

# Automated tip functionalization via machine learning in scanning probe microscopy ☆, ☆☆



Benjamin Alldritt <sup>a,1</sup>, Fedor Urtev <sup>a,b,1</sup>, Niko Oinonen <sup>a</sup>, Markus Aapro <sup>a</sup>, Juho Kannala <sup>b</sup>, Peter Liljeroth <sup>a,\*</sup>, Adam S. Foster <sup>a,c,\*\*</sup>

<sup>a</sup> Department of Applied Physics, Aalto University, P.O. Box 11100, 00076 Aalto, Espoo, Finland

<sup>b</sup> Department of Computer Science, Aalto University, P.O. Box 11100, 00076 Aalto, Espoo, Finland

<sup>c</sup> WPI Nano Life Science Institute (WPI-NanoLSI), Kanazawa University, Kakuma-machi, Kanazawa 920-1192, Japan

## ARTICLE INFO

### Article history:

Received 5 August 2021

Received in revised form 18 November 2021

Accepted 10 December 2021

Available online 16 December 2021

Dataset link: <https://www.dropbox.com/s/wqhgbdvdlfb6snic/datasetNew6.tar.gz?dl=0>

### Keywords:

Atomic force microscopy

Machine learning

Tip preparation

Automation

Carbon monoxide

## ABSTRACT

Auto-CO-AFM is an open-source software package for scanning probe microscopes that enables the automatic functionalization of scanning probe tips with carbon monoxide molecules. This enables machine operators to specify the quality of the tip needed utilizing a pre-trained library with off-the-shelf software. From a single image, the software package can determine which molecules on a surface are carbon monoxide, perform the necessary tip functionalization procedures, interface with microscope software to control the tip position, and determines the centeredness of the tip after a successful functionalization. This is of particular interest for atomic force microscopy imaging of molecules on surfaces, where the tip functionalization is a necessary and time consuming step needed for sub-molecular resolution imaging. This package is freely available under the MIT License.

### Program summary

*Program Title:* Auto-CO-AFM

*CPC Library link to program files:* <https://doi.org/10.17632/khfvjhx4vb.1>

*Developer's repository link:* <https://github.com/SINGROUP/Auto-CO-AFM>

*Licensing provisions:* MIT

*Programming language:* Python

*Supplementary material:* Supplementary Information as PDF

*Nature of problem:* Scanning probe microscopy experiments are limited by the lack of automated tools in tip preparation, particularly when functionalizing an adatom or molecule on the end of a tip. These tip functionalizations are commonly manually-driven processes that are time-consuming for the operator.

*Solution method:* We have developed an automated solution for tip functionalization for carbon monoxide molecules, which combines machine learning descriptors with automated software control of the tip preparation process. The Python program interfaces with a major vendor's software to scan, functionalize, and verify the tip quality for the operator. The source code, documentation, and basic program are posted online.

*Additional comments including restrictions and unusual features:* Current version of software in publication requires STMAFM software from CreaTec GmbH in physical connection with a CreaTec DSP.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

☆ The review of this paper was arranged by Prof. Volker Blum.

☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

\* Corresponding author.

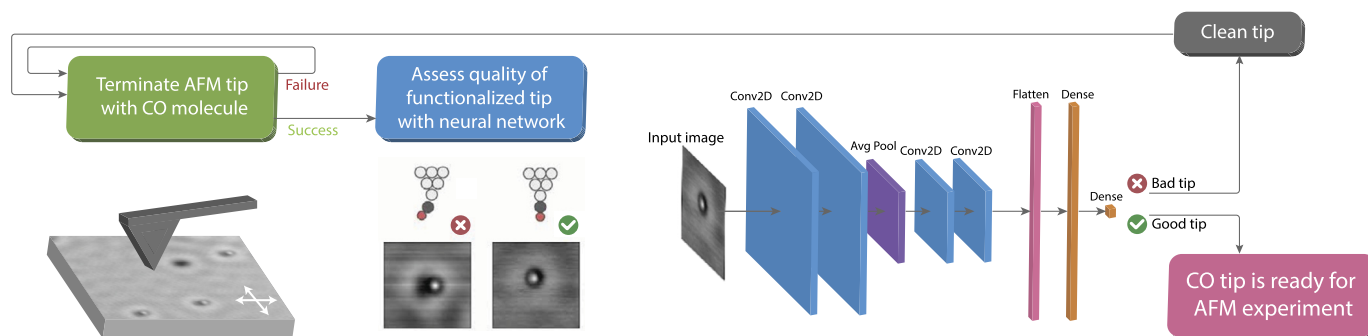
\*\* Corresponding author at: Department of Applied Physics, Aalto University, P.O. Box 11100, 00076 Aalto, Espoo, Finland.

E-mail addresses: [peter.liljeroth@aalto.fi](mailto:peter.liljeroth@aalto.fi) (P. Liljeroth), [adam.foster@aalto.fi](mailto:adam.foster@aalto.fi) (A.S. Foster).

<sup>1</sup> These authors contributed equally.

## 1. Introduction

The leading techniques in Scanning Probe Microscopy (SPM), Scanning Tunneling Microscopy (STM) [1] and Atomic Force Microscopy (AFM) [2] have enabled the investigation of surfaces [3–5] and adsorbates on surfaces with atomic precision for a wide range of materials. STM utilizes an atomically sharp and biased metal tip to measure small currents from a conductive surface, based on the quantum mechanical tunnelling effect. On the other hand, AFM measures the interaction force between the tip and the substrate.



**Fig. 1.** Schematic pipeline of the autonomous tip preparation procedure. The forward direction is from left to right: a bare metal tip picks up individual CO molecule from the Cu(111) substrate; a tip terminated with a CO molecule scans over the surface to get the STM image of CO molecules on the substrate; STM images are used as input to a convolutional neural network (CNN) which transforms individual images into a classification of the CO-tip centeredness; based on quality of the tip, a decision is made whether to clean it and repeat the functionalization procedure, or to follow next step of STM/AFM experiment preparation.

Using modern non-contact AFM approaches operating in the frequency modulation mode [6] and with stiff cantilevers and small oscillation amplitudes [7–9], it is possible to reach a regime where the tip-sample interaction is dominated by the chemical interactions between the last atom of the tip with the topmost substrate atoms. Finally, it is possible to integrate both STM and AFM modes in the same experimental setup using quartz tuning fork force sensors, which allows mapping the STM and AFM responses of the surface simultaneously. The spatial contrast of STM and AFM depends on the geometric structure and the chemical species at the end of the tip [10]. Developments in the capability of vertical atomic and molecular manipulation in SPM [11], means that it is possible to functionalize the end of the SPM tip with single atoms and molecules [12–19]. In particular, by terminating the end of the metal tip with a carbon monoxide (CO) molecule [13], it is possible to reproducibly image organic molecules with sub-molecular resolution [14,20] in the AFM mode. However, despite the well-defined process for functionalizing a CO molecule onto a metal tip, the process is not guaranteed to produce the required tip for experimental imaging in terms of the required tip stability, symmetry and the lack of spurious features in the resulting images.

There has been significant recent interest in automated methods for preparing and analyzing tip quality for the operator [21–39]. These advances have helped move the field of SPM forward by reducing the time and resources spent on the preparation of metallic tips. Of these different methods, only a few have focused on a truly automated approach to tip conditioning and all have been focused only on STM tip preparation.

In this work, we build upon earlier ideas in automated STM tip preparation to automatically prepare functionalized CO tips. This is made possible through the application of a convolutional neural network (CNN) model [40] to identify the quality of CO functionalization, in conjunction with automated processes with experimental hardware to prepare a functionalized CO tip. To identify tip quality we take advantage of the fact that a CO molecule adsorbed on a Cu(111) surface can be utilized to image the tip apex, which is also the basis of the widely used carbon-monoxide front atom identification (COFI) method [41–43]. We perform STM imaging with a CO-functionalized tip apex, which gives characteristic sombrero-shaped images of the surface adsorbed CO [13] (see Fig. 1). The symmetry of these images allows determination of the configuration of the tip-adsorbed CO molecule and also distinguishes CO from other adsorbates on the substrate. Auto-CO-AFM provides a working model to identify CO molecules from a variety of other impurities, control the hardware to perform spectroscopy on a particular CO molecule, and then confirm that the tip has been functionalized and the quality of the functionalization.

This paper is organized as follows. In Section 2, we present the experimental methodology and in Section 3 the computational ap-

proach. In Section 4, we introduce an example of functionalizing a metal tip with a carbon monoxide molecule on a Cu(111) surface. In Section 5, we introduce a general overview of the software. In Section 6, we introduce the installation and basic usage.

## 2. Experimental setup

All experiments were performed with a Createc LT-STM/AFM with a commercial qPlus sensor with a Pt/Ir tip, operating at approximately  $T = 5$  K in UHV at a pressure of  $1 \times 10^{-10}$  mbar. Tips were sharpened initially by electrochemical etching, then with focused ion beam [44] to  $\sim 20$  nm. *In situ* tip conditioning was performed with multiple, controlled indentations into the metallic surface and applying bias pulses up to 2 V until the tip was sufficiently coated in Cu to observe single atom resolution. A polished Cu(111) single-crystal (MaTeck/Germany) was prepared by repeated Ne<sup>+</sup> sputtering (0.75 keV, 15 mA, 20 min) and annealing (850–900 K, 5 min) cycles. Surface cleanliness was evaluated for impurities and terrace size using scanning tunneling microscopy (STM). Sample temperatures during annealing were measured with a pyrometer (SensorTherm Metis MI16).

CO was deposited onto the surface via a leak-valve connected to the microscope chamber. The shutter door was open for 10 s while CO gas was leaked into the chamber at a pressure of  $1 \times 10^{-6}$  mbar. During this time, the estimated sample temperature was  $< 30$  K. After deposition, the system was cooled to approximately  $T = 5$  K and CO coverage was verified via STM. During the experiments, the tip apex was functionalized with a CO molecule [13] and checked first by an operator, then later by the automated protocols described in this paper. The STM images were recorded in constant-current mode at multiple setpoint current and bias values.

## 3. Machine learning architecture

We consider a binary classification problem of assessing the quality of a CO functionalized tip to be good or bad. Input data is spatially ordered due to pixels' relations in STM images. This makes it reasonable to consider a CNN approach for classification. The model is implemented with the Tensorflow [45] machine learning package in Python. Our implementation of the model and the trained weights can be found at our Github page [46].

We present here an artificial neural network (ANN) architecture which includes a CNN based encoder and a binary classifier (see (Table 1)). The encoder part of the network has two convolutions blocks, each with two 2D convolutional layers ('2D Conv' on Fig. 1). The first block is followed by the Average Pooling layer that reduces the size of activation maps by a factor of 2 in (x, y) dimensions. The amount of channels remains unchanged. Then we

**Table 1**  
Model architecture. Amount of trainable parameters: 2157.

| Layer type | Output dimension        | Kernel size  | Stride | Parameters |
|------------|-------------------------|--------------|--------|------------|
| Input      | $16 \times 16 \times 1$ | –            | –      | –          |
| 2D Conv    | $14 \times 14 \times 4$ | $3 \times 3$ | (1, 1) | 40         |
| 2D Conv    | $12 \times 12 \times 4$ | $3 \times 3$ | (1, 1) | 148        |
| Avg Pool   | $6 \times 6 \times 4$   | $2 \times 2$ | (2, 2) | –          |
| Dropout    | $6 \times 6 \times 4$   | –            | –      | –          |
| 2D Conv    | $4 \times 4 \times 8$   | $3 \times 3$ | (1, 1) | 296        |
| 2D Conv    | $2 \times 2 \times 8$   | $3 \times 3$ | (1, 1) | 584        |
| Dropout    | $2 \times 2 \times 8$   | –            | –      | –          |
| Flatten    | 32                      | –            | –      | –          |
| Dense      | 32                      | –            | –      | 1056       |
| Dense      | 1                       | –            | –      | 33         |

apply a Dropout layer which randomly vanishes input values with an adjustable rate. A Dropout (0.25) is applied after each of two convolutional blocks (basically randomly vanishing 1/4 of inputs). The number of output channels of ‘2D Conv’ layers is 4 in the first block, and it doubles to 8 in the second block. The main idea of using two similar sets of convolutional layers is that they process input images with a different level of complexity. The filter size for 2D convolution layers is  $3 \times 3$  and stride equals 1. There is no padding before the convolutional layers. The pooling layer has kernel size  $2 \times 2$  with the same stride, which makes it a non-overlapping operation.

In the classifier head, which follows the encoder, the spatially structured convolutional layers are flattened into fully connected layers. The classification is performed with two dense (fully-connected) layers (‘Dense’ on Fig. 1) with sizes 32 and 1. Output from last layer then yields us a prediction of tip quality measured in a range from 0 (Bad CO tip) to 1 (Good CO tip).

Activation functions follow each layer of the CNN to introduce non-linearity in the process of collecting features. The Rectified Linear Unit (ReLU) activation function was applied for all layers except the last one:

$$\text{ReLU}(z) = \max(0, z), \quad (1)$$

which basically cuts the negative part of activation from convolution layer leaving unchanged positive values.

The last layer uses the Sigmoid activation function:

$$S(z) = \frac{1}{1 + e^{-z}}, \quad (2)$$

which fits output values into a [0,1] range for assessment of the CO tip’s quality.

This type of architecture allows the effective feature extraction with a minimal number of trainable parameters.

### 3.1. Loss function

For a binary classification problem the standard loss function is a binary cross entropy:

$$\text{Loss} = -(y \log(p) + (1 - y) \log(1 - p)), \quad (3)$$

where  $y$  – true values, and  $p$  – predictions of a model. We choose the Adaptive Moment Estimation (Adam) [47] as optimizer for the gradient descent. We set the learning rate to 0.001 and the decay to  $10^{-5}$ , otherwise we use the default parameters as defined in Keras [48].

### 3.2. Training data

The challenge of assessing tip quality is a binary classification problem. Human-labeled experimental STM data was used to train

the CNN classifier. Our database contains 21 and 45 images with multiple bad and good samples correspondingly. Each STM image includes several CO molecules which are tilted based on the tip functionalization orientation. To ensure a direct comparison, a SURF algorithm was applied to split and rescale each CO molecule to  $16 \times 16$  pixels. The total dataset consists of 346 samples of CO molecules with approximately an 80/20 train/test split ratio. Individual images of CO-terminated tips have strong correlation to other images cropped from the same experimental STM image, since the orientation of the functionalized CO remains the same. In order to ensure that the test set is uncorrelated with the training set, the sets should be split by source STM images. Augmentation was applied for training data (flips, rotations) and regularization during training process (Dropouts) [49] in order to expand beyond the limited amount of available experimental data.

## 4. Results

To visualize the performance of CNN classifier we plotted its results on a test set with experimental AFM data sorted by most and least accurate predictions in Fig. 3. We can conclude that with the default settings of threshold = 0.5 for splitting into positive and negative classes, very good results are obtained.

### 4.1. Accuracy metrics

A more accurate picture of results is presented in a confusion matrix (see in the left at Fig. 4). A confusion matrix describes the performance of a classification model on a test data set. It has four cells which count the following events: true negative prediction (TN), false positive (FP), false negative (FN) and true positive (TP). It allows easy identification of mislabeling between classes. We consider here positive and negative events correspond to good or bad CO tips detected. Most performance measures are computed from the confusion matrix components:

Accuracy (ACC) is given by the relation:

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4)$$

and reflects the rate of correctly classified CO tips.

Sensitivity, Recall or True Positive Rate (TPR) is the ratio of correctly predicted good tip examples divided by true number of good CO tips. High Recall indicates the class is correctly recognized (small number of FN). Recall is given by the relation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

Value of *precision* is the number of correctly classified good CO tip samples divided by the total number of tips predicted as good. A high Precision indicates that samples labeled as good are indeed good (small number of FP). Precision is given by the relation:

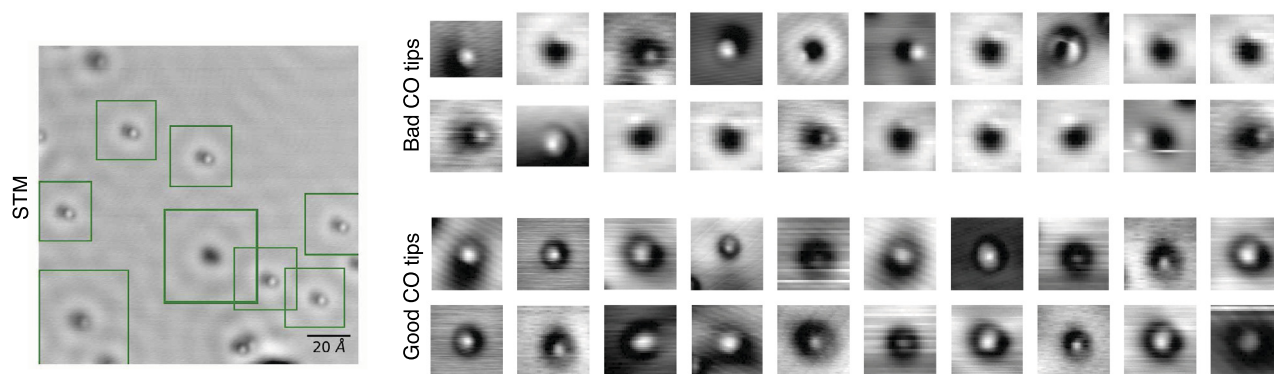
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6)$$

An *F-score* combines Precision and Recall by use of an Harmonic Mean in place of an Arithmetic Mean, punishing extreme values more. The F-score will always be nearer to the smaller value of Precision or Recall.

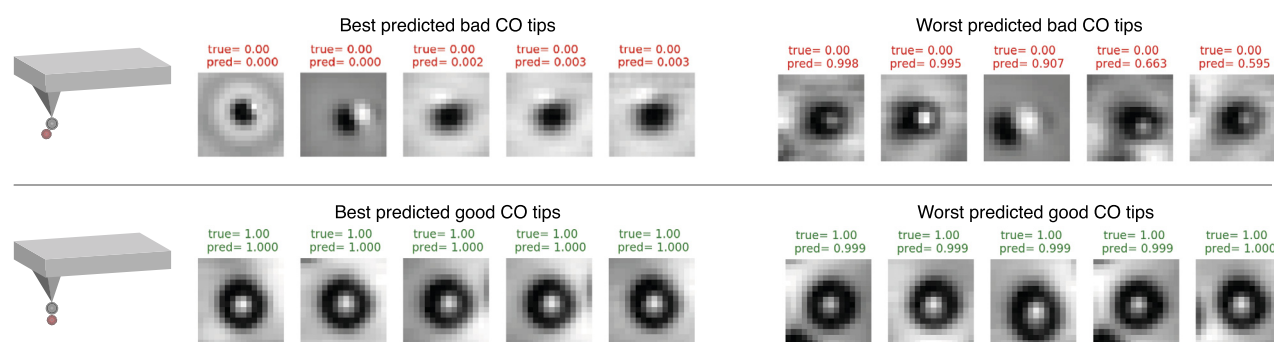
$$\text{F-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \quad (7)$$

*Fall-out* or False Positive Rate (FPR) is the probability of a bad CO tip being predicted as a good one:

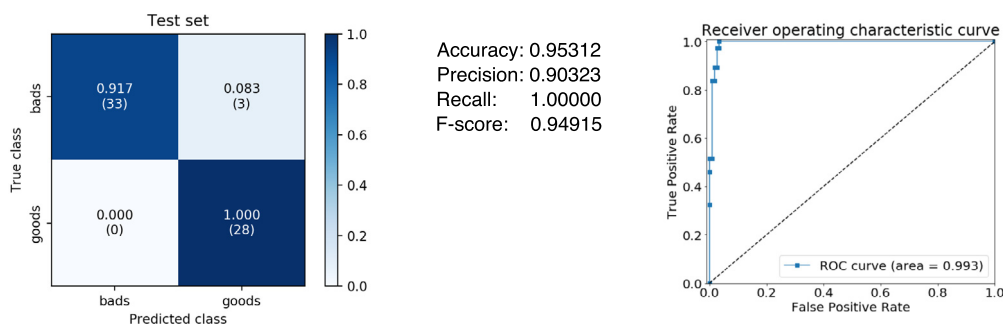
$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (8)$$



**Fig. 2.** Formation of the training database with CO-functionalized tips. *Left* Example experimental image with identified surface COs highlighted. *Right* Examples from the training dataset. *Top-right* Random 20 bad CO-functionalized tips on Cu(111) substrate at similar tip-sample distances from our training database are shown. The CO tips are not centered or contain significant measurement artifacts. *Bottom-right* Random 20 good CO-functionalized tips on Cu(111) substrate at similar tip-sample distances from our training database are shown. CO tips are centered, which meets the target tip state.



**Fig. 3.** An illustration of a trained CNN classifier performance on STM images of individual CO molecules from the test set. Samples are sorted by the accuracy of the predicted class: with the best predictions (on the left) and the worst predictions (on the right) for the both classes. Bad CO tip examples are on the top row and the good CO tips are on the bottom.



**Fig. 4.** A summary of prediction results of CO tips classifier: a confusion matrix at 0.9 confidence level, numerical metrics of success and a ROC curve. In the confusion matrix, the number of correct and incorrect predictions are summarized with count values and broken down by each class: true negative prediction (TN), false positive (FP), false negative (FN) and true positive (TP). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

In the goal of automatizing the assessment of CO tip quality, we care mostly about reducing the false positive rate (FPR) and suffer less from false negative cases. In other words, we want to find a balance between TPR and FPR which are directly affected by setting of discrimination threshold (initially is 0.5) to split probability between two classes. A receiver operating characteristic curve, or ROC curve, is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings (see in the right of Fig. 4). The area under the curve (AUC = 0.993) indicates the quality of the classification model in comparison to a random guess model, along the diagonal with AUC = 0.5, and best model with AUC = 1.0. On this plot, reducing of the threshold parameter initially increases TPR very quickly at a low FPR (that is a good thing), and later as the FPR increases, the TPR remains constant. The optimal threshold can be always adjusted later, but

for the pretrained model on the current test data set the value of threshold = 0.9 looks to be a reasonable choice. Overall, all accuracy metrics demonstrate reliable results for the proposed CO tip quality classifier.

## 5. Software overview

We utilize machine learning to automate tip functionalization for CO-terminated tips on a Cu(111) surface. It is based on a CNN pipeline Fig. 1, which is trained to recognize and assess the centeredness of a CO-terminated tip. By averaging the prediction values of the visible CO molecules on a Cu(111) substrate, it is possible to determine the CO functionalization quality. This process is repeated until a target CO tip is found.



1. The user prepares a metallic tip to single atom resolution.
2. Utilizing a Speeded Up Robust Features (SURF) algorithm [50], the software identifies the target CO molecule, then performs vertical spectroscopy over the center of the CO with the following parameters: 2.6 V at an initial set-point of 0.1 V and 100 pA.
3. Rescanning the same area confirms the success or failure of tip functionalization. After multiple tip functionalization failures, the software will attempt a tip cleaning with user defined parameters at a location away from the target area. If multiple failures are still detected, the system will return control back to the user.
4. In case of successful tip termination, the CO centeredness is determined. To achieve high resolution and details in AFM experiment, the CO molecule should terminate the tip as close as possible to orthogonal to Cu(111) surface. After the tip termination procedure the tip CO molecule is obviously no longer visible and the quality of CO-functionalized tip is assessed by imaging of other CO molecules on a Cu(111) surface.
5. A local features detector algorithm is utilized to grab individual images for each distinguishable CO molecule on a Cu(111) surface (see in the left of Fig. 2). SURF is again used to automatically split STM images into individual images of CO molecules. A random subset of such individual tip's images demonstrate broken symmetry for bad CO tips and high centeredness for good CO tips (see in the right of Fig. 2).
6. At the last stage of automated process for preparing a high-quality tip for an AFM experiment we implement a trained classifier based on artificial neural networks. The trained model outputs a probability value which measures CO tip centeredness (with a range from 0 – for bad CO tips to 1 – for good CO tips). *Bad* CO tips are not centered or contain significant measurement artifacts (see the top-right of Fig. 2). *Good* CO tips are centered, which meets the target tip state (see the bottom-right of Fig. 2). Averaging of prediction values achieved for visible CO molecules on a Cu(111) substrate provides a metric for the CO functionalized tip quality. A CNN model was trained to accomplish this task, using a human-labeled dataset of experimental STM images with CO molecules for training.

## 6. Software installation and usage

The Auto-CO-AFM software package is fully open source, released on Github under the MIT License, <https://github.com/SINGROUP/Auto-CO-AFM/blob/main/LICENSE>, and it can be downloaded directly from the public code archive: <https://github.com/SINGROUP/Auto-CO-AFM>.

To run the Auto-CO software with CreaTec STM integration, it is necessary to install Anaconda on Windows along with the CreaTec STMAFM software (or use a virtual environment to run the CreaTec software in another OS), in addition to the CreaTec STMAFM COM Automation Server. In Anaconda, create the required Python environment with:

```
$ conda env create -f environment.yml
```

This will create a conda environment named `tf-gpu` with the all the required packages. It also has a suitable version of the CUDA toolkit and cuDNN already installed. Activate the environment with:

```
$ conda activate py3-tf12
```

To create the datasets and train the models, run a Jupyter notebook in the repository folder, open the [train\\_TF.ipynb](#) notebook, and follow the instructions therein. The folder [pretrained\\_weights](#) holds the weights for pretrained model. To predict quality of

CO tips on some set of images, open the [predict\\_TF.ipynb](#) notebook, and follow the instructions therein. To run an automated CO pickup, open [auto-co.ipynb](#) and follow the instructions.

## 7. Conclusions

In conclusion, we have introduced an efficient automated tip functionalization method which is capable of determining the tip functionalization quality for SPM experiments and can perform tip conditioning until the target tip functionalization is found. This method is based on a CNN that continually evaluates the tip state by reviewing over images for changes in the tip condition. This was tested by training a system on images of centered and non-centered CO tips on a Cu(111) substrate. For experiments requiring functionalized tips, the increased efficiency in preparation translates directly into additional experimental measurement time. This method requires no additional hardware to implement, allowing for its application in existing SPM laboratories. The combination of existing, open-source libraries and a small dataset would enable any size of lab to effectively utilize this method. For future developments, we would anticipate that this method would work on a variety of different substrates and tip types, leading to a wider variety of tip functionalizations compared to the widely-used standard of CO tips.

## Credit author statement

**Benjamin Alldritt:** Conceptualization, Methodology, Software, Data curation, Validation, Visualization, Writing-Original draft preparation. **Fedor Urtev:** Methodology, Software, Data curation, Visualization, Writing-Original draft preparation. **Niko Oinonen:** Software. **Markus Aapro:** Conceptualization, Validation. **Juho Kannala:** Conceptualization, Supervision. **Peter Liljeroth:** Conceptualization, Supervision, Visualization, Writing-Reviewing and Editing. **Adam S. Foster:** Conceptualization, Supervision, Writing-Original draft preparation, Writing-Reviewing and Editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Code availability

The source code and the trained weights for the model in TensorFlow available at: <https://github.com/SINGROUP/Auto-CO-AFM>.

## Data availability

Our dataset for classification is available at <https://www.dropbox.com/s/wqhgbd1fb6snc/datasetNew6.tar.gz?dl=0>.

## Acknowledgements

Computing resources from the Aalto Science-IT project and CSC, Helsinki are gratefully acknowledged. This research made use of the Aalto Nanomicroscopy Center (Aalto NMC) facilities and was supported by the European Research Council (ERC 2017 AdG no. 788185 “Artificial Designer Materials”), and the Academy of Finland (Projects no. 311012, 314862, 314882, Centres of Excellence Program project no. 284621, and Academy professor funding no. 318995 and 320555). ASF has been supported by the World Premier International Research Center Initiative (WPI), MEXT, Japan.

## References

- [1] G. Binnig, H. Rohrer, C. Gerber, E. Weibel, *Phys. Rev. Lett.* 49 (1982) 57–61, <https://doi.org/10.1103/PhysRevLett.49.57>.
- [2] G. Binnig, C.F. Quate, C. Gerber, *Phys. Rev. Lett.* 56 (1986) 930–933, <https://doi.org/10.1103/PhysRevLett.56.930>.
- [3] G. Binnig, H. Rohrer, C. Gerber, E. Weibel, *Phys. Rev. Lett.* 50 (1983) 120–123, <https://doi.org/10.1103/PhysRevLett.50.120>.
- [4] F.J. Giessibl, *Science* 267 (5194) (1995) 68–71, <https://doi.org/10.1126/science.267.5194.68>.
- [5] M.F. Crommie, C.P. Lutz, D.M. Eigler, *Science* 262 (5131) (1993) 218–220, <https://doi.org/10.1126/science.262.5131.218>.
- [6] T.R. Albrecht, P. Grütter, D. Horne, D. Rugar, *J. Appl. Phys.* 69 (2) (1991) 668–673, <https://doi.org/10.1063/1.347347>.
- [7] F. Giessibl, *Rev. Mod. Phys.* 75 (2003) 949–983.
- [8] W. Hofer, A. Foster, A. Schluger, *Rev. Mod. Phys.* 75 (4) (2003) 1287–1331, <https://doi.org/10.1103/RevModPhys.75.1287>.
- [9] F.J. Giessibl, *Rev. Sci. Instrum.* 90 (1) (2019) 011101, <https://doi.org/10.1063/1.5052264>.
- [10] N. Moll, L. Gross, F. Mohn, A. Curioni, G. Meyer, *New J. Phys.* 12 (12) (2010) 125020, <https://doi.org/10.1088/1367-2630/12/12/125020>.
- [11] D.M. Eigler, C.P. Lutz, W.E. Rudge, *Nature* 352 (6336) (1991) 600–603, <https://doi.org/10.1038/352600a0>.
- [12] J.A. Stroscio, D.M. Eigler, *Science* 254 (5036) (1991) 1319–1326, <https://doi.org/10.1126/science.254.5036.1319>.
- [13] L. Bartels, G. Meyer, K.-H. Rieder, *Appl. Phys. Lett.* 71 (2) (1997) 213–215, <https://doi.org/10.1063/1.119503>.
- [14] L. Gross, F. Mohn, N. Moll, P. Liljeroth, G. Meyer, *Science* 325 (5944) (2009) 1110–1114, <https://doi.org/10.1126/science.1176210>.
- [15] R. Temirov, S. Soubatch, O. Neucheva, A.C. Lassise, F.S. Tautz, *New J. Phys.* 10 (5) (2008) 053012, <https://doi.org/10.1088/1367-2630/10/5/053012>.
- [16] G. Kichin, C. Weiss, C. Wagner, F.S. Tautz, R. Temirov, *J. Am. Chem. Soc.* 133 (42) (2011) 16847–16851, <https://doi.org/10.1021/ja204624g>.
- [17] F. Mohn, B. Schuler, L. Gross, G. Meyer, *Appl. Phys. Lett.* 102 (7) (2013) 073109, <https://doi.org/10.1063/1.4793200>.
- [18] A.M. Sweetman, S.P. Jarvis, H. Sang, I. Lekkas, P. Rahe, Y. Wang, J. Wang, N.R. Champness, L. Kantorovich, P. Moriarty, *Nat. Commun.* 5 (1) (2014) 3931, <https://doi.org/10.1038/ncomms4931>.
- [19] H. Mönig, D.R. Herness, O. Díaz Arado, M. Todorović, A. Timmer, S. Schüer, G. Langewisch, R. Pérez, H. Fuchs, *ACS Nano* 10 (1) (2016) 1201–1209, <https://doi.org/10.1021/acsnano.5b06513>.
- [20] L. Gross, B. Schuler, N. Pavliček, S. Fatayer, Z. Majzik, N. Moll, D. Peña, G. Meyer, *Angew. Chem., Int. Ed.* 57 (15) (2018) 3888–3908, <https://doi.org/10.1002/anie.201703509>.
- [21] J.A. Stroscio, E.W. Hudson, S.R. Blankenship, R.J. Celotta, A.P. Fein, in: M.C. Peckler, M.T.P. Jr. (Eds.), *Nanostructure Science, Metrology, and Technology*, in: *International Society for Optics and Photonics*, vol. 4608, SPIE, 2002, pp. 112–115.
- [22] R.A.J. Woolley, J. Stirling, A. Radocea, N. Krasnogor, P. Moriarty, *Appl. Phys. Lett.* 98 (25) (2011) 253104, <https://doi.org/10.1063/1.3600662>.
- [23] J. Stirling, R.A.J. Woolley, P. Moriarty, *Rev. Sci. Instrum.* 84 (11) (2013) 113701, <https://doi.org/10.1063/1.4827076>.
- [24] J.C. Straton, T.T. Bilyeu, B. Moon, P. Moeck, *Cryst. Res. Technol.* 49 (9) (2014) 663–680, <https://doi.org/10.1002/crat.201300240>.
- [25] R.J. Celotta, S.B. Balakirsky, A.P. Fein, F.M. Hess, G.M. Rutter, J.A. Stroscio, *Rev. Sci. Instrum.* 85 (12) (2014) 121301, <https://doi.org/10.1063/1.4902536>.
- [26] F.E. Kalf, M.P. Rebergen, E. Fahrenfort, J. Girovsky, R. Toskovic, J.L. Lado, J. Fernández-Rossier, A.F. Otte, *Nat. Nanotechnol.* 11 (11) (2016) 926–929, <https://doi.org/10.1038/nnano.2016.131>.
- [27] M. Möller, S.P. Jarvis, L. Guérinet, P. Sharp, R. Woolley, P. Rahe, P. Moriarty, *Nanotechnology* 28 (7) (2017) 075302, <https://doi.org/10.1088/1361-6528/28/7/075302>.
- [28] M. Ziatdinov, A. Maksov, S.V. Kalinin, *npj Comput. Mater.* 3 (1) (2017) 31, <https://doi.org/10.1038/s41524-017-0038-7>.
- [29] M.R. Slot, T.S. Gardenier, P.H. Jacobse, G.C.P. van Miert, S.N. Kempkes, S.J.M. Zevenhuizen, C.M. Smith, D. Vanmaekelbergh, I. Swart, *Nat. Phys.* 13 (7) (2017) 672–676, <https://doi.org/10.1038/nphys4105>.
- [30] T. Huff, H. Labidi, M. Rashidi, L. Livadaru, T. Diemel, R. Achal, W. Vine, J. Pitters, R.A. Wolkow, *Nat. Electron.* 1 (12) (2018) 636–643, <https://doi.org/10.1038/s41928-018-0180-3>.
- [31] R. Achal, M. Rashidi, J. Croshaw, D. Churchill, M. Taucer, T. Huff, M. Cloutier, J. Pitters, R.A. Wolkow, *Nat. Commun.* 9 (1) (2018) 2778, <https://doi.org/10.1038/s41467-018-05171-y>.
- [32] O. Gordon, P. D'Hondt, L. Knijff, S.E. Freeney, F. Junqueira, P. Moriarty, I. Swart, *Rev. Sci. Instrum.* 90 (10) (2019) 103704, <https://doi.org/10.1063/1.5099590>.
- [33] W. Ko, C. Ma, G.D. Nguyen, M. Kolmer, A.-P. Li, *Adv. Funct. Mater.* 29 (52) (2019) 1903770, <https://doi.org/10.1002/adfm.201903770>.
- [34] O.M. Gordon, P.J. Moriarty, *Mach. Learn.: Sci. Technol.* 1 (2) (2020) 023001, <https://doi.org/10.1088/2632-2153/ab7d2f>.
- [35] M. Rashidi, J. Croshaw, K. Mastel, M. Tamura, H. Hosseinzadeh, R.A. Wolkow, *Mach. Learn.: Sci. Technol.* 1 (2) (2020) 025001, <https://doi.org/10.1088/2632-2153/ab6d5e>.
- [36] A. Krull, P. Hirsch, C. Rother, A. Schiffrin, C. Krull, *Artificial-intelligence-driven scanning probe microscopy*, *Commun. Phys.* 3 (1) (2020) 54, <https://doi.org/10.1038/s42005-020-0317-3>.
- [37] S.V. Kalinin, M.A. Ziatdinov, J. Hinkle, S. Jesse, A. Ghosh, K.P. Kelley, A.R. Lupini, B.G. Sumpter, R.K. Vasudevan, *Automated and autonomous experiment in electron and scanning probe microscopy*, *ACS Nano* 15 (8) (2021) 12604, <https://doi.org/10.1021/acsnano.1c02104>.
- [38] J. Sotres, H. Boyd, J.F. Gonzalez-Martinez, *Nanoscale* 13 (2021) 9193–9203, <https://doi.org/10.1039/D1NR01109J>.
- [39] S. Wang, J. Zhu, R. Blackwell, F.R. Fischer, *J. Phys. Chem. A* 125 (6) (2021) 1384–1390, <https://doi.org/10.1021/acs.jpca.0c10731>.
- [40] Y. LeCun, Y. Bengio, G. Hinton, *Nature* 521 (7553) (2015) 436–444, <https://doi.org/10.1038/nature14539>.
- [41] J. Welker, F.J. Giessibl, *Science* 336 (6080) (2012) 444–449, <https://doi.org/10.1126/science.1219850>.
- [42] J. Welker, A.J. Weymouth, F.J. Giessibl, *ACS Nano* 7 (8) (2013) 7377–7382, <https://doi.org/10.1021/nn403106v>.
- [43] M. Emmrich, F. Huber, F. Pielmeier, J. Welker, T. Hofmann, M. Schneiderbauer, D. Meuer, S. Polesya, S. Mankovsky, D. Ködderitzsch, H. Ebert, F.J. Giessibl, *Science* 348 (6232) (2015) 308–311, <https://doi.org/10.1126/science.aaa5329>.
- [44] M.J. Vasile, D.A. Grigg, J.E. Griffith, E.A. Fitzgerald, P.E. Russell, *Rev. Sci. Instrum.* 62 (9) (1991) 2167–2171, <https://doi.org/10.1063/1.1142334>.
- [45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, *TensorFlow: large-scale machine learning on heterogeneous systems*, software available from <https://www.tensorflow.org/>, 2015.
- [46] Model implementation and trained weights are available at <https://github.com/SINGROUP/Auto-CO>, 2021.
- [47] D. Kingma, J. Ba Adam, *A method for stochastic optimization*, *arXiv preprint arXiv:1412.6980*.
- [48] F. Chollet, et al., *Keras*, <https://keras.io>, 2015.
- [49] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *J. Mach. Learn. Res.* 15 (1) (2014) 1929–1958, <http://jmlr.org/papers/v15/srivastava14a.html>.
- [50] H. Bay, T. Tuytelaars, L. Van Gool Surf, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), *Computer Vision – ECCV 2006*, in: *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 404–417.