with machine-learning-based emittance measurements and Bayesian optimization, Microsc. Microanal. 27 (S1) (2021) 810–812, http://dx.doi.org/10.1017/S1431927621003214, 3 citations (Crossref) [2022-10-17].

[22] R. Sagawa, F. Uematsu, K. Aibara, T. Nakamichi, S. Morishita, Aberration measurement and correction in scanning transmission electron microscopy using machine learning, Microsc. Microanal. 27 (S1) (2021) 814–816, http://dx.doi.org/10.1017/S1431927621003226, 1 citations (Crossref) [2022-10-17].

[23] M. De Graef, Introduction to Conventional Transmission Electron Microscopy, Cambridge University Press, Cambridge, 2003, http://dx.doi.org/10.1017/CBO9780511615092, URL https://www.cambridge.org/core/books/introduction-to-conventional-transmission-electron-microscopy/257FBB684B79174B1C6D3ACCC256ECC0.

[24] C. Gatel, A. Masseboeuf, E. Snoeck, F. Bonnilla, T. Blon, L.-M. Lacroix, A. Meffre, J. Einsle, R. Bowman, M. Bashir, M. Gubbins, Off-axis electron holography for the quantitative study of magnetic properties of nanostructures: From the single nanomagnet to the complex device, Microsc. Microanal. 21 (S3) (2015) 2147–2148, http://dx.doi.org/10.1017/S1431927615011514.

[25] J. Dupuy, Contrôle Dynamique et Optimisation des Observations en Microscopie Électronique en Transmission (Ph.D. thesis), Toulouse 3 - Paul Sabatier, 2021.

[26] Y. Kubo, C. Gatel, E. Snoeck, F. Houdellier, Optimising electron microscopy experiment through electron optics simulation, Ultramicroscopy 175 (2017) 67–80, http://dx.doi.org/10.1016/j.ultramic.2017.01.007, 6 citations (Crossref) [2022-10-17].

[27] S. Bozinovski, Reminder of the first paper on transfer learning in neural networks, 1976, Informatica (Ljubl.) 44 (3) (2020) http://dx.doi.org/10.31449/inf.v44i3.2828, URL http://www.informatica.si/index.php/informatica/article/view/2828.

[28] H.-C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (5) (2016) 1285–1298, http://dx.doi.org/10.1109/TMI.2016.2528162, URL https://ieeexplore.ieee.org/document/7404017/.

[29] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, http://dx.doi.org/10.48550/arXiv.1502.03167, arXiv:1502.03167 [cs].

[30] L. Lu, Y. Shin, Y. Su, G.E. Karniadakis, Dying ReLU and initialization: Theory and numerical examples, Commun. Comput. Phys. 28 (5) (2020) 1671–1706, http://dx.doi.org/10.4208/cicp.OA-2020-0165, arXiv:1903.06733 [cs, math, stat].

[31] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de las Casas, C. Donner, L. Fritz, C. Galperti, A. Huber, J. Keeling, M. Tsimpoukelli, J. Kay, A. Merle, J.-M. Moret, S. Noury, F. Pesamosca, D. Pfau, O. Sauter, C. Sommariva, S. Coda, B. Duval, A. Fasoli, P. Kohli, K. Kavukcuoglu, D. Hassabis, M. Riedmiller, Magnetic control of tokamak plasmas through deep reinforcement learning, Nature 602 (7897) (2022) 414–419, http://dx.doi.org/10.1038/s41586-021-04301-9, 47 citations (Crossref) [2022-10-25] Number: 7897 Publisher: Nature Publishing Group.

[32] D. Arumugam, J.K. Lee, S. Saskin, M. Littman, Deep reinforcement learning from policy-dependent human feedback, Conf. Neural Inf. Process. Syst. (2019) http://dx.doi.org/10.48550/arXiv.1706.03741.

[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, 2013, arXiv:1312.5602 [cs].