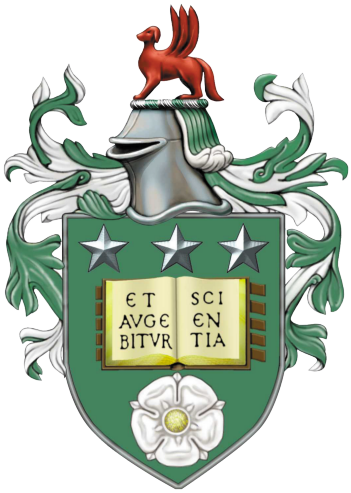
Automating Scanning Tunnelling Microscopy: A Comparative Study of Machine Learning and Deterministic Methods



Dylan Stewart Barker

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy of Physics

To the

University of Leeds

School of Physics and Astronomy

Supervisors

Dr. Adam Sweetman & Dr. Simon Connell

Leeds, September 2024

Acknowledgements

I have enjoyed the last few years immensely, and I owe that to a number of hugely important people in my life. Firstly, I'd like to thank my supervisor Adam for giving me the opportunity to work on this project, and for his support, guidance, and seemingly limitless patience for my general lab clumsiness.

A special thanks goes to Phil, for his incredible knowledge, and for the time and effort he put into supporting me throughout the entirety of my project (before leaving to get a "real job"). His insights and ideas into my project, teaching me the ways of UHV, and the generous sharing of experience and encouragement were invaluable. My PhD simply wouldn't have been possible, or nearly as enjoyable, without his support.

There are too many members of MNP who deserve a mention, but in particular (and in no special order): Cal, Anj, Vicky, Linda, Varun, Matt, Liza and Kalila - thank you for being such great and supportive friends to work with. For Chris, Connor, Mum, Dad, Nanny, Grandpa, Mike, Tom and all other family and friends past and present too great in number to mention - but Bunnie, Lucy and Iz *especially* - I truly cannot put into words how much I appreciate you. You've helped me become who I am today, so you can only blame yourselves really.

And finally, I would like to thank you for taking the time to read this.

Abstract

Scanning probe microscopy (SPM) methods allow for atomic scale investigation of material surfaces in real space and provide the potential to construct atomically precise structures atom-by-atom. Although these techniques have been available for decades, full automation of the processes involved in these experiments has not yet been fully realised. Currently, the process of setting up, running, and maintaining the SPM remains a laborious, time-consuming process, often as a result of the constantly changing shape of the probe tip.

Manual identification and correction of the state of the probe tip *in situ* requires a human operator to compare the probe quality via manual inspection of topographical images after any change in the probe. Previous attempts to automate the classification of the scanning probe state have predominantly relied on machine learning (ML) techniques. However, training these models demands large, labelled datasets for each surface under study. These datasets are extremely time-consuming to create, and are not always available, especially when considering a new substrate or adsorbate system.

This thesis focuses on automating the classification of the probe tip in scanning tunnelling microscopy (STM), addressing both imaging and spectroscopic applications. The use of ML in this automation is compared to less data-intensive, deterministic techniques, exploring the broader need for ML in autonomous scripting. We find that using these deterministic methods, comparable results in classifications can be achieved to those obtained with the use of ML. ML models were trained when possible to demonstrate the efficacy of the deterministic methods, via direct comparisons between the two classification techniques. The applicability of deterministic approaches is further validated through the utilisation of these classifiers in

automated experiments on various substrate systems. In addition to this, a scale-invariant method for surface mapping using Fourier space analysis is presented, which could aid in further automated experimentation.

Contents

1 Introduction

1.1 Motivation

Atomic resolution scanning probe microscopy (SPM) has revolutionised our ability to investigate nanoscale phenomena [1–3] and manipulate matter with exceptional precision [4–8]. In recent years, atomic manipulation has been extended to materials including superconductors [9, 10], semiconductors [11, 12], and 2D materials [13–15] in order to create and observe effects not found in naturally occurring materials. This level of manipulation could lead to progress in the miniaturisation of computational devices, including logic gates [6, 16, 17], memory storage [18, 19], transistors [7], wires [20], and even Boltzmann machines [21]. Additionally, the ability to functionalise the tip (pick up an atom or molecule using the tip) can improve tip quality for specific imaging and spectroscopy experiments [3, 22–27].

At present however, the fabrication of these structures is largely limited to manual work, carried out by an experienced SPM operator. Creating these structures by hand is an extremely time-consuming undertaking, involving the individual movement of atoms and molecules into the desired positions, whilst being constantly vigilant of changes to the probe tip, loss of contact with the surface, or other occurrences which could interrupt, or even restart the process. Overcoming this need for a human operator is the main hurdle in opening up the large scale fabrication of these systems, and as such, automating the procedures and methods involved in general SPM experimentation is a logical next step to this goal.

One of the key tools available to a probe microscopist to achieve this is the scanning tunnelling microscope (STM), which relies heavily on the sharpness of the probe tip both for imaging and manipulation. Prior to automating STM as a whole, the ability to both prepare, and maintain a

probe tip automatically needs to be realised. Work has been carried out in this area previously (as will be discussed in detail later) with some success [28–30], however attempts thus far have primarily utilised only machine learning (ML) methods. Whilst the use of ML is not explicitly a problem, the large, balanced datasets required for training are not common place in nanoscience as a whole [31]. As a result, a large amount of time and effort is required, firstly to create the dataset itself, followed by the labelling and analysis of the contents of the dataset to ensure it is representative of all use cases. Moreover, once trained, it is difficult to decipher exactly what a model has learned about a dataset in order to make classifications; this lack in transparency can cause problems in identifying biases, and causes challenges in troubleshooting and improving models.

It is currently common in SPM research to use ML to solve these automation problems [28–30, 32–34], without considering either the drawbacks of the methods, or the other options available. In the case of SPM images, where datasets of a specific surface have a common appearance, well established image analysis techniques could be sufficient to solve the problems being tackled by ML. In this thesis we will explore ways of automating SPM processes with and without the use of ML.

1.2 Thesis Outline

This thesis will cover various areas where the automation of SPM techniques could alleviate the challenges faced by researchers, with a focus on how it could free up a researchers time in menial experimental preparation tasks. Specifically, we will discuss the automation of the classification and preparation of scanning probe tips in STM, with a focus on discussing the alternatives to using ML for this process.

We begin in Chapter 2, where we first discuss the history and key physics concepts behind the experimental instruments used for data collection, particularly the STM. Following this, we will continue our theory discussions with an introduction to ML methods, concentrating on supervised ML methods such as those used throughout this thesis. A review of the

current literature on automated classification attempts made in SPM, which predominantly employs ML based models, will also be presented.

From here, Chapter 3 will discuss the experimental methods used in this project, including ultra-high vacuum (UHV) techniques, sample and probe preparation, and the software used throughout. Following this we will briefly discuss the computational methods which made this project possible.

Moving onto the first section of results in Chapter 4, an image analysis method is introduced which utilises Fourier filtering to automatically locate the atomic positions, included those which are obscured by adsorbates and defects, on the Si(111) - 7×7 and B:Si(111) - $(\sqrt{3} \times \sqrt{3})R30^{\circ}$ substrates at room temperature. Contaminants and defects are commonly observed on these substrates, despite this, mapping the underlying structure could facilitate the automation STM manipulation. This is done by taking the Fast Fourier Transform (FFT) of an input image, and isolating the frequency peaks in Fourier space which correspond to the repeating structure of the surface. This method is able to accurately locate the atomic positions, whilst being scale invariant, and requiring minimal knowledge of the surface itself.

In Chapter 5, the theoretical discussions from Chapters 2 and 3 are applied to the automated classification of the scanning probe tip state in STM based on imaging. We compare two classification methods: the commonly used ML-based approach and a novel deterministic classifier. An analysis of their efficacy is presented, demonstrating comparable accuracy and precision. The robustness of these methods are showcased by their application to a various systems of differing topographical appearances: Si(111) - 7×7, B:Si(111) - $(\sqrt{3} \times \sqrt{3})R30^{\circ}$, and Cu(111) in two configurations - one with a deposition of Cu adatoms and CO molecules, and another with the addition of C_{60} molecules. Previous attempts from the literature to solve similar problems using ML are discussed, and where possible, our deterministic technique is applied to these open-source datasets. We also introduce a final automated tip preparation tool, utilising the deterministic classification scheme, highlighting its ease of use and application to new systems.

Finally, we conclude in Chapter 6 with an additional method of tip state

classification, this time based on the I(V) spectroscopy taken on a bare metal substrate, Au(111). This part of the project integrates methods from Chapter 5 and introduces new techniques to demonstrate an automated scanning tunnelling spectroscopy (STS) experiment on Au(111) with a low coverage of Tin Phthalocyanine (SnPc). We then conclude with a summary of our findings and their implications for the future of automated SPM in Chapter 7.

References

(1) Binnig, G.; Rohrer, H.; Gerber, C.; Weibel, E. *Physical Review Letters* **1983**, *50*, 120–123.

(2) Sugimoto, Y.; Pou, P.; Abe, M.; Jelinek, P.; Pérez, R.; Morita, S.; Custance, Ó. *Nature 2006 446:7131* **2007**, *446*, 64–67.

(3) Gross, L.; Mohn, F.; Moll, N.; Liljeroth, P.; Meyer, G. *Science* **2009**, *325*, 1110–1114.

(4) Stroscio, J. A.; Celotta, R. J. *Science* **2004**, *306*, 242–247.

(5) Ternes, M.; Lutz, C. P.; Hirjibehedin, C. F.; Giessibl, F. J.; Heinrich, A. J. *Science* **2008**, *319*, 1066–1069.

(6) Kolmer, M.; Godlewski, S.; Lis, J.; Such, B.; Kantorovich, L.; Szymonski, M. *Microelectronic Engineering* **2013**, *109*, 262–265.

(7) Fuechsle, M.; Miwa, J. A.; Mahapatra, S.; Ryu, H.; Lee, S.; Warschkow, O.; Hollenberg, L. C.; Klimeck, G.; Simmons, M. Y. *Nature Nanotechnology* **2012**, *7*, 242–246.

(8) Leinen, P.; Esders, M.; Schütt, K. T.; Wagner, C.; Müller, K. R.; Stefan Tautz, F. *Science Advances* **2020**, *6*.

(9) Kim, H.; Palacio-Morales, A.; Posske, T.; Rózsa, L.; Palotás, K.; Szunyogh, L.; Thorwart, M.; Wiesendanger, R. *Science Advances* **2018**, *4*.

(10) Liebhaber, E.; Rütten, L. M.; Reecht, G.; Steiner, J. F.; Rohlf, S.; Rossnagel, K.; von Oppen, F.; Franke, K. J. *Nature Communications* **2021**, *13*.

(11) Schofield, S. R.; Studer, P.; Hirjibehedin, C. F.; Curson, N. J.; Aeppli, G.; Bowler, D. R. *Nature Communications 2013 4:1* **2013**, *4*, 1–7.

(12) Fölsch, S.; Yang, J.; Nacci, C.; Kanisawa, K. *Physical Review Letters* **2009**, *103*, 096104.

(13) Wyrick, J.; Natterer, F. D.; Zhao, Y.; Watanabe, K.; Taniguchi, T.; Cullen, W. G.; Zhitenev, N. B.; Stroscio, J. A. *ACS Nano* **2016**, *10*, 10698–10705.

(14) Cortés-del Río, E.; Mallet, P.; González-Herrero, H.; Lado, J. L.; Fernández-Rossier, J.; Gómez-Rodríguez, J. M.; Veuillen, J. Y.; Brihuega, I. *Advanced Materials* **2020**, *32*, 2001119.

(15) González-Herrero, H.; Gómez-Rodríguez, J. M.; Mallet, P.; Moaied, M.; Palacios, J. J.; Salgado, C.; Ugeda, M. M.; Veuillen, J. Y.; Yndurain, F.; Brihuega, I. *Science* **2020**, *352*, 437–441.

(16) Huff, T.; Labidi, H.; Rashidi, M.; Achal, R.; Livadaru, L.; Dienel, T.; Pitters, J.; Wolkow, R. A. *Nature Electronics* **2017**, *1*, 636–643.

(17) Broome, M. A.; Gorman, S. K.; House, M. G.; Hile, S. J.; Keizer, J. G.; Keith, D.; Hill, C. D.; Watson, T. F.; Baker, W. J.; Hollenberg, L. C.; Simmons, M. Y. *Nature Communications 2018 9:1* **2018**, *9*, 1–7.

(18) Achal, R.; Rashidi, M.; Croshaw, J.; Churchill, D.; Taucer, M.; Huff, T.; Cloutier, M.; Pitters, J.; Wolkow, R. A. *Nature Communications 2018 9:1* **2018**, *9*, 1–8.

(19) Kalff, F. E.; Rebergen, M. P.; Fahrenfort, E.; Girovsky, J.; Toskovic, R.; Lado, J. L.; Fernández-Rossier, J.; Otte, A. F. *Nature Nanotechnology* **2016**, *11*, 926–929.

(20) Weber, B.; Mahapatra, S.; Ryu, H.; Lee, S.; Fuhrer, A.; Reusch, T. C.; Thompson, D. L.; Lee, W. C.; Klimeck, G.; Hollenberg, L. C.; Simmons, M. Y. *Science* **2012**, *335*, 64–67.

(21) Kiraly, B.; Knol, E. J.; van Weerdenburg, W. M.; Kappen, H. J.; Khajetoorians, A. A. *Nature nanotechnology* **2021**, *16*, 414–420.

(22) Bartels, L.; Meyer, G.; Rieder, K. H. *Applied Physics Letters* **1997**, *71*, 213–215.

(23) Temirov, R.; Soubatch, S.; Neucheva, O.; Lassise, A. C.; Tautz, F. S. *New Journal of Physics* **2008**, *10*, 053012.

(24) Kichin, G.; Weiss, C.; Wagner, C.; Tautz, F. S.; Temirov, R. *Journal of the American Chemical Society* **2011**, *133*, 16847–16851.

(25) Mohn, F.; Schuler, B.; Gross, L.; Meyer, G. *Applied Physics Letters* **2013**, *102*, 73109.

(26) Sweetman, A. M.; Jarvis, S. P.; Sang, H.; Lekkas, I.; Rahe, P.; Wang, Y.; Wang, J.; Champness, N. R.; Kantorovich, L.; Moriarty, P. *Nature Communications 2014 5:1* **2014**, *5*, 1–7.

(27) Mönig, H.; Hermoso, D. R.; Arado, O. D.; Todorović, M.; Timmer, A.; Schüer, S.; Langewisch, G.; Pérez, R.; Fuchs, H. *ACS Nano* **2016**, *10*, 1201–1209.

(28) Rashidi, M.; Wolkow, R. A. *ACS Nano* **2018**, *12*, 5185–5189.

(29) Gordon, O.; D'Hondt, P.; Knijff, L.; Freeney, S.; Junqueira, F.; Moriarty, P.; Swart, I. *Review of Scientific Instruments* **2019**, *90*, 103704.

(30) Krull, A.; Hirsch, P.; Rother, C.; Schiffrin, A.; Krull, C. *Communications Physics* **2020**, *3*, 1–8.

(31) Barnard, A. S.; Motevalli, B.; Parker, A. J.; Fischer, J. M.; Feigl, C. A.; Opletal, G. *Nanoscale* **2019**, *11*, 19190–19201.

(32) Ziatdinov, M.; Maksov, A.; Kalinin, S. V. *npj Computational Materials* **2017**, *3*, 31.

(33) Wang, S.; Zhu, J.; Blackwell, R.; Fischer, F. R. *Journal of Physical Chemistry A* **2021**, *125*, 1384–1390.

(34) Alldritt, B.; Urtev, F.; Oinonen, N.; Aapro, M.; Kannala, J.; Liljeroth, P.; Foster, A. S. *Computer Physics Communications* **2022**, *273*, 108258.

2 Theory and Literature Review

2.1 Scanning Tunnelling Microscopy

The main principle behind scanning tunnelling microscopy (STM) is the measurement of a current produced by electrons transferring between the scanning probe tip and the sample with a separation of distance d. This separation causes a potential barrier, which classically, inhibits the flow of electrons unless their energy is sufficient to overcome the barrier itself. However, quantum mechanics allows for electrons to traverse the gap, without sufficient energy to do so classically, with a finite probability, in a process known as quantum tunnelling. This phenomenon was first observed by Giaever [1] when measuring a current between two metals separated by a thin insulating film whilst applying a potential difference between them. Here, rather than an insulating film, the barrier results (usually) from the vacuum gap between the tip and sample. This principle was later developed into the first scanning tunnelling microscope by Binnig and Rohrer in 1981 [2] for which they received the Nobel Prize in Physics in 1986.

2.1.1 Quantum Tunnelling

The feedback signal used in STM to control the distance between the tip and the sample is the tunnel current, I. In quantum mechanics, characteristics of a particle are described by it's wavefunction, $\psi(z)$, from which one is able to calculate the probability of a particle to be at any given position:

$$P(z) = |\psi(z)|^2 \tag{2.1}$$

It is possible to calculate the dependence of the current on the tip-sample separation by first considering the time-independent Schrödinger equation in one dimension:

$$-\frac{\hbar^2}{2m}\frac{d^2}{dz^2}\psi(z) + U(z)\psi(z) = E\psi(z) \tag{2.2}$$

where E is the energy of the electron, U is the height of the barrier and m is it's mass. An electron on the allowed side of the barrier has the solution:

$$\psi(z) = \psi(0)e^{\pm ikz} \tag{2.3}$$

where $k = \frac{\sqrt{2m(E-U)}}{\hbar}$ is the wave-vector. When the electron is within the potential barrier, the solution to Equation 2.2 is:

$$\psi(z) = \psi(0)e^{-\alpha z} \text{ in potential barrier} \tag{2.4}$$

where $\alpha = \frac{\sqrt{2m(U-E)}}{\hbar}$ is the decay constant of the electron whilst inside the potential barrier region. This shows that while the separation, z, is finite, there is a non-zero probability of finding an electron within, and hence, beyond the potential barrier:

$$P(z) = |\psi(0)^2 e^{-2\alpha z}| \tag{2.5}$$

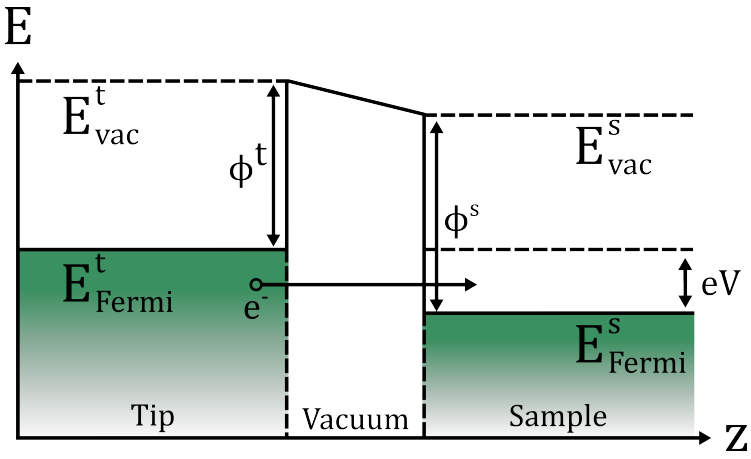


Figure 2.1: Energy diagram of the tip/vacuum/sample system where E^t_{Fermi} and E^s_{Fermi} are the Fermi energies, E^t_{vac} and E^s_{vac} are the vacuum levels and ϕ^t and ϕ^s are the work functions for the tip and the sample respectively.

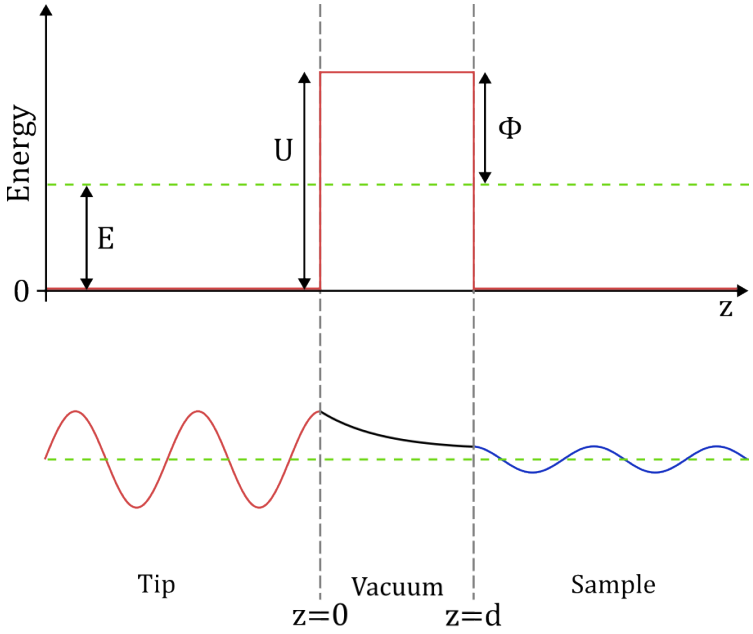


Figure 2.2: One dimensional representation of the tip-vacuum-sample junction. The upper section shows the potential barrier of height U, which given that the energy of the electrons, E, is less than U, is a classically forbidden region. The lower sections shows the resultant decay of the wavefunction within the vacuum region.

Hence, it is possible that an electron is able to tunnel through a potential barrier, caused in STM by the separation of the probe tip and sample. The one-dimensional potential barrier system (Figure 2.1) is a good basic model to describe the principles of STM. Here, the work function, ϕ, is the minimum energy needed for an electron to transfer from the highest occupied state in a material into the vacuum without tunnelling. The magnitude of this work function varies depending on the sample and orientation.

This electron tunnelling is equally bi-directional in the absence of an applied bias to either the tip or sample. A net current can be generated by biasing either the sample or tip, which alters the Fermi energies of the materials (E_F^s and E_F^t respectively) by an amount eV, dependent on the bias voltage applied, V. By convention in STM, the sample is usually biased and the tip grounded. A positive bias being applied to the sample has the effect of lowering E_F^s (due to $E = -qV$), resulting in preferential tunnelling of electrons from occupied states in the tip into unoccupied states (of energies ranging between E_F^s and E_F^t) in the sample. With a negative bias applied to the surface, tunnelling occurs preferentially from the sample to the tip.

2.1.1.1 STM Theory

Imaging contrast in STM is a result of not only the topography, but the electronic structure of the sample, causing the interpretation of STM image to be somewhat complicated; rather than an image representing solely the topography of the surface, it is more accurate to describe it as a contour map of the local density of states (LDOS) of a surface. Through a series of approximations, it is possible to derive some of the key characteristics of STM. With a simplified model of a one-dimensional metal-insulator-metal junction, described by a rectangular potential as shown in Figure 2.1, the tip-vacuum-sample junction can be described. The characteristics of a tunnelling electron are described by the wavefunction, $\psi(z)$, with z, the displacement of the tip, and assuming a vacuum width of d, with potential U_0, the potential across the junction is dependent on the position of an electron at any point in z:

$$U(z) = \begin{cases} 0, & z < 0 \\ U_0, & 0 < z < d \\ 0, & z > d \end{cases} \tag{2.6}$$

The transmission probability can be calculated using the one-dimensional time-independent Schrödinger equation from Equation 2.2. For an electron of mass m and energy E, a solution to this is that of a plane wave with a wavevector, k, travelling from the left of the junction:

$$\psi = e^{ikz} \tag{2.7}$$

Therefore using Equation 2.2:

$$k = \sqrt{\frac{2m}{\hbar^2}[E - U(z)]} \tag{2.8}$$

Outside of the vacuum potential barrier, the potential is zero and so:

$$k = \sqrt{\frac{2m}{\hbar^2}E} \tag{2.9}$$

Inside the vacuum potential, $U = U_0$, and $E - U_0 < 0$. Therefore, k from equation 2.8 is imaginary. By defining α such that $k = i\alpha$:

$$\alpha = \sqrt{\frac{2m}{\hbar^2}[U_0 - E]} \tag{2.10}$$

In this case, the wavevector inside the barrier region (Equation 2.10) is complex, and hence the wavefunction within the barrier is a real, exponentially decaying oscillating wave:

$$\psi = e^{-\alpha z} \tag{2.11}$$

This is shown pictorially in figure 2.2. There is an additional solution to the right-travelling wave resulting from reflections of the wavefunctions on the barrier edges which results in a left-travelling wave, $\psi = e^{-ikz}$. Both solutions have the same form inside and outside of the potential barrier, therefore the final solution is a superposition of these two waves. The resulting wavefunction, ψ, can be described by the following equations:

$$\psi(z) = \begin{cases} Ae^{ikz} + Be^{-ikz}, & z < 0 \\ Ce^{-\alpha z} + De^{\alpha z}, & 0 < z < d \\ Fe^{ikz} + Ge^{-ikz}, & z > d \end{cases} \tag{2.12}$$

where A, B, C, D, F and G are all constants.

The transmission coefficient, T, can then be calculated based on the ratio of coefficients A and F as $T = \mid F \mid^2 / \mid A \mid^2$:

Firstly, in this model there is no barrier to the right of the rightmost region resulting in no reflection of the wave for $z > a$, therefore $G = 0$. We can now exploit the fact that both $\psi(z)$ and $\frac{d\psi(z)}{dz}$ are continuous at the boundaries of the potential barrier. We therefore find that at $z = 0$:

$$Ae^{ik\cdot 0} + Be^{-ik\cdot 0} = Ce^{\alpha\cdot 0} + De^{-\alpha\cdot 0}, \tag{2.13}$$

$$\therefore A + B = C + D \tag{2.14}$$

and

$$\frac{d}{dz}[Ae^{ik\cdot 0} + Be^{-ik\cdot 0}] = \frac{d}{dz}[Ce^{\alpha\cdot 0} + De^{-\alpha\cdot 0}], \tag{2.15}$$

$$\therefore ik[A - B] = \alpha[C - D] \tag{2.16}$$

And repeating for for $z = d$:

$$Ce^{\alpha\cdot d} + De^{-\alpha\cdot d} = Fe^{ik\cdot d} \tag{2.17}$$

$$\alpha[Ce^{\alpha\cdot d} + De^{-\alpha\cdot d}] = ikFe^{ik\cdot d} \tag{2.18}$$

We can now solve Equations 2.14, 2.16, 2.17 and 2.18 for $T = \mid F \mid^2 / \mid A \mid^2$ as simultaneous equations starting by multiplying Equation 2.14 by ik and equating it to Equation 2.16:

$$2ikA = C(ik + \alpha) + D(ik - \alpha) \tag{2.19}$$

Taking the assumption that the potential barrier is sufficiently large ($\alpha z \gg 1$), we can infer than $e^{-\alpha d} \approx 0$. Therefore by multiplying Equation 2.17 by $e^{-\alpha d}$ we find:

$$\begin{gathered} e^{-\alpha d}[Ce^{\alpha d} + De^{-\alpha d}] = Fe^{ikd}e^{-\alpha d} \\ C + D(e^{-\alpha d})^2 = Fe^{ikd}e^{-\alpha d} \\ \therefore C \approx 0 \end{gathered} \tag{2.20}$$

Which simplifies Equation 2.19:

$$2ikA \approx D(ik - \alpha) \tag{2.21}$$

Next we combine Equations 2.17 and 2.18 by equating around $Ce^{\alpha d}$:

$$\begin{gathered} \frac{ik}{\alpha} Fe^{ikd} + 2De^{-\alpha d} = Fe^{ikd} \\ \boxed{2D\alpha e^{-\alpha d} = Fe^{ikd}(\alpha - ik)} \end{gathered} \tag{2.22}$$

Combining Equations 2.21 and 2.22 we can eventually obtain an expression relating F and A:

$$\begin{gathered} ikA = \frac{Fe^{ikd}e^{\alpha d}}{2\alpha}(\alpha - ik) \\ \boxed{\frac{4ik\alpha}{ik - \alpha} Ae^{-\alpha d} \approx Fe^{ikd}(\alpha - ik)} \end{gathered} \tag{2.23}$$

Finally, the transmission coefficient, T can be determined from Equation 2.23:

$$\boxed{T = \frac{|F|^2}{|A|^2} \approx \frac{16k^2\alpha^2}{(k^2 + \alpha^2)^2} e^{-2\alpha z}} \tag{2.24}$$

The one-dimensional derivation above is sufficient to explain the exponential dependence of the tunnelling current on the distance between the tip and the sample, however it does not capture the entirety of the interaction. For this we need to consider the three-dimensional description as shown by Tersoff and Hamann [3]. The tunnel current for two metals in a vacuum can be described as:

$$I = \frac{2\pi e}{\hbar} \sum_{t,s} f\left(E_t\right) \left[1 - f\left(E_s + eV\right)\right] \left|M_{ts}\right|^2 \delta\left(E_t - E_s\right) \tag{2.25}$$

where $f(E)$ is the Fermi function, V is the applied voltage, M_{ts} is the tunnelling matrix element between states of the probe, ψ_t, and the surface, ψ_s, which describes the probability of the tip and sample wavefunctions overlapping. Additionally, the $f(E_t)$ and $[1 - f(E_s + eV)]$ terms ensure tunnelling between filled states in the tip and empty states in the biased sample respectively.

Taking the assumption that we are working at low temperature and only a small bias is applied between the tip and sample, Tersoff and Hamann [3] find that Equation 2.25 simplifies to:

$$I = \frac{2\pi e^2 V}{\hbar} \sum_{t,s} \left|M_{ts}\right|^2 \delta\left(E_t - E_F\right) \delta\left(E_s - E_F\right) \tag{2.26}$$

Considering the limit where the tip is modelled as a point probe at position r_0, this can be further simplified to:

$$I \propto \sum_{s} \left|\psi_s\left(r_0\right)\right|^2 \delta\left(E_s - E_F\right). \tag{2.27}$$

This case represents the ideal tip, with maximum possible resolution. With this, the tip wave functions are arbitrarily localised and therefore the matrix element is simply proportional to the amplitude of ψ_t at the position r_0, thus it is found that the current, I, is directly proportional to the local density of states (LDOS) of the surface [3] and images represent a contour map of the surface LDOS. The only electron states which will contribute to the tunnelling will be states between E_F and $E_F + eV$ and so we can adapt this accordingly:

$$I \propto \int_{E_F}^{E_F+eV} |\psi(\mathbf{r}_0)|^2 \delta(E - E_F) T dE. \tag{2.28}$$

It is worth noting that the assumption that the tip apex is a perfect point is poor, however it is difficult to analytically model the effect of the tip shape [4] on the surface interaction.

Using Equations 2.11 and 2.28, we can see that the tunnel current decays exponentially with the barrier height. With general work functions on the order of a few electron volts, it can be shown that every 1Å change in the tip-sample separation results in a change in the tunnel current by approximately an order of magnitude. This strong dependence of the current on the tip-sample separation results in the extremely high spatial resolution of STM. Even with a tip which is macroscopically blunt, all that is needed for atomic resolution imaging to occur is a small cluster to be protruding from the bulk tip.

2.1.2 Taking a Scan

At this point the nature of the measured tunnelling current in STM is well understood, and so we can discuss how this measurement is exploited to probe a surface.

STM can be used to map the topography of the LDOS of the surface, or take spectra over a fixed position, known as scanning tunnelling spectroscopy (STS). To obtain a two-dimensional image of the surface, the tip is rastered across the surface, building up a representation of the surface line-by-line. The current (or tip height depending on imaging mode) at each point can then be mapped to the pixel intensity producing a false-colour image of our sample. This process can be seen in Figure 2.3.

Due to the extremely small scale of the features being measured in STM, movements of the probe tip must be accurate to less than one tenth of an Ångstrom (10 pm). Because of this, piezoelectric actuators are used for all tip movements. With this precision, we are able to probe the properties of

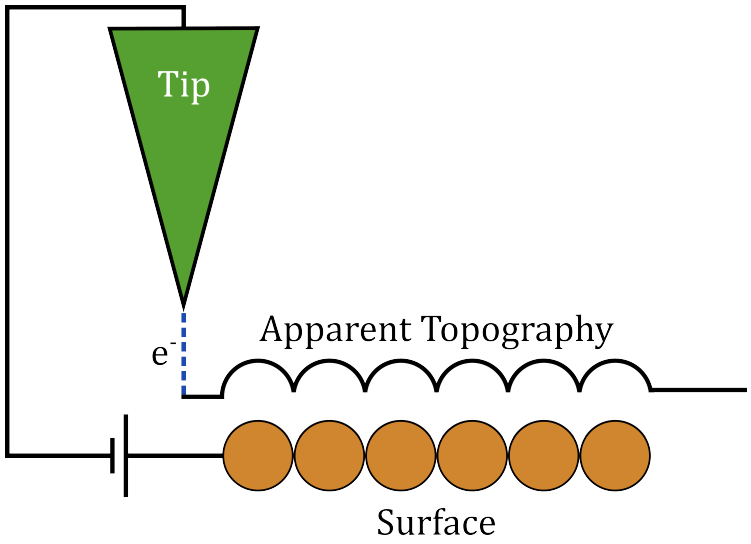


Figure 2.3: Single line of a constant current topograph in STM. The current is maintained at a fixed value by varying the tip-sample distance. This is repeated over a series of lines to obtain a 2d topographical image.

specific sites on a surface, manipulate atoms and molecules, and image at the atomic scale [5], producing images such as those in Figure 2.4.

STM images can be taken in one of two operational modes: constant-current or constant-height. The more common of the two is constant-current mode as is depicted in Figure 2.3, where a Proportional-Integral (PI) controller is used to vary the tip-sample distance in order to maintain a user defined tunnel current setpoint. The gain, P, and the integration time, I, parameters can be considered as the controller's magnitude, and reaction speed of the loop response respectively. These parameters need to be chosen carefully to maximise the signal-to-noise ratio for a given system and scan speed. If the values for P and I are too high, the loop may resonate, causing self excitation in the signal which appears as ringing, however if the parameters are set too low the controller may not be able to react in time to large changes in height (such as a step edge.) It is much easier for the PI controller to keep up with changes to the topography of the image with a lower scan speed. Image artefacts caused by the PI controller will be discussed in Section 2.1.2.3. Alternatively, in constant-height mode the image is taken whilst the feedback controller is disabled, with the tip at a set distance from

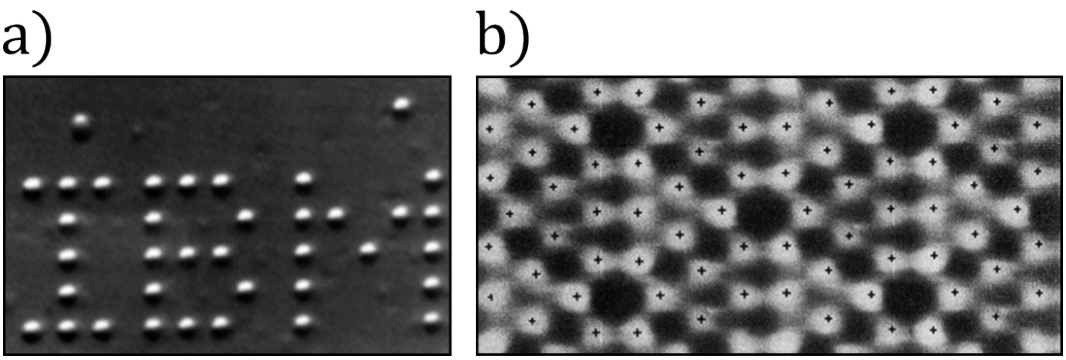


Figure 2.4: a) The IBM logo constructed from Xe atoms on the Ni(110) surface [5] ($T = 4$ K, $V = 0.01$ V. $I = 1$ nA), one of the earliest STM images published involving atomic manipulation, each letter is 5 nm tall. b) The first published STM image showing the Si(111) - 7×7 surface structure [6].

the surface and raster scanning the tip across the surface, measuring the current at each point. A constant-height scan can be carried out faster than in constant-current (as the scan speed is no longer limited by the feedback controller), and provides a map of the LDOS. Scanning in constant-height mode can increase the risk of the tip coming into contact with the surface, which will cause the STM data to be invalidated and most likely result in a change to the geometry of the probe itself.

Due to the nature of the samples investigated in this thesis, all work needs to be undertaken in ultra-high vacuum (UHV) conditions. In atmosphere, samples will be covered in multiple layers of oxygen, water and other contaminants which will negatively effect the appearance of the surface in STM.

2.1.2.1 Vibration Isolation

With the extreme precision needed for UHV STM, it is vital that the system is sufficiently isolated from the many forms of mechanical vibrations which would otherwise be visible in imaging. These vibrations (usually of the order of a few Hz), travel through the system and cause the tip-sample junction to oscillate, and are the dominant cause of noise in the tunnel current. Scanners

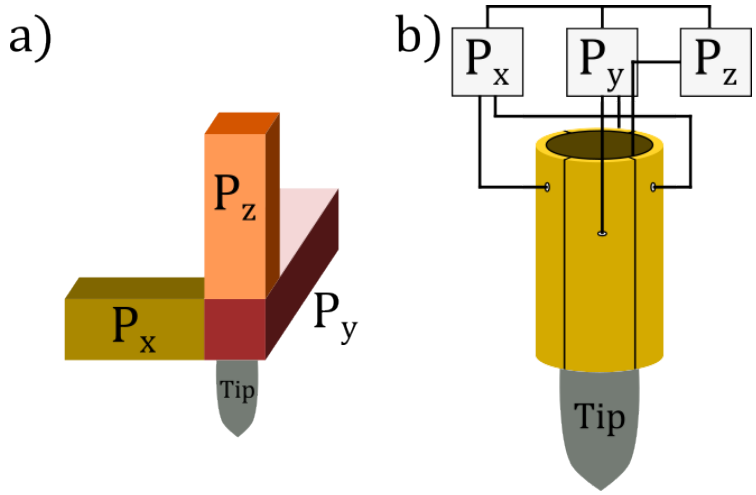


Figure 2.5: a) Original tripod piezoelectric scanner design. b) Tube scanner design [7].

are therefore set up in areas with little foot-traffic, on heavy tables with air-legs which decouple the system from the floor and nearby pumping systems. In addition to this, the STM scan head is suspended from springs with eddy current damping to further reduce the effect of mechanical vibrations.

2.1.2.2 Fine and Coarse Motor Control

As discussed previously, the high precision required for STM operation necessitates the use of piezoelectric actuators for fine control of the position of the tip. These actuators take advantage of the piezoelectric effect, where by applying a voltage across a piezoelectric material, the crystal undergoes a transverse extension or contraction depending on the polarity of the voltage applied. This movement of the actuator, once calibrated with respect to the applied voltage, can be utilised to perform controlled subnanometer movements. STM piezoelectric actuators are commonly made from lead zirconium titanate (PZT) ceramics [7].

Figure 2.5 shows two of the most well known piezodrive actuators which facilitate precise movement of the tip in three-dimensions. The earliest scanner design is a tripod of three orthogonal "sticks" made from piezoelectric material. The STM tip (or sample) is placed on the tripod and the movement

of the "sticks" allow the tip move along the x, y, z direction. The single tube scanner, depicted in Figure 2.5b, was first employed by Binnig and Smith [8]. Here, the outer electrode of a PZT tube is divided into four equal quadrants, with the inner electrode kept whole. By applying voltages to opposite quadrants on the tube, the tube is able to bend perpendicular to the axis of the tube. Tube scanners exhibit a smaller level of cross-talk, and a higher resonance frequency when compared with the tripod scanner, and therefore contribute less to mechanical noise.

The z-piezo used for fine control of the STM tip has a finite range of expansion, and so is not able to make the large movements needed to macroscopically approach the tip to the sample. The system involved in these larger movements to reduce the tip-sample distance to within the working range of the z-piezo is known as the coarse approach, which needs to be able to both move the tip far enough away to allow sample transfer, and close enough to detect a tunnel current (< 1 nm). Multiple designs for coarse motion are available [2, 9], however the design used throughout the work done in this thesis uses slip-stick motion [10]. The operation of a slip-stick motor is shown in Figure 2.6, where a piezo stack is in contact with the object which is being moved. The piezo is extended slowly, Figure 2.6b, due to a slow increase in voltage, where the object moves together with the piezo due to the fictional force between the contact point and the object. Once at full piezo extension, it is quickly retracted by applying a rapidly decreasing voltage, Figure 2.6c. Due to the inertia of the object, the contact point slides without moving it. By repeating these two steps, macroscopic movements can be achieved.

2.1.2.3 Image Distortions

Due to the high precision required for STM, there are several sources of uncertainty in the true position of the tip compared to the topograph which need to be taken into account.

Piezoelectric actuator hysteresis is a somewhat less common distortion which manifests as a stretching of the features within an image in the fast-scan direction at the start of each line. This occurs as strain (displacement) in a piezoelectric material is, in part, a result of the reorientation of ferroelectric

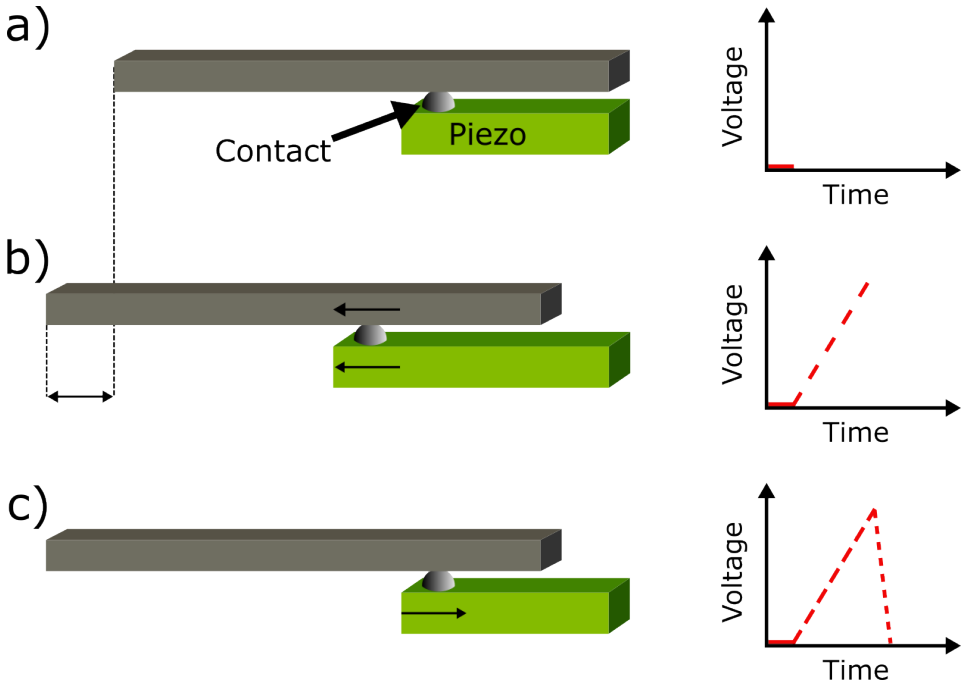


Figure 2.6: Working principle of a slip-stick motor used for macroscopic movements. a) The initial setup showing a piezo stack in green making contact with an object which will be moved. b) A gradual increase in voltage is applied to the piezo which extends it slowly, moving the object with it due to the friction at the point of contact. c) At full piezo extension, the piezo is quickly retracted by applying a rapid decrease in voltage, causing the piezo to move without carrying the object with it.

domains within the material [11]. This reorientation however, in response to an applied voltage, depends not only on the nature of the voltage applied to the material, but also on the internal state of the system (current orientation and history of the domains). This means that the displacement of the piezo is not linear with applied voltage, and can be offset from the expected value by $10 - 25\%$ for different voltages [7]. The effect of this hysteresis is more prevalent at larger voltages (larger scan sizes).

Piezoelectric creep is another effect which is directly related to the use of piezoelectric materials. As explained previously, voltage applied to a piezoelectric actuator causes both strain and domain alignment; creep is caused by the slow relaxation of domains into an aligned state due to an applied voltage. The effect of this becomes evident when an image is taken shortly after a large movement of the piezos such as initial sample approach and when selecting a scan window. Creep appears as a shear of the features at the start of an image. The effect of creep can be reduced by moving the actuator at a lower speed for large movements, and waiting for a few minutes after these movements before starting a scan.

Thermal drift is a distortion in imaging due to a difference in the relative expansion/contraction of all components within the microscope, and so is not a problem specific to piezoelectric actuators. These differences in expansions alter the motion of the STM tip relative to the sample in the scan head resulting from thermal gradients across the instrument, as well as differing coefficients of thermal expansion between materials. This leads to parts of the microscope expanding at different rates as the temperature fluctuates. Thermal drift appears as a semi-consistent stretching or compression of the features within an image, caused by the scan frame drifting as the scan is taken, as can be seen in Figure 2.7. This drift can be compensated for by estimating the drift vector and compensating for it in the scan controller, however, due to the constant fluctuations (especially whilst working at room temperature), this may need to be repeated periodically. These effects can also be alleviated significantly by operating at cryogenic temperatures (~ 4.5 K), reducing thermal drift velocities by orders of magnitude [12].

If the PI parameters in the feedback controller are not adequately optimised for the scan speed, the apparent topography of the image can appear distorted when encountering large changes in the height. If the feedback

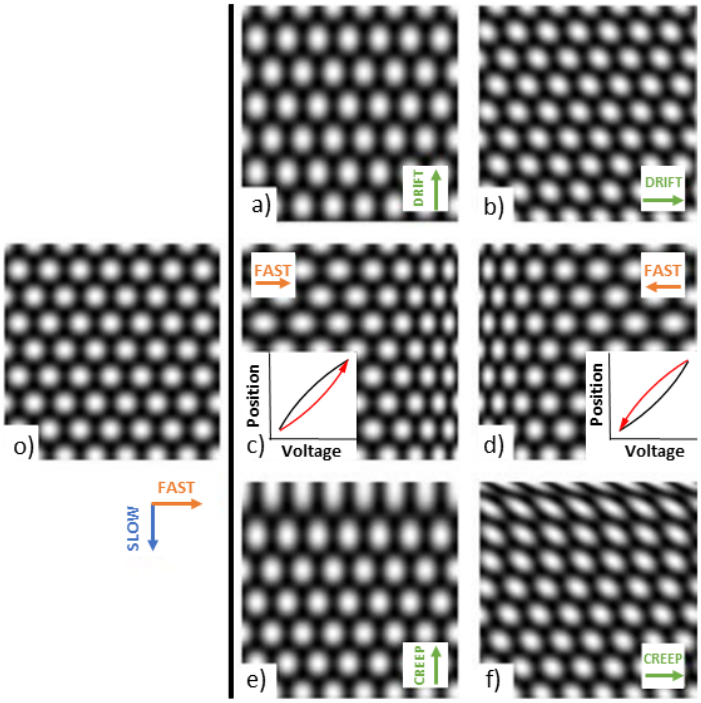


Figure 2.7: Example of distortion effects on a simulated STM image of graphite. o) The undistorted image. Thermal drift in the vertical direction a) and horizontal b). Hysteresis in the trace image c) and retrace image d) with insets showing the dependence of the position on history of the crystal. Piezo creep opposite to the vertical direction e) and horizontal f). Figure from Yothers *et al.* [13]

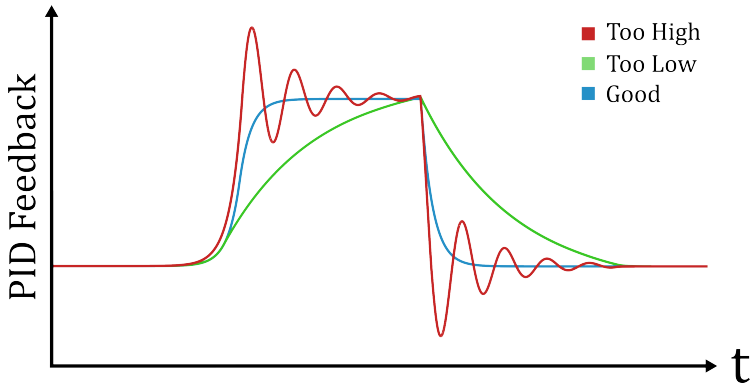


Figure 2.8: Feedback reactions to a signal in the shape of a top hat function with overcompensation in red, under compensation in green and a good level of compensation in blue.

settings are set such that reactions occur too fast, compensation may overshoot when encountering a sudden change in height such as a step or large adsorbate; subsequent reactions to the initial compensation may also overshoot and cause ringing to appear in the topograph. In the case that the feedback reaction is too slow, features will appear blurred. Figure 2.8 shows an example feedback reaction to a change in an imaging signal, with a correct reaction shown by the blue curve, too fast feedback settings in red, and too slow in green. Physical examples of these effects are shown in Figure 2.9

One final, common image artefact seen in scanning probe microscopy (SPM) occurs due to mechanical noise in the system. Noise with a high enough amplitude will show up as stripes superimposed over the topography of an surface. The frequency at which these stripes appear will change depending on the scan speed and the frequency of the noise itself. These effects can be reduced by correctly isolating the microscope from causes of such noise, as discussed in Section 2.1.2.1.

2.1.2.4 Effect of Probe Tips on Images

Due to the nature of STM operation, the shape of the probe tip plays a significant role in the interpretation of topographical images. Scans of

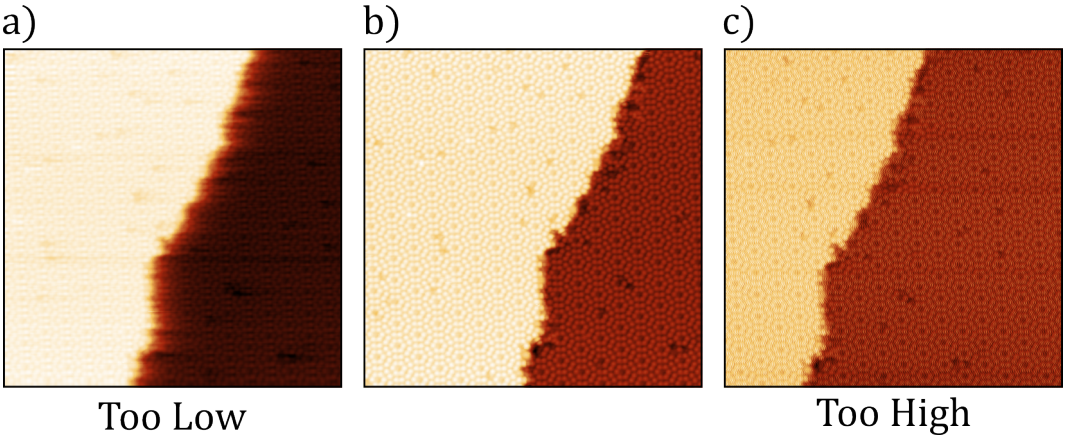


Figure 2.9: Room temperature STM scans of a step edge features on the Si(111) - 7×7 surface taken at 1.3 V, 100 pA at different PI settings. a) Gain settings too low, where the features are smeared due to a slow reaction to changes in height (proportional-gain 10 pm, integral-gain 5 nm/s.) b) Gain settings correct for ideal imaging, resulting in well defined features (proportional-gain 10 pm, integral-gain 60 nm/s.) c) Gain setting too high, where ringing is present throughout the scan (proportional-gain 5 pm, integral-gain 250 nm/s.)

topographic features which have a larger aspect ratio than the tip are not always representative of the feature itself. This is because the image acquired is a convolution of both the shape of the feature, and of the tip itself. The principle of how the shape of the tip can effect the apparent topography of a surface is shown in Figure 2.10a, where it can be seen that a sharp feature can only be imaged accurately with an equally sharp (or sharper) tip, whereas for a blunt tip imaging a sharp feature, an effective image of the tip is produced.

Figure 2.10a also shows an example of "dead-zones" in a line scan, which are areas where the tip is not sharp enough to probe between sharper features, and so no signal is measured from this area at all. This results in the complete loss of information from such areas which can only be rectified by scanning with a sharper probe.

An example of the effect of the tip state on imaging can be seen in Figure 2.11a, where throughout a single scan the tip changes twice, changing the apparent topography of the surface. Here, the Si(111) - 7×7 should appear

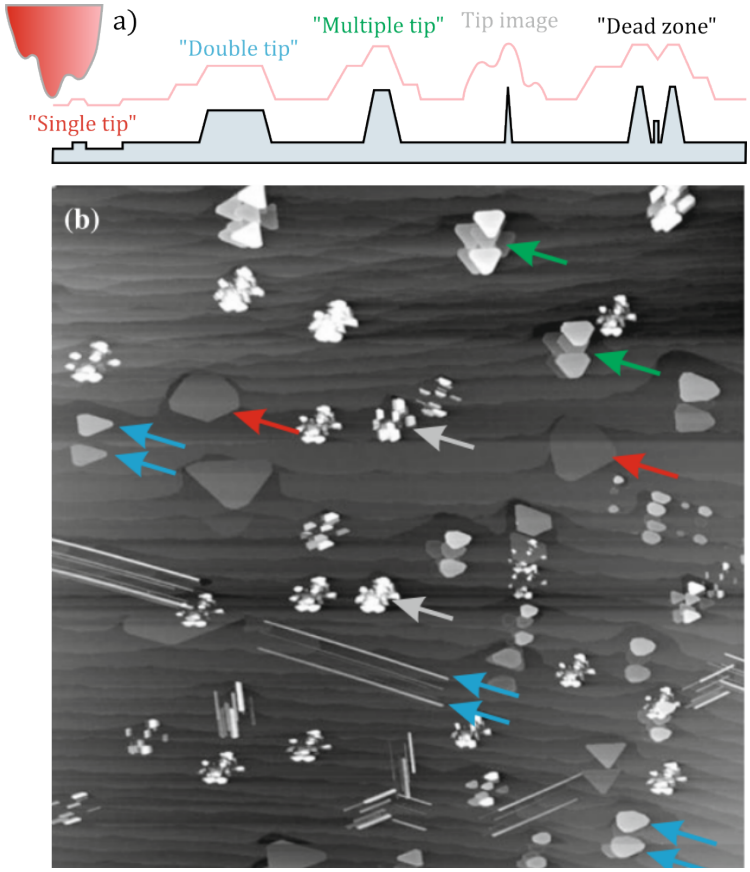


Figure 2.10: a) Sketch of a multi-tip giving rise to varied imaging of surface features. The *red line* shows the trace of the tip as it travels over the surface, attempting to keep the imaging signal constant. If the tip is too blunt to be able to image between certain features, ”dead zones” can appear as shown on the far right. b) Example of silicide nano islands and nano wires imaged in STM. Features with a higher aspect ratio result is a stronger tendency toward multi-features. Low aspect ratio features can be imaged with a single apex (*red arrows*), whereas higher can result in doubled (*blue arrows*) or multi-tip (*green arrows*) features. Very high aspect ratio surface features result in imaging of the structure of the tip rather than the surface feature (*gray arrows*). Figure adapted from Voigtlander *et al.* [7].

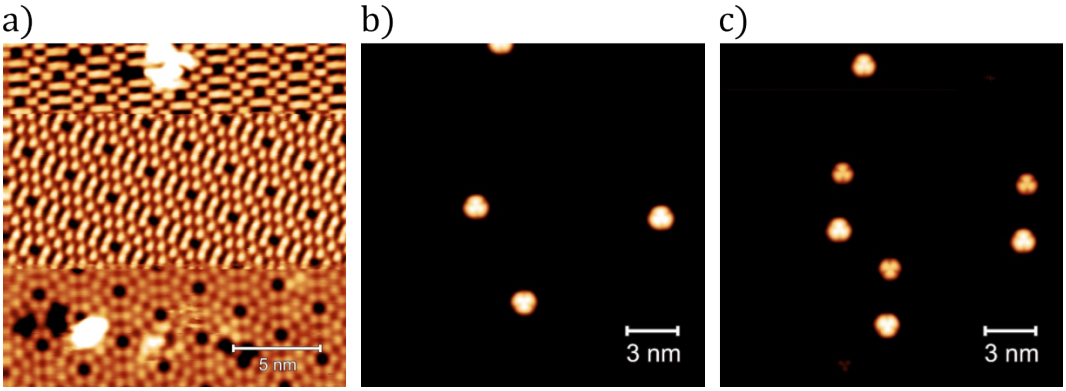


Figure 2.11: a) Si(111) - 7 × 7 imaged in STM at 2 V, 200 pA with multiple tip changes throughout the image. b-c) Cu(111) with a low coverage of C_{60} molecules imaged at 5 K, 100 mV, 100 pA showing a sharp probe tip in b) and a doubled tip, resulting in shadows in c).

as in Figure 2.4b, with spherically shaped, well resolved atoms, however, when scanning using differently shaped tips, the atoms can change apparent shape considerably. In the image shown in Figure 2.11b, the Cu(111) surface can be seen with a low coverage of C_{60} molecules. Comparing this image to that shown in Figure 2.11c, additional molecules appear in the same pattern as in Figure 2.11b, however both images were taken over the scan area. This is an example of shadows or "ghost" features appearing in an image resulting from additional minor apexes positioned further up the shaft of the tip, which result in a current signal being generated from additional locations to the main tip apex.

In general, if a topographical feature is sharper than the scanning tip, these features will not be imaged correctly [7]. Attempts have been made to deconvolve a real-space topographical image [14], however these are often not very successful for a few reasons: a) The exact shape of the probe tip is generally unknown, with the only way to "measure" the state being scanning a very sharp feature on the surface which is not always practicable, and b) The state of the probe tip changes often, and so deconvolution of images would need repeating just as often.

Effects such as these and the distortions discussed in Section 2.1.2.3 means that images should be considered carefully, and measurements should be

reproduced with multiple tips in order to exclude artefacts as much as is possible.

2.1.2.5 Tip Preparation

The preparation of probe tips is incredibly important in STM, as the shape of the tip defines the resolution and overall quality of images. As discussed previously, tips should have a minimal radius of curvature in order to be able to probe pits in a surface. STM tips are usually made of one of two materials: PtIr wire, commonly used under ambient conditions due to its resistance to oxidation in air [7] and Tungsten (W) wire, which is usually used under UHV conditions due to its high melting point and high mechanical strength. The latter is the material which was used for the entirety of this project.

For a W tip, preparation begins outside of the chamber commonly with electrochemical etching, shown in Figure 2.12, where a piece of W wire is placed into a solution of NaOH and held at a positive potential, toward a stainless steel counter electrode. The etching process takes place at the surface of the solution, where convection brings in fresh OH^- ions which react with the W wire, followed by the heavy W anions flowing down the wire, protecting the lower section and causing a natural necking which eventually breaks, producing a sharp tip.

Fresh from etching, the tip will be covered with an oxide layer and other contaminants, which will need to be removed before use, to ensure an ideal metallic tunnelling junction is present. There are several in-vacuum tip treatment methods which can be applied to achieve this. Tip heating via electron bombardment is a commonly used tip treatment method. Here, the tip is biased close to a filament (source of electrons) which is restively heated. On heating, this filament will emit electrons which are attracted to the biased tip, causing it to heat and expel contaminants, removing the outer oxide layer.

Compared to these *ex-situ* tip preparation methods, *in situ* (in the scan head during scanning) methods are somewhat lacking in finesse, where microscopists use a combination of experience and serendipity to crash a

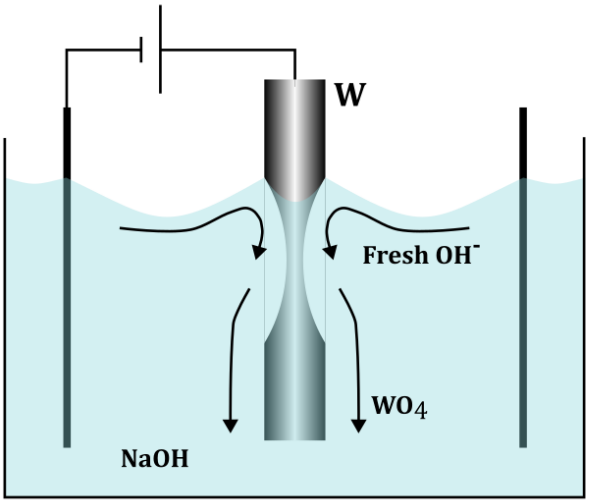


Figure 2.12: Schematic of an electrochemical tip etching setup.

tip repeatedly into a surface, hoping to pick up or remove material from the tip, or applying a sudden bias pulse to the tip in the hope of coaxing the tip into a better shape. More commonly than actually preparing a suitable tip, a microscopist will spend multiple hours using a combination of the two with no such luck. This problem with *in situ* preparation of one of the major motivations for this thesis, specifically Chapters 5 and 6.

2.1.3 Scanning Tunnelling Spectroscopy

An additional use of STM is the ability to obtain spectroscopic data, probing specific sites on a surface and allowing for the investigation of the electronic properties of a system with atomic resolution. The tunnelling current in STM can be expressed as [7]

$$I = \frac{4\pi e}{\hbar} \int_0^{eV} \rho_{tip}(\epsilon - eV)\rho_{sample}(\epsilon)|M(\epsilon)|^2 d\epsilon \tag{2.29}$$

with ρ_{tip} and ρ_{sample} corresponding to the density of states of the tip and

sample, and the $|M(\epsilon)|$ term being the tunnelling matrix, whose energy dependence is relative to the sample Fermi level. STS aims to measure the LDOS of a sample, which can be achieved by measuring the current-to-voltage characteristic of the tunnel junction. A small increase in the voltage, dV, shifts the sample states down, increasing the current by a small amount, dI. To first approximation, this additional current dI per voltage increase $dV = d\epsilon/e$ is given by the integrand in Equation 2.29, taken at the upper limit of the integral $\epsilon = eV$ [7]

$$\frac{dI}{dV} \approx \frac{4\pi e^2}{\hbar}\rho_{tip}(0)\rho_{sample}(eV)|M(eV)|^2 \tag{2.30}$$

In the simplest approximation, the density of states of the tip and the tunnelling matrix are considered voltage independent, hence the differential conductance is proportional to the density of states of the sample

$$\frac{dI}{dV} \propto \rho_{sample}(eV) \tag{2.31}$$

Using this approximation, the differential conductance measures the LDOS at the position of the tip.

A number of omissions are made to allow for this approximation, including the effect of the effect of the density of states of the tip on the differential conductance, and the current contributions from all levels with an energy lower than eV.

An STS spectrum (differential conductance) can be obtained in practice using one of two methods. Both begin by positioning the STM tip at a desired lateral position on the surface, whilst scanning in STM feedback. At this point, the feedback loop is disabled, keeping the tip-sample distance constant throughout the spectroscopy measurement. The voltage is then swept through a range of values whilst measuring the current, which is obtained as a function of the varying voltage, $I(V)$. This curve can then be differentiated with respect to the voltage to obtain the differential conductance, $\frac{dI}{dV}$, spectra.

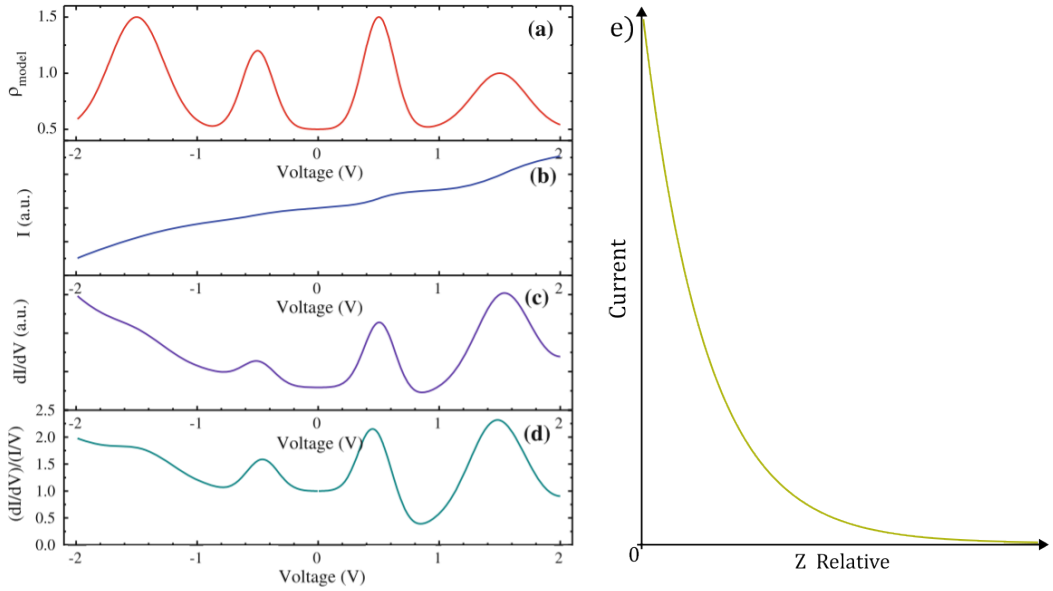


Figure 2.13: Examples of different STS spectra. a) Model LDOS of a sample. b) $I(V)$ curve which would be measured, from which the conductance or $\frac{dI}{dV}$ curve, c), can be calculated and then normalised as shown in d). e) shows the exponential dependence of the current on the tip-sample distance as would be measured in an ideal $I(Z)$ curve. a-e) from Voigtlander *et al.* [7].

Alternatively, the derivative signal, $\frac{dI}{dV}$, as a function of voltage, can be directly measured using the lock-in technique. In this method, an AC signal is generated by applying a small modulation voltage, $V_M cos(\omega t)$ to the bias. Due to this modulation, the measured current is expressed as

$$I(t) = f(V + V_M cos(\omega t)) \tag{2.32}$$

Where V_M is the modulation amplitude and ω is the frequency. Applying this modulation around a central voltage, creates a corresponding modulation in the measured tunnel current, with an amplitude proportional to the gradient of the $\frac{dI}{dV}$ curve at that bias. Therefore, once the tip is in position, the bias can be swept through a range, whilst applying the modulation. The resultant current can then be detected by a lock-in amplifier, where its amplitude for small values of V_M is proportional to $\frac{dI}{dV}$, as can be seen in Figure 2.14.

The $\frac{dI}{dV}$ spectra can be used to locate distinct energy levels within a sample, as the availability, and thus the conductivity, of these states change with the applied bias. An example $I(z)$, $I(V)$ and $\frac{dI}{dV}$ can be seen in Figure 2.13b, c and f respectively. To emphasise the desired signal from the $\frac{dI}{dV}$, it can also be normalised by the total conductance, I/V, as shown in Figure 2.13d.

An additional spectroscopy measurement which can be obtained in STM is the current dependence on the tip-sample distance, $I(z)$. This is obtained by positioning a tip laterally over a desired feature, before sweeping through a set of z-heights. $I(z)$ spectra enable us to determine whether a tip is in tunnelling, as opposed to mechanical contact, with the surface; when in tunnelling, the $I(z)$ curve will show an exponential dependence of the current on the tip height (Figure 2.13e), whereas in mechanical contact the curve will show a different trend. In addition to this, the decay rate observed when in tunnelling contact can, in principle, be used to estimate the work function, ϕ, of a material.

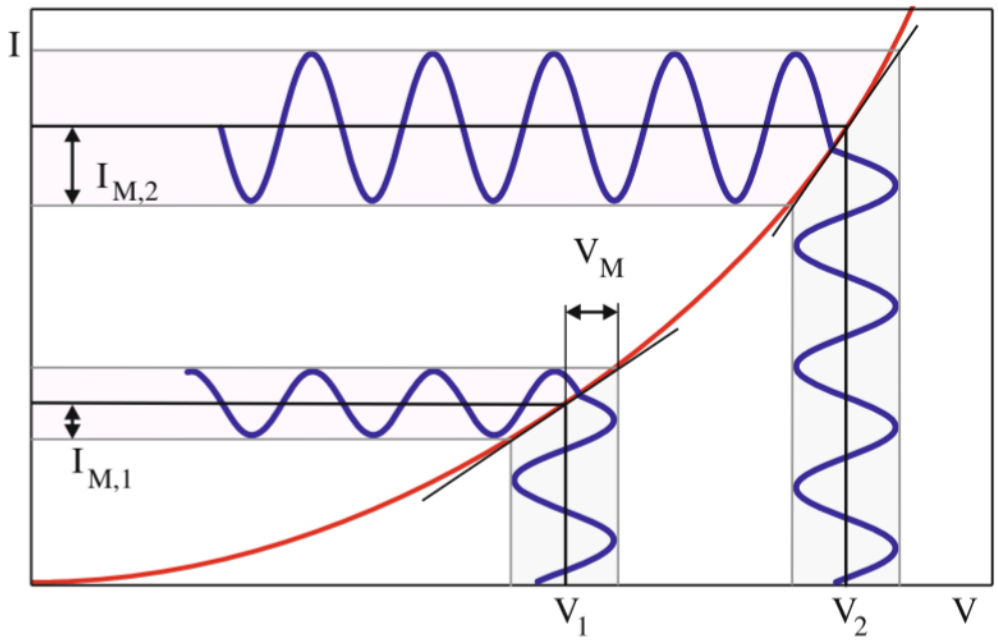


Figure 2.14: Graphical representation of direct measurement of the $\frac{dI}{dV}$ signal. The voltage is modulated around V_1 or V_2, by a small amount V_M, where the measured current amplitudes, $I_{M,1}$ or $I_{M,2}$, are proportional to the gradient of the $\frac{dI}{dV}$ curve at V_1 and V_2 respectively. Figure from Voigtlander *et al.* [7].

2.2 Machine Learning

2.2.1 Introduction

Machine learning (ML) is a branch of computer science which aims to write programs which are able to solve specific tasks through "learning" rather than relying on specific programming. Although the core concepts of machine learning date back almost a century, the popularity of these techniques have only began to increase in the past few years, to the point of being used in almost all aspects of daily life, with examples including earthquake detection [15], adaptive battery and brightness management in smartphones [16], sports injury prediction [17], mental illness prediction based on social media profiles [18], and lameness detection in cattle [19]. One area in which ML is often held in high regard is image analysis, where a ML model is able to make classifications based on complex patterns found within data, corresponding to manual labels given to the image as a whole. A key factor behind the explosion in the use of ML is the prevalence of open source tools such as TensorFlow, Keras, and Pytorch, in addition to the increased computing power available to the general user. These have allowed for the mainstream use of, among other things, convolutional neural networks (CNNs), which work exceptionally well at extracting features from 2D images, allowing for image classification tasks.

The "Hello, world" program for a CNN is generally considered to be a classification network using the widely used MNIST [20] dataset, containing 60,000 handwritten digits for training, and 10,000 for testing. Conventional computing methods would require each individual configuration of pixels which could combine to make a digit to be defined, in order to make accurate classifications. However, subjective tasks such as these are easily solvable using neural networks.

The most common use of ML, including all work presented in this thesis, comes under the category of supervised ML, which will be discussed in detail in Section 2.2.2. This kind of ML focuses on constructing models from an initial dataset (training set), and using this to perform actions on another dataset (test set).

This section will provide an overview of the fundamentals of ML, with a specific focus of deep learning and CNNs, before summarising some of the ML methods which have been applied to SPM.

2.2.2 Supervised Machine Learning

The category of supervised machine learning encompass a large range of different methods. Classification is one of the most common learning methods and involves training a model which, given a set of training data, is able to put new examples of similar data into classes based on observed features within the training set; this will be discussed in detail in the following sections. Regression is a supervised task, in which a program is trained to predict a numerical value given some input; an example of this would be the prediction of housing prices based on details of the house. Anomaly detection is a method where a model is trained to sift through a set of events or objects, and flag unusual or atypical items [21], this is commonly used in fraud detection. Synthesis and sampling involves a task where the model aims to generate new examples which are similar to those contained within the training data; an example of this would be generating a waveform to mimic spoken language from a sentence. However, only the classification method was used in this work.

2.2.2.1 Basic Classifier

In a basic supervised classification network, we start with a vector input, $x \in \mathbb{R}^n$, which contains n data points. In general, x could contain any kind of information; in SPM, this would be a topographic scan containing n pixels, or a spectra with n data points. The final goal here is to map this data, x to another vector, y, which describes a data label, of length k. In a classification network such as this, x will always belong to one of k distinct states. In a binary classification $k = 2$ (*e.g.* does this image contain a dog?), however multi-category classifiers are also trainable where $k > 2$ (*e.g.* what animal does this image contain?).

We can carry out this mapping of x to y, also known as "learning", by optimising a function $f(x)$ such that

$$y = f(x) \tag{2.33}$$

$f(x)$ is optimised in such a way as to estimate y as close to $\hat{y}$, the ground truth label, as possible [21]. For our particular case in SPM tip classification, the states of k in y could represent different imaging modes/tip states seen in a topographical scan. Specifically for the work carried out in this project, we make a binary classification between a good and a bad tip.

A basic neural network model will start with an input layer, with a number of nodes corresponding to the number of data points in the input data (for an image classifier this will be the total number of pixels). Following the input layer, are a number of layers known as dense training layers. The number of nodes per layer and the total number of layers can vary, but each of the nodes within the layer will have a weight and bias associated with them which will be discussed in detail in Subsection 2.2.2.2. After the training layers, the model will finish with an output layer, with the number of nodes depending on the desired configuration of the output.

2.2.2.2 Nodes, Weights, and Biases

The function $f(x)$ links x to y via L layers, containing a varying number of nodes, as shown in Figure 2.15. Each layer passes on a weighted average of the nodes contained within the layer, $\mathbf{A}^{l-1}$ onto the next layer, $\mathbf{A}^{l}$. The input at $\mathbf{A}^{0}$ is simply the training data, x, and the output at $\mathbf{A}^{L}$ is the estimated value of y. The weights are passed between layers by the following equations [21]:

$$\mathbf{Z}^{l} = (\mathbf{W}^{l})^{\mathbf{T}} \cdot \mathbf{A}^{l-1} + \mathbf{b}^{l} \tag{2.34}$$

$$\mathbf{A}^{l} = g(\mathbf{Z}^{l}) \tag{2.35}$$

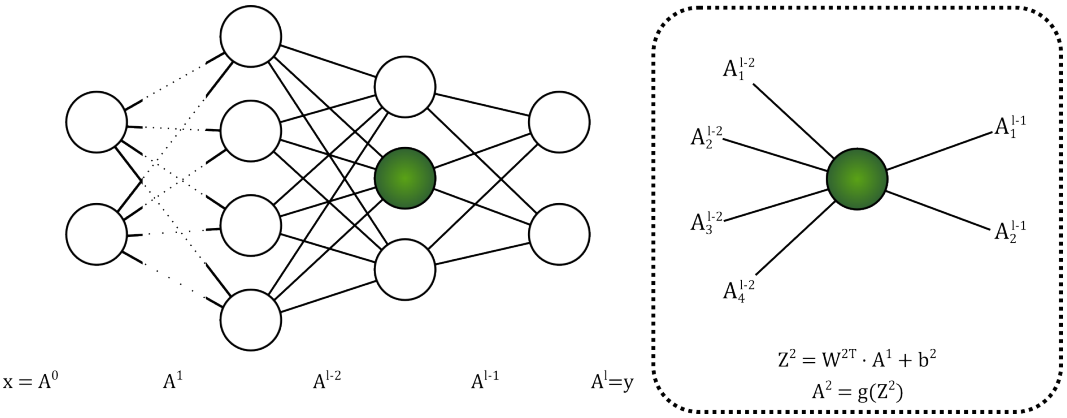


Figure 2.15: Pictoral representation of nodes transfering information starting with an input, x, in the input layer, to the final output layer $\mathbf{A}^l$ in a neural network.

Where $\mathbf{Z}^l$ represents the initial outputs of the nodes in layer l, and $\mathbf{b}^l$ and $\mathbf{W}^l$ represent the learnable hyper-parameters, biases and weights of each layer, and $\mathbf{A}^l$ are the activations of the nodes after passing through the activation function, g^l. This is shown in Figure 2.15. The goal of training in a network is to find the optimal values of both $\mathbf{W}^l$ and $\mathbf{b}^l$ which will result in the best estimate of y for a given input, x. The output of each node is passed through a non-linear activation function, $g(\mathbf{Z})$, which is then input into further nodes and will eventually result in a prediction, y.

2.2.2.3 Activation Functions

Currently, the output of each node, $\mathbf{Z}^l$, is linear and can take any value $-\infty \leq \mathbf{Z}^l \leq \infty$. The activation function, $g(\mathbf{Z}^l)$ (sometimes referred to as the transfer function), is used to regulate this output into a more usable form, usually the function will be a non-linear function which allows the output to adapt better to non-linearities in the data it is given. In addition, these activation functions help to constrain the output from each node to a suitable range for the problem at hand (*i.e.* constraining the final output of a model such that $0 \leq \mathbf{Z} \leq 1$). There are many examples of activation functions which can be used in different instances: sigmoid, tanh, ReLU,

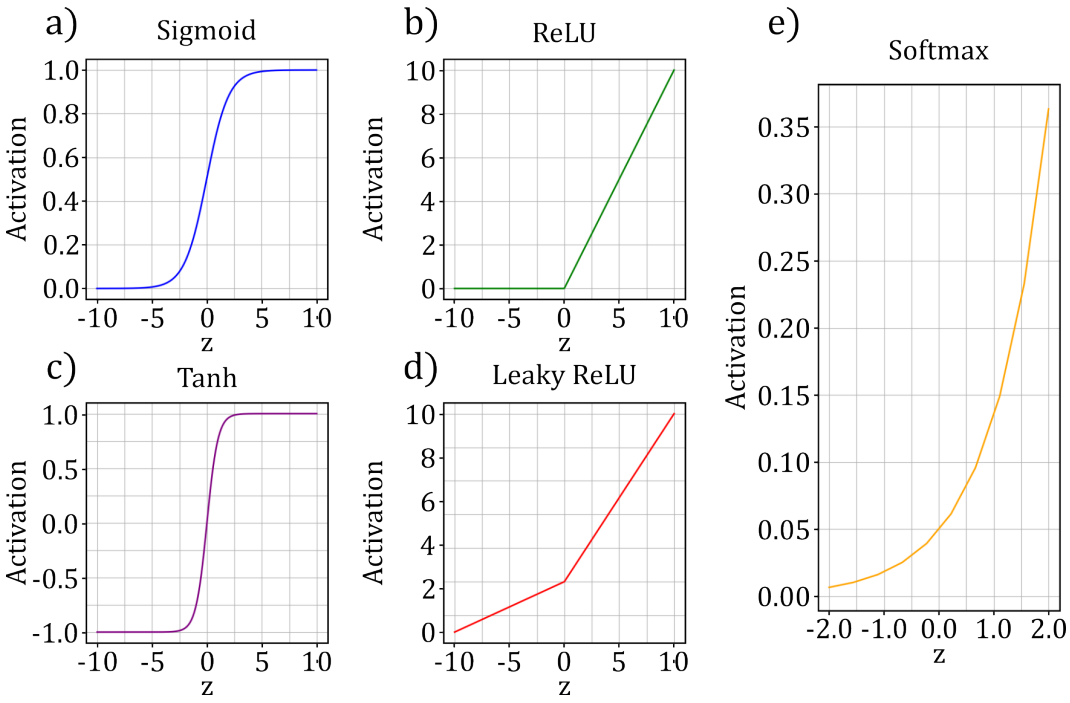


Figure 2.16: Graphical representations of the a) sigmoid, b) ReLU, c) tanh, d) Leaky ReLU and e) softmax activation functions.

Leaky ReLU, and softmax are the most commonly used examples and are shown in Figure 2.16.

$$\text{sigmoid}(z) = \frac{1}{1+e^{-z}} \tag{2.36}$$

$$\text{ReLU}(z) = \max(0, z) \tag{2.37}$$

$$\text{Leaky ReLU}(z) = \max(0.1 * z, z) \tag{2.38}$$

$$\text{softmax}(z) = \frac{e^{z}}{\sum_{k} e^{z_k}} \tag{2.39}$$

One important aspect of these activation functions is that they are differentiable, with the first derivative in the range $0 \leq g'(\mathbf{Z}^l) \leq 1$, which aids

in the training of the model, discussed later in Subsection 2.2.2.5. For the final output layer of the model, activation functions are typically chosen to have an output of $0 \leq g(\mathbf{Z}^l) \leq 1$, which allows for the output node to give a percentage confidence in each of its classifications. Here, an output of 1 would represent a high confidence that x belongs to a given category, while an output of 0 would denote a high confidence that x does not belong in that category. Additionally, a confidence around 0.5 corresponds to a classification which is very unclear, and so is unreliable.

Typically, nodes within the model (including dense training layers and convolutional layers) use a ReLU activation function, shown in Equation 2.37; these functions are used commonly as they prohibit all neurons in the layer from being activated at once, which is computationally efficient. For the final output layer, common choices for $g(\mathbf{Z}^L)$ are the sigmoid or softmax functions (Equations 2.36 and 2.39 respectively). The sigmoid function outputs k values, corresponding to the number of classes available for classification, where the sum of the k outputs must sum to 1. For this reason, the sigmoid function is commonly used for models where the input data can only belong to a single class. The softmax function outputs the same k values, but these are constrained to be between 0 and 1, and represent independent confidences for each class. Therefore the softmax function is more commonly used to multi-label models.

2.2.2.4 Convolutional Layers

One way of significantly improving the prediction accuracy of a neural network, specifically when the input is in the form of an image, is the use of convolutional layers before the dense training layers. These convolutional layers have the effect of converting the input image into specific features present within the image, as well as reducing the number of optimisable parameters within the model itself, which simplifies the overall learning task and improves performance. In addition, convolutional layers introduce translational invariance, whereby patterns are detected within the image regardless of their spatial location. This results in CNNs being particularly effective in image analysis tasks, as they can recognise objects and features, even when shifted or partially obstructed. Each convolutional layer will

contain three important parameters: the number of kernels, the kernel size and stride length. The effect of a kernel applied to an input image is shown in Figure 2.17.

Different kernels are able to extract different features within the input image, such as edge detection about different directions, sharpening and blurring [22]. The total number of kernels chosen per layer will decide how many are applied to the input image, and therefore how many feature extractions will be passed onto the next layer. Here, all kernels will be applied to the same image and the outputs of each passed on to the next layer, with future convolutional layers applying kernels to the outputs of the previous layer. The kernel size parameter decides on it's shape; larger kernels will allow for greater feature compression, however will lose more detail when compared to smaller ones. The size of the kernel will also have the effect of changing the features selected within the input image, so the kernel size must be carefully chosen to represent its input. The stride length decides on how many pixels the kernel is shifted over the input per step, which again must be chosen carefully to balance feature compression and useful output. The effect of kernel size and stride length is shown in Figure 2.17.

Commonly, layers known as pooling layers are used after convolutional layers which, similarly, apply a kernel of variable size to the output of the previous layer. This kernel will apply different operations on the input image depending on the type of pooling being used. A popular choice is max pooling, where the kernel will simply output the maximum pixel value contained within it at each step. The pooling layer has both the effect of further reducing the number of output features, and helps to make the output invariant to small translations of the input as a result of the convolutional layers [21].

2.2.2.5 Learning and Back-propagation

Looking back at Equation 2.34, we know that the values for $\mathbf{W}$ and $\mathbf{b}$ in each node determine how the classifier makes its predictions for each piece of input data. At the beginning, the values for these will be random, and contain no helpful information on making a prediction; we therefore need to start the learning phase. In this phase, the aim is to optimise these

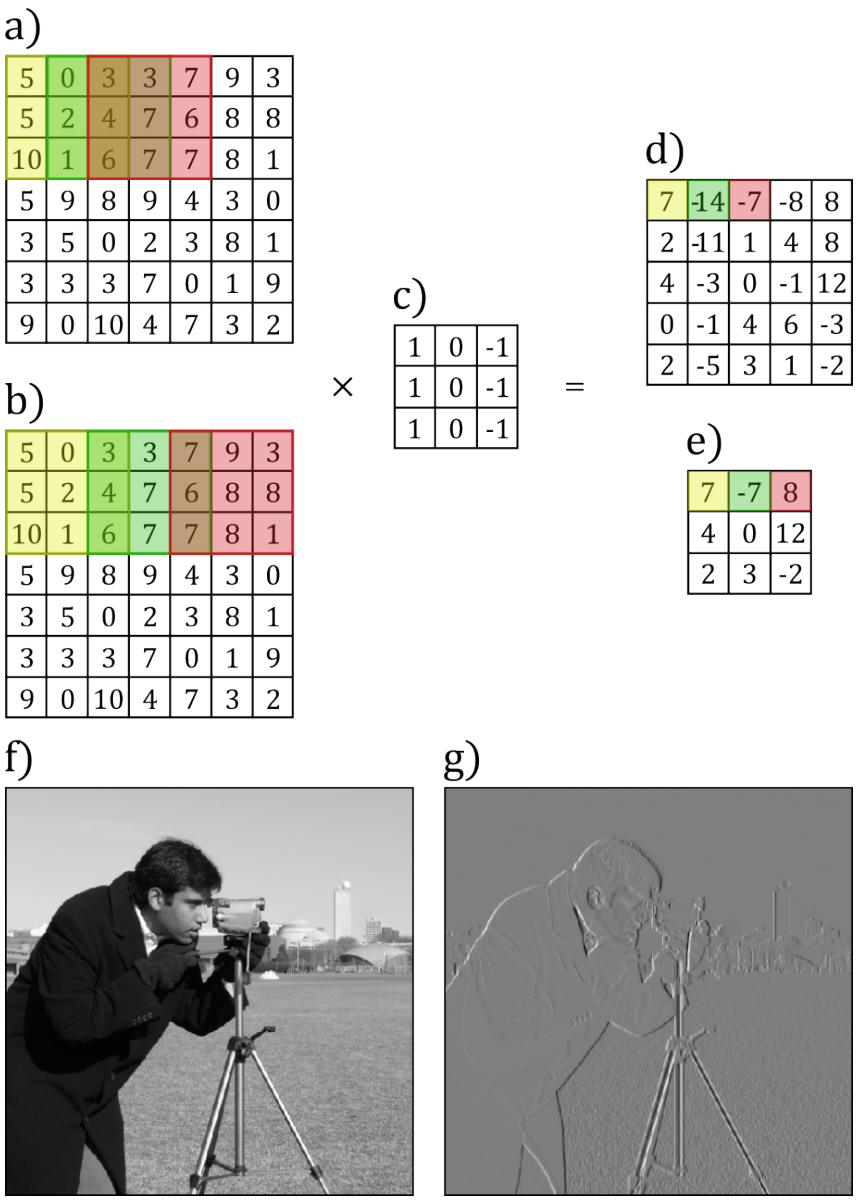


Figure 2.17: Pictorial representation of the effect of convolutional kernels on 2d input data with different stride lengths. a-b) Input array with a 3×3 kernel applied with stride lengths of 1 and 2 respectively. c) 3×3 kernel used for vertical edge detection. d-e) result of applying c) to a-b) respectively. f) Input image, which then has the vertical edge detection kernel, c), applied with a stride length of 1, resulting in the output, g).

parameters to values which best map the input, x, to the output, y. One, extremely poor, method of carrying out this optimisation would be a brute force attempt, where we try all combinations for each node and take the values which result in the highest overall accuracy. The VGG16 network [23] is an example of a well established convolutional neural network (CNN), trained for object detection which is able to classify images between 1000 different categories with an accuracy of 92.7%. This network is 16 layers deep, and as such contains a total of 1.38×10^8 trainable parameters. Obviously for a network such as this, the brute force method would be not be viable.

We are able to tweak our weights and biases in a much more efficient way by utilising the continuous nature of the output of each neuron. As mentioned in Section 2.2.2.3, the output $0 \leq y \leq 1$ represents the confidence in a model's prediction. If our model is making an incorrect prediction, with a high confidence, it stands to reason that the weights and biases at this point in time need adjusting much more than a network which is making a confident correct classification. To account for this, we define what is known as a loss function, $C(y, \hat{y})$. This function compares the prediction of the model for a specific input to the ground truth label assigned by a labeller, which is assumed to be correct. Similar to activation functions, the loss function can be defined in many different ways, depending on the problem. In classification problems we usually use either the binary cross-entropy function, or categorical cross entropy function described below [21]:

$$C_{\text{categorical}}(y, \hat{y}) = -\frac{1}{m} \sum_{i=1}^{m} (\sum_{k} y_k \log(\hat{y}_k)) \tag{2.40}$$

$$C_{\text{binary}}(y, \hat{y}) = -\frac{1}{m} \sum_{i=1}^{m} (y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{2.41}$$

Where m is the total amount of training data used per batch. After a single forward run through the network, $C(y, \hat{y})$ will have been calculated. In our model, $C(y, \hat{y})$ can only be reduced by altering one of three values: the weights, biases and the activations being input into the final output layer.

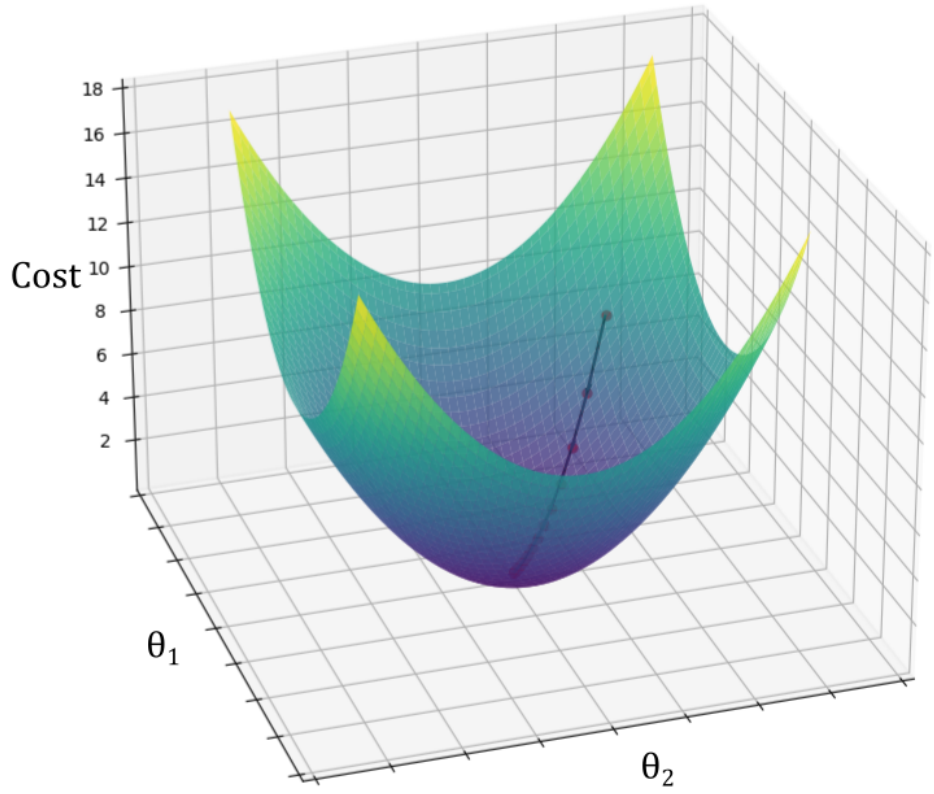


Figure 2.18: Example of a simple cost function with respect to two tune-able variables, θ_1 and θ_2. The dots represent steps in gradient descent, where the variables are adjusted to minimise the cost/loss.

Whereas the weights and biases can be directly altered, the activations can only be changed by varying the weights and biases being input to that layer. Therefore, we use an algorithm known as *back-propagation* to update the values for $\mathbf{W}$ and $\mathbf{b}$ by a small amount $d\mathbf{W}$ and $d\mathbf{b}$, based on $C(y, \hat{y})$.

Back-propagation is a method of determining how a single training example would alter the weights and biases input into a specific node in order to most rapidly decrease the cost, $C(y, \hat{y})$. This is done by calculating the gradient of $C(y, \hat{y})$ with respect to the weights and biases applied to inputs for the current node. The negative of the gradient represents the direction a weight or bias needs to be adjusted, in order to decrease the overall cost

of the model as shown in Figure 2.18. The gradient is calculated via the chain rule, propagating the error backward through the network in order to improve the classification accuracy. Using this, we can differentiate $C(y, \hat{y})$ to find $d\mathbf{W}$ and $d\mathbf{b}$ with the following equations and using equations 2.34 and 2.35:

$$\frac{\partial C}{\partial \mathbf{A}^{l-1}} = \frac{\partial C}{\partial \mathbf{Z}^{l}} \frac{\partial \mathbf{Z}^{l}}{\partial \mathbf{A}^{l-1}} = (\mathbf{W}^{l})^{\mathbf{T}} \frac{\partial C}{\partial \mathbf{Z}^{l}} \tag{2.42}$$

$$\frac{\partial C}{\partial \mathbf{Z}^{l}} = \frac{\partial C}{\partial \mathbf{A}^{l}} \frac{\partial \mathbf{A}^{l}}{\partial \mathbf{Z}^{l}} = \frac{\partial C}{\partial \mathbf{A}^{l}} g'(\mathbf{Z}^{l}) \tag{2.43}$$

followed by

$$\frac{\partial C}{\partial \mathbf{W}^{l}} = \frac{\partial C}{\partial \mathbf{Z}^{l}} \frac{\partial \mathbf{Z}^{l}}{\partial \mathbf{W}^{l}} = \frac{\partial C}{\partial \mathbf{Z}^{l}} (\mathbf{A}^{l-1})^{\mathbf{T}} \tag{2.44}$$

$$\frac{\partial C}{\partial \mathbf{b}^{l}} = \frac{\partial C}{\partial \mathbf{Z}^{l}} \frac{\partial \mathbf{Z}^{l}}{\partial \mathbf{b}^{l}} = \frac{\partial C}{\partial \mathbf{Z}^{l}} \tag{2.45}$$

This is the reason that it is essential for the cost functions to be differentiable. Using these equations, we can update the values of $\mathbf{W}$ and $\mathbf{b}$ with the following equations:

$$\mathbf{W}^{l} = \mathbf{W}^{l} - \alpha \frac{\partial C}{\partial \mathbf{W}^{l}} \tag{2.46}$$

$$\mathbf{b}^{l} = \mathbf{b}^{l} - \alpha \frac{\partial C}{\partial \mathbf{b}^{l}} \tag{2.47}$$

The value of α in Equations 2.46 and 2.47 is known as the learning rate which is not adjusted by the back-propagation process itself, but is slowly reduced over time by an optimiser. If α is too high, the adjustments to the learning parameters may never converge on a minima of $C(y, \hat{y})$, however if it is too low, the model may either converge on a local minima in $C(y, \hat{y})$, or simply take too long to reach a minimum. The role of the optimiser therefore is to adjust α such that our model reaches a global minima in a sufficient timescale. Commonly, the Adam [24] optimiser is used.

For each node in the current layer, the optimal adjustments to the input weights and biases are calculated, averaged and applied to the nodes in the layer. We then move to previous layers and repeat the back-propagation process to calculate the optimal adjustments for all the weights and biases in the model. This process is repeated for all training examples, averaging all adjustments before applying these to the model. In practise, rather than taking an average of the optimal adjustments for all training examples, the training data is split into batches, whose adjustments are averaged and applied. Applying the updates in these batches is known as stochastic gradient decent, and is computationally faster than averaging the adjustments of the dataset as a whole. We then repeat these steps of forward and back-propagation multiple times, or epochs, over the same dataset, until the value of $C(y, \hat{y})$ is sufficiently small or has reached a minimum.

This training of a model is computationally expensive, however once completed, the optimised weights and biases for the model can be saved, after which fast predictions can be made using efficient matrix multiplication, this is why inference is so fast whilst training is slow. At the end of each epoch, the model is tested on an isolated group of data known as the validation set. This set contains the same sort of variance of data as the training set, but is usually a fraction of the size and allows us to quantify the true performance of the classifier on unseen data. After this validation step, it is possible to tell if the model has overfit.

2.2.2.6 Overfitting and Underfitting

The main challenge in machine learning is to train a model which is able to perform well on *previously unseen* inputs, as opposed to those used to train

the model itself. The models ability to do this is known as generalisation. This generalisation however, is not always a given when training a model, which is why it is extremely important to separate out a portion of our input data for a final step known as validation, where our final model is tested on data which it has not seen before.

If a model is not performing as well as expected at the end of the training step it has usually either overfit or underfit to the data. Overfitting is where a model learns to make classifications based on noise from within the input data. Here, noise refers to any feature within the image which does not necessarily represent the classes present, but may, from the model's perspective, seem to be representative due to the configurations of the dataset and the model itself. Underfitting on the other hand, is where the model is not able to obtain a low enough error value on the training set, meaning that the model was unable to learn anything significant from the input data. Examples of over and underfitting are show in Figure 2.19.

Overfitting can be a result of a poor choice of dataset, over-training of the model (running for too many epochs) or poor configuration of the model. There are a few additions which can be made in order to reduce the chance of overfitting, including dropout, and class-weighting. Dropout has the effect of temporarily nullifying random nodes within a layer, which allows for the model to become less reliant on specific nodes, helping with generalisation. Class-weighting is often used when a poorly balanced dataset is input into the model, and allows for the model to consider less represented classes with a higher weight. For example, in a binary classification scheme, if one class makes up 95% of the entire dataset, the model may learn that it is able to achieve a 95% accuracy by simply classifying all new data as that of the majority class, which would not be ideal. Unbalanced datasets such as these are common place in SPM, as it is relatively rare that any random tip will result in a well defined image. In practise, class weighting is done by scaling up the importance of the data in the minority class, hence giving them a higher weighting in the calculation of the loss function when optimising features in a ML model. In this way, a more balanced ML network can be trained.

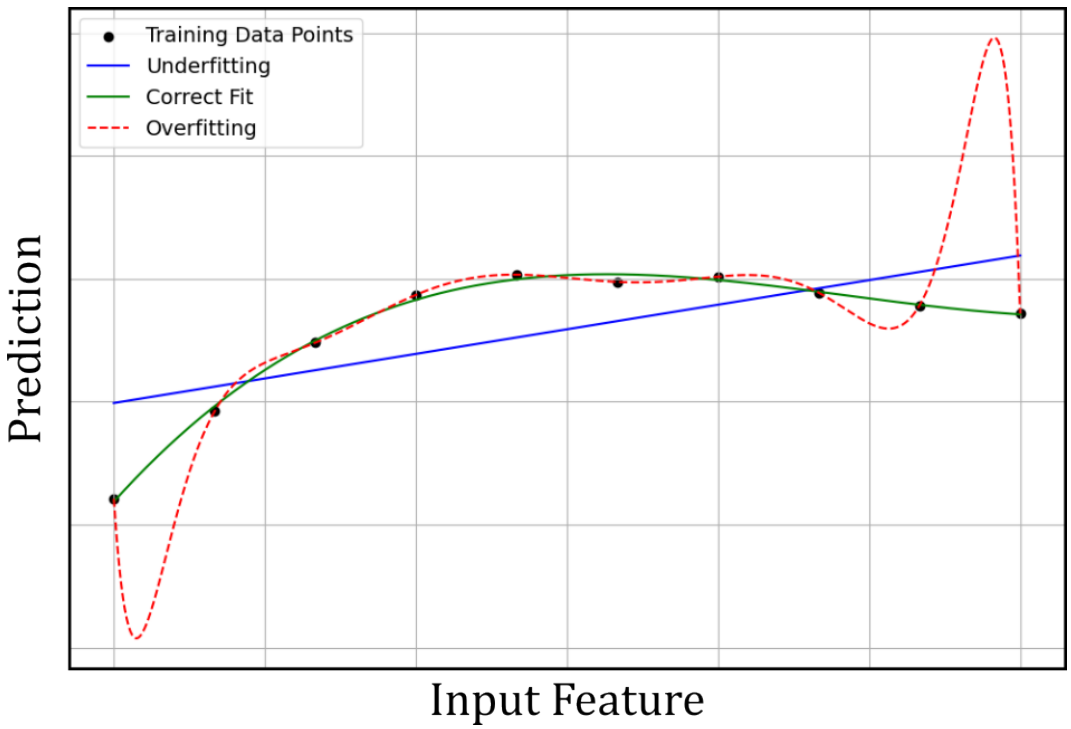


Figure 2.19: Illustration of the effects of underfitting, correct fitting, and overfitting in predictions on a dataset shown as black points. The blue line shows an underfitting model which fails to capture the trend of the data. The green line represents a correctly fitting model which appropriately captures the trend of the data. The red dashed line shows an overfitting model, which fits the training data perfectly but exhibits excessive variance, capturing noise rather than the true pattern.

2.2.3 Assessing performance

It is worthwhile when building a classifier to consider more performance metrics than simply the accuracy. The reason for this is that in specific cases, which are common with the types of datasets used in SPM, the accuracy may not truly be representative of the ability of the model to do the job it was trained to do. For example, if we were attempting to train a model to detect spam emails, with a dataset containing 99% non-spam emails, our model would achieve a 99% accuracy by simply predicting every email it comes across as not spam. This would be an apt assessment, in that this is the correct accuracy that the model is achieving on the dataset, however metric we have chosen tells us very little about the model's ability to do the required task. This problem is particularly evident in SPM, where when preparing a tip for imaging, the majority of the tips obtained would be "bad".

It is for this reason that it is best practise to consider other metrics when evaluating our models. As mentioned previously, our binary models will output a final classification in the form $0 \leq \mathbf{y} \leq 1$, where $\mathbf{y}$ represents the confidence of a prediction. To output a prediction from this value, $\mathbf{y}$ is rounded up or down, however the value which this is rounded about can be varied. With this varying rounding point, comes varying values of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). For image classification in SPM, a TP is where our model classifies an image as "good" when it has been manually labelled as "good", a TN where an image is classified as "bad" when manually labelled "bad", a FP where an image is classified as "good" when it is "bad" and a FN where an image is classified as "bad" when it has been manually labelled as "good". Changing this rounding position can effect the classifiers ability drastically. For example, increasing the confidence threshold needed to classify a tip as "good" would result in a reduced number of FPs, at the possible expense of increasing the number of FNs, and therefore decreasing the accuracy.

To quantify this, in addition to the overall accuracy of the model we can calculate the precision, the True Positive Rate (TPR) and the recall (also known as the False Positive Rate (FPR)) using the equations below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.48}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.49}$$

$$Recall/FPR = \frac{TN}{TN + FP} \tag{2.50}$$

$$TPR = \frac{TP}{TP + FN} \tag{2.51}$$

By comparing the TPR and FPR at different confidence thresholds, we can form the receiver-operator-characteristic (ROC) curves as shown in Figure 2.20. We can use the ROC curve to calculate another metric, the area under the ROC (AUROC), which quantifies the all around performance of a model. A perfect classifier will achieve an AUROC value of 1, and random guessing will give a value of 0.5. One important quality of this metric is that it is not effected by class imbalance, and therefore is much more descriptive of the ability of a classifier than relying solely on the accuracy.

When training a new ML model, it is common practice to train multiple models using varied hyper-parameters. The final chosen model is typically the one which demonstrates the highest overall performance at the end of training. Commonly varied hyper-parameters include the number of convolutional and dense layers, the number of neurons per dense layer, the number of filters in each convolutional layer, the convolution kernel size, and the use of dropout layers.

After training, the models are evaluated based on metrics such as accuracy and precision, which are plotted to show how these values evolve over a number of epochs. These plots illustrate the progress of a model during training, evaluated on both the training and validation datasets. During training, accuracy on the training dataset usually increases with more epochs, which will eventually lead to overfitting. To mitigate this overfitting, weights and biases are typically saved from the epoch at which validation accuracy stops improving, even if the model continues to perform better on the training dataset.

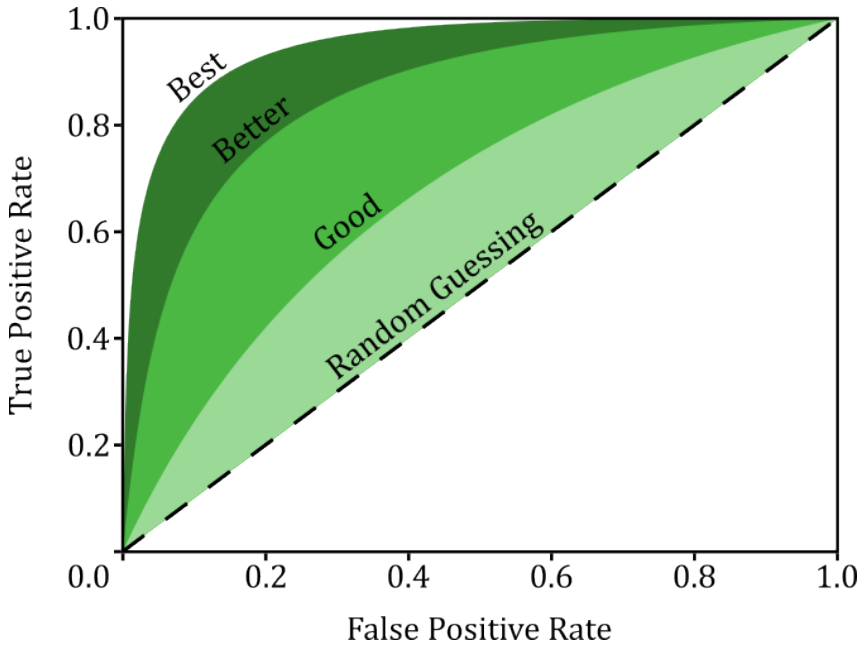


Figure 2.20: Receiver-Operator-Characteristic curve, showing various curves of differing abilities. A perfect classifier will achieve an area under ROC of 1, while guessing will have an area of 0.5.

2.3 Classification in SPM

Machine learning methods such as those discussed in Section 2.2 have been applied to various SPM methods with the aim of automating common tasks with some success. The overall focus of this is to reduce idle time on equipment, and automate time-consuming processes so as to open up an operators time to be applied elsewhere. In this section, some notable examples of ML based automation in SPM will be discussed, as well as a small number of examples without the use of ML.

2.3.1 Image Classification

For full automation of SPM methods, the first hurdle to overcome is the ability to prepare and maintain a scanning probe tip without the need for human intervention. Image classification in SPM can be used to tackle a number of problems; most prominent are the classification of the state of a probe tip, and classification of the surface itself. Both of these will use an topographical scan as input, however the classification of the surface will assume that the probe is already in an ideal state.

An early attempt to automate tip state classification (and preparation), was in the work carried out by Woolley *et al.* [25]. Here, the state of the probe was classified based on a deterministic comparison between an obtained topograph and one which is atomically flat, obtaining a metric known as the universal similarity metric (USM), which describes the similarity between two images, whilst working highly ordered pyrolytic graphite (HOPG) in ambient conditions using STM. The tip was altered *in situ* by applying voltage pulses, after which a scan would be taken, repeating this until imaging resulted in a flat topograph.

A somewhat more complex tip state classification was made by Rashidi *et al.* [26], where the group were attempting to make a binary tip state classification, between "sharp" or "double", by imaging dangling bonds (DBs) on the H:Si(100) surface. When imaged with a sharp tip, the DBs appear as single bright, round protrusions on the surface as shown in Figure 2.21b, whereas in the case that the tip is doubled, DBs appear as a grouping

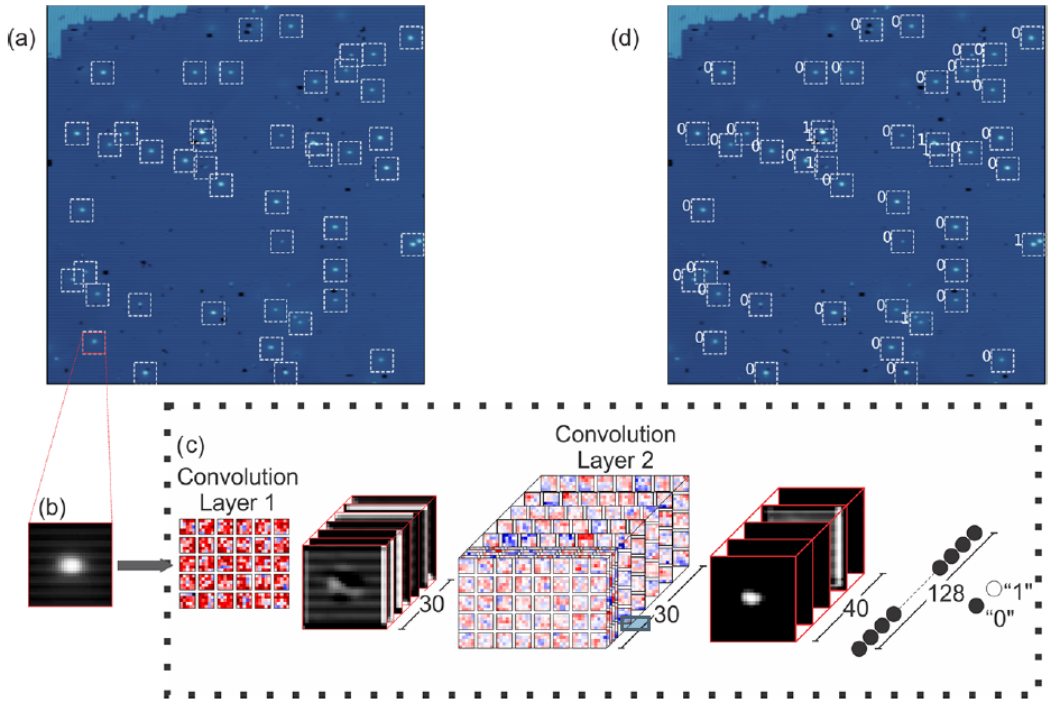


Figure 2.21: Tip quality detection using a CNN. (a) $100 \times 100nm^2$ STM image of hydrogen terminated Si(100). Bright features on the surface are dangling bonds, each white dashed box is used as input for the CNN. (b) Close up of a single DB used as input for the network taken from (a). (c) Schematic of the CNN used to determine the tip quality. (d) Outputs of the CNN for each DB found within (a) with "0" for a sharp tip and "1" for double tips. Image from [26].

of multiple protrusions. A small area around each protrusion is used for training and classification of a CNN. In addition to a CNN classification, the script employed majority voting by taking a large overview with dozens of DBs present, classifying each and taking the label which had the highest majority in the overview. With this setup, the group was able to achieve an overall accuracy of 99%. It should be noted however that the classification between a doubled and non-doubled tip is rather simple, and so a high accuracy is expected.

This tip state classification scheme was expanded upon further in work carried out by Gordon *et al.* [27], where rather than a binary classification, a

CNN model was trained to distinguish between five tip states which result in different imaging resolutions on the H:Si(100) surface (see Figure 2.22a-c). In this work, multiple, well established CNN architectures [23, 26, 28] were used for model training, with the final classification chosen as the majority label from the top three performing models. On H:Si(100), the final accuracy in multi-class classification was 78%, with a precision of 89%, whereas when converted to a binary "good" or "bad" classification, the final accuracy was 93% and a precision of 96%. In a later publication, Gordon *et al.* [29] advanced on this by allowing for tip state classifications to be made using only partial scans, which further reduces the time taken in tip classification routines by an approximate factor of one hundred. Additional ML based tip state classification attempts have also been attempted on full topographical scans [30, 31], as well as an attempt to classify the state of a tip based on spectroscopy in STM in work by Wang *et al.* [32] . In the latter publication, I(V) spectra were classified into five categories (shown in Figure 2.22e) based on quality, which resulted in a final ML based classifier with a precision of 97%.

In addition to semi-random *in situ* tip preparation, it is possible to enhance the resolution of scanning by the intentional functionalisation [34] of the probe by picking up a molecule on the end of a tip. In the work of Alldritt *et al.* [33], this functionalisation was automated using a CNN to classify the appearance of CO molecules using a CO functionalised tip. If the tip functionalisation is successful, the contrast of the CO molecules should switch from the depressions seen in STM with a metallic tip, to a characteristic "sombrero" shape, the appearance of both is shown in Figure 2.22g. Using CNN classifier, the group was able to achieve an overall accuracy of 95% and a precision of 90%

2.3.1.1 Surface Classification

As mentioned previously, image classification has also been applied to not only classify the state of a scanning probe tip, but the surface itself. This kind of a classification is usually for general experiment automation rather than the preparation of the probe. Here the work aims to locate features on

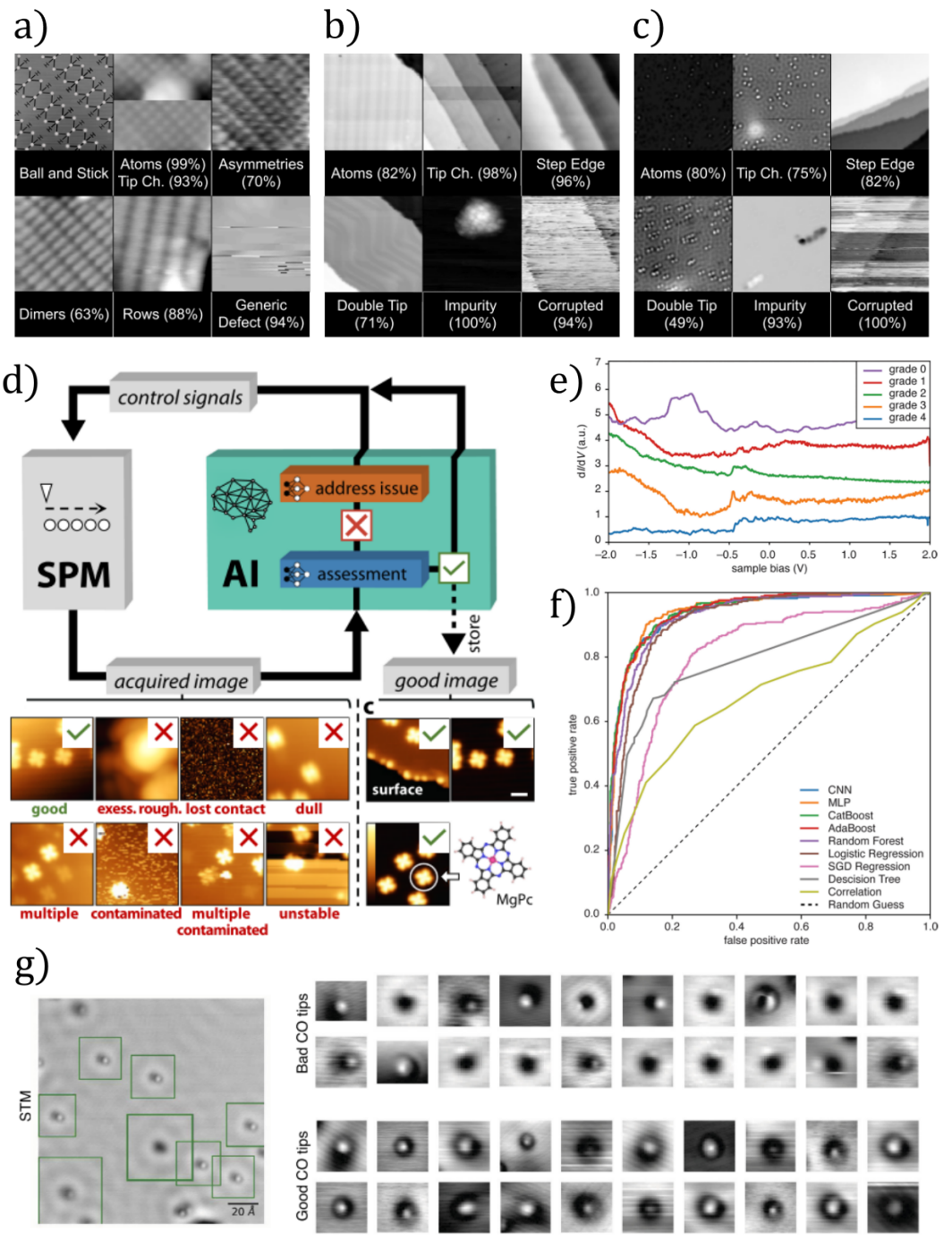


Figure 2.22: a)-c) Selection of STM images demonstrating the various tip states classified using ML in Gordon *et al.* [27], showing H:Si(100), Au(111) and Cu(111) respectively. d) Flow chart showing the automated tip classifier from Krull *et al.* [30], along with the various tip states which are classified. e) Example STS curves with different manual assignments, which were then classified via multiple ML classifiers, the ROC curves of which are shown in f). e)-f) from Wang *et al.* [32]. g) *Left*: Example experimental image with identified CO molecules highlighted by green squares. *Right*: Example of manual labels applied to the different molecules used to train the ML model for classification. g) from Alldritt *et al.* [33].

the surface or different imaging modes, assuming a good tip as a precursor to the experiment.

One such example of this is shown in work carried out by Woolley *et al.* [25], where after initial *in situ* tip preparation, a non-ML based method was used to automatically optimise the imaging parameters at which atomic resolution would be visible on HOPG. In STM, the imaging parameters (bias, tunnel current setpoint, and PI controller gains) are chosen by the user using (mostly) experience, rather than a well defined metric. The final script is able to optimise these parameters by making adjustments, and constantly comparing to an example of an ideal topograph.

In work carried out by Gordon *et al.* [27], discussed previously, in addition to the H:Si(100) surface, classifications were made on the Au(111) and Cu(111) surfaces, however rather than classifying the state of the tip, other features present in the images were taken into account such as the presence of step edges, and impurities in addition to the tip state. The same CNN methods were used, with accuracies and precisions comparable to the results from the H:Si(100).

In the work of Ziatdinov *et al.* [35], a CNN was trained to be able to classify structural and rotational states of individual molecules in an ensemble in STM. This method is able to distinguish between molecules adsorbed in different orientations with a high accuracy. This group has also worked previously on similar surface characterisations in scanning transmission electron microscopy (STEM) [36, 37], in one specific example, the group developed a ML model which is able to track complex defect transformations from unprocessed STEM data [38].

In work by Rashidi *et al.* [39], a ML model was produced which is able to make pixel-wise classifications in an image, labelling different defects and features visible on a H:Si(100) surface as shown in Figure 2.23. The training data for a model such as this involves labelling each pixel within an image, depending on the structure observed there.

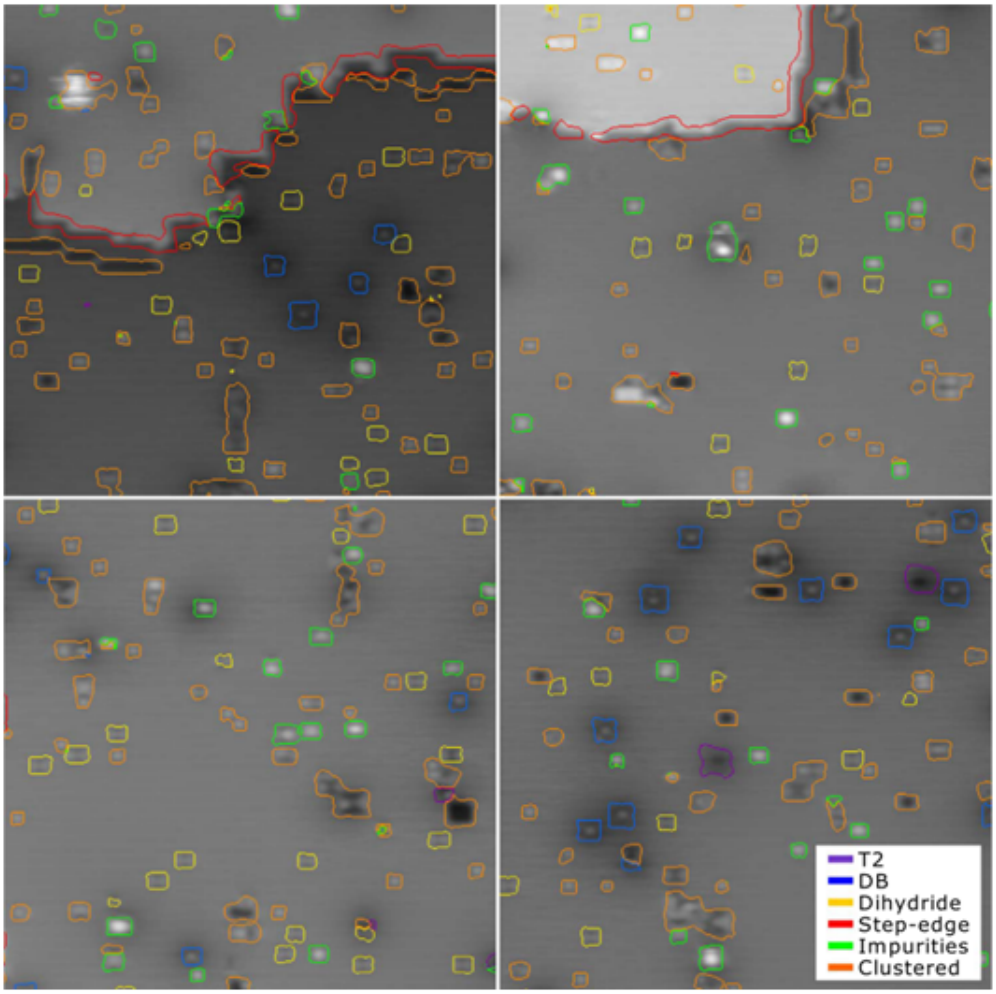


Figure 2.23: Four constant current STM topographs of the H:Si(100) surface with labels traced directly onto the images as carried out by Rashidi *et al.* [39].

2.3.2 Automated Manipulation and Patterning

One of the final aims of automation in SPM is to automate the process of manipulating individual atoms and molecules on surfaces, in order to create structures at the nanoscale, possibly leading to wide-scale fabrication of nano-structures. To this end, multiple attempts have been made to automate these positioning methods.

In the work of Celotta *et al.* [40], the group was able to create more complex nanostructures automatically without the use of ML. This work was carried out on a Cu(111) surface, with Co adatoms used as the medium for fabrication. Individual adatoms were identified by first scanning the surface, and using a threshold to identify adatoms. These could then be manipulated individually, using a set algorithm involving placement of the tip over the atom, before adjusting the voltage in order to bond the atom to the tip. Prior to movement, a path planning routine maps out the manipulation paths needed to move the visible adatoms into the desired shape, which was manually input by the user. Using this method, multiple structures were created, showing that ML is not a requirement. It should be noted that this work assumed the existence of a stable, sharp probe which does not change over the course of the experiment, as well as well defined manipulation and imaging parameters.

Specific parameters and methods for atom/molecule manipulation are not always immediately known for a new system. However one form of ML, known as reinforcement learning (RL), is widely used in cases such as these, e.g., improving self-driving cars [41], and learning to play games (often outperforming humans) such as Go [42]. RL employs a technique of exploring the options available, with a specific cost being applied to each action depending on how close to the goal it was able to achieve.

For example, the manipulation of a complex molecule from a surface is not a trivial task. In the case of a monolayer of PTCDA, manipulation involves forming a bond with one end of the molecule, and moving the tip away from the surface in a specific path. This manipulation path was found using RL in the work of Leinen *et al.* [43], where the model was allowed to move in any direction once bonded to the molecule, in addition to choosing the

location of the bond. Once trained, the model was able to remove a PTCDA molecule from the monolayer with a high precision.

A similar problem occurs with the manipulation of single adatoms on surfaces, where it is commonly preferable to perform lateral manipulation, where adsorbates are dragged across a surface, with the adsorbate hopping from site to site. The parameters needed here are the allowed manipulation directions (along the axes of the underlying lattice) and the tip-sample separation required for the tip to interact with the atom enough for it to be dragged across the surface. These parameters, again, are not known *a priori*, however in work carried out by Chen *et al.* [44], they were determined through RL. Using these parameters, the group was able to create artificial structures with atomic precision using Ag adatoms on Ag(111).

2.3.3 General Automated Experimentation

In addition to automated nanoscale fabrication experiments, generic processes involved in general experimentation have been automated. Firstly, in work by Krull *et al.* [30], the group created a ML model which is able to correctly maintain a probe tip throughout an experiment, stopping periodically to check the tip and correct it if needed. With this, a full automated exploration of a large 850×850 nm^2 area was completed by taking individual scans, mapping the area into suitable and non-suitable areas for imaging.

A potential issue with automated experimentation such as these is the fact that as there is no experimentalist watching the script as it scans through different areas, therefore, it is possible that novel or interesting features could be observed without being probed any further. This has led to some groups attempting to train models which are able to recognise these features, before taking higher resolution images [45] or spectra over specific sites [46].

References

(1) Giaever, I. *Physical Review Letters* **1960**, *5*, 147–148.

(2) Binnig, G.; Rohrer, H. *Surface Science* **1983**, *126*, 236–244.

(3) Tersoff, J.; Hamann, D. R. *Physical Review B* **1985**, *31*, 805–813.

(4) Sweetman, A.; Jarvis, S.; Danza, R.; Moriarty, P. *Beilstein J. Nanotechnol* **2012**, *3*, 25–32.

(5) Eigler, D. M.; Schweizer, E. K. *Nature* **1990**, *344*, 524–526.

(6) Binnig, G.; Rohrer, H.; Gerber, C.; Weibel, E. *Physical Review Letters* **1983**, *50*, 120–123.

(7) Bert Voigtländer, *Scanning probe microscopy*; NanoScience and Technology; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007, pp 77–81.

(8) Binnig, G.; Smith, D. P. *Review of Scientific Instruments* **1986**, *57*, 1688–1689.

(9) Wiesendanger, R.; Anselmetti, D.; Guntherodt, H.-J. *Europhysics News* **1990**, *21*, 72–73.

(10) Frohn, J.; Wolf, J. F.; Besocke, K.; Teske, M. *Review of Scientific Instruments* **1989**, *60*, 1200–1201.

(11) Peng, J.; Chen, X. *Modern Mechanical Engineering* **2013**, *03*, 1–20.

(12) Rahe, P.; Schütte, J.; Schniederberend, W.; Reichling, M.; Abe, M.; Sugimoto, Y.; Kühnle, A. *Review of Scientific Instruments* **2011**, *82*, 63704.

(13) Yothers, M. P.; Browder, A. E.; Bumm, L. A. *Review of Scientific Instruments* **2017**, *88*, 013708.

(14) Wang, Y. F.; Kilpatrick, J. I.; Suzi Jarvis, S. P.; Frank Boland, F. M.; Kokaram, A.; Corrigan, D. *IEEE Transactions on Image Processing* **2016**, *25*, 2774–2788.

(15) Mousavi, S. M.; Ellsworth, W. L.; Zhu, W.; Chuang, L. Y.; Beroza, G. C. *Nature Communications* **2020**, *11*.

(16) Samat, S. Android P: Packed with smarts and simpler than ever Web, 2018, https://blog.google/products/android/android-p/ (accessed 09/17/2024).

(17) Fiscutean, A. *Nature* **2021**, *592*, S10–S11.

(18) Kim, J.; Lee, J.; Park, E.; Han, J. *Scientific Reports 2020 10:1* **2020**, *10*, 1–6.

(19) Byabazaire, J.; Olariu, C.; Taneja, M.; Davy, A. *2019 16th IEEE Annual Consumer Communications and Networking Conference, CCNC* **2019**.

(20) Deng, L. *IEEE Signal Processing Magazine* **2012**, *29*, 141–142.

(21) Goodfellow, I.; Bengio, Y.; Courville, A., *Deep Learning*; MIT Press: 2016.

(22) Géron, A., *Hands-On Machine Learning with Scikit-Learn and TensorFlow*; O'Reilly: 2019.

(23) Simonyan, K.; Zisserman, A. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* **2014**.

(24) Kingma, D. P.; Ba, J. L. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* **2014**.

(25) Woolley, R. A.; Stirling, J.; Radocea, A.; Krasnogor, N.; Moriarty, P. *Applied Physics Letters* **2011**, *98*, 253104.

(26) Rashidi, M.; Wolkow, R. A. *ACS Nano* **2018**, *12*, 5185–5189.

(27) Gordon, O.; D'Hondt, P.; Knijff, L.; Freeney, S.; Junqueira, F.; Moriarty, P.; Swart, I. *Review of Scientific Instruments* **2019**, *90*, 103704.

(28) Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; Keutzer, K. **2016**.

(29) Gordon, O.; Junqueira, F.; Moriarty, P. *Machine Learning: Science and Technology* **2020**, *1*, 015001.

(30) Krull, A.; Hirsch, P.; Rother, C.; Schiffrin, A.; Krull, C. *Communications Physics* **2020**, *3*, 1–8.

(31) Zhuo, D.; Hou, L.; Masayuki, A. *Applied Physics Express* **2023**, *16*, 1–5.

(32) Wang, S.; Zhu, J.; Blackwell, R.; Fischer, F. R. *Journal of Physical Chemistry A* **2021**, *125*, 1384–1390.

(33) Alldritt, B.; Urtev, F.; Oinonen, N.; Aapro, M.; Kannala, J.; Liljeroth, P.; Foster, A. S. *Computer Physics Communications* **2022**, *273*, 108258.

(34) Gross, L.; Mohn, F.; Moll, N.; Liljeroth, P.; Meyer, G. *Science* **2009**, *325*, 1110–1114.

(35) Ziatdinov, M.; Maksov, A.; Kalinin, S. V. *npj Computational Materials* **2017**, *3*, 31.

(36) Kalinin, S. V.; Liu, Y.; Biswas, A.; Duscher, G.; Pratiush, U.; Roccapriore, K.; Ziatdinov, M.; Vasudevan, R. *Microscopy Today* **2024**, *32*, 35–41.

(37) Kalinin, S. V.; Ziatdinov, M. A.; Hinkle, J.; Jesse, S.; Ghosh, A.; Kelley, K. P.; Lupini, A. R.; Sumpter, B. G.; Vasudevan, R. K. *CoRR* **2021**, *abs/2103.1*.

(38) Ziatdinov, M.; Dyck, O.; Maksov, A.; Li, X.; Sang, X.; Xiao, K.; Unocic, R. R.; Vasudevan, R.; Jesse, S.; Kalinin, S. V. *ACS Nano* **2017**, *11*, 12742–12752.

(39) Rashidi, M.; Croshaw, J.; Mastel, K.; Tamura, M.; Hosseinzadeh, H.; Wolkow, R. A. *Machine Learning: Science and Technology* **2019**, *1*, 025001.

(40) Celotta, R. J.; Balakirsky, S. B.; Fein, A. P.; Hess, F. M.; Rutter, G. M.; Stroscio, J. A. *Review of Scientific Instruments* **2014**, *85*.

(41) El Sallab, A.; Abdou, M.; Perot, E.; Yogamani, S. *IS and T International Symposium on Electronic Imaging Science and Technology* **2017**, 70–76.

(42) Silver, D. et al. *Nature* **2017**, *550*, 354–359.

(43) Leinen, P.; Esders, M.; Schütt, K. T.; Wagner, C.; Müller, K. R.; Stefan Tautz, F. *Science Advances* **2020**, *6*.

(44) Chen, I. J.; Aapro, M.; Kipnis, A.; Ilin, A.; Liljeroth, P.; Foster, A. S. *Nature Communications 2022 13:1* **2022**, *13*, 1–8.

(45) Sotres, J.; Boyd, H.; Gonzalez-Martinez, J. F. *Nanoscale* **2021**, *13*, 9193–9203.

(46) Huang, B.; Li, Z.; Li, J. *Nanoscale* **2018**, *10*, 21320–21326.

3 Experimental and Computational Methods

This chapter contains an overview of the apparatus used to carry out scanning tunnelling microscopy (STM) experiments for this thesis. Specifics on substrate preparation and depositions are discussed, as well as a brief overview of the computational methods.

3.1 Equipment

The data presented in this thesis were acquired using two ultra-high vacuum (UHV) scanning probe microscopy (SPM) systems: The room temperature data (silicon based systems) used a Scienta Omicron variable temperature (VT) combined STM/NC-AFM (Figure 3.1a), and the low temperature (LT) data (metallic systems) used a third generation Scienta Omicron LT STM (Figure 3.1b), both controlled using a RC5 Nanonis controller, under UHV conditions. The VT system scan head temperature was regulated using a Lakeshore temperature controller 331, connected to a temperature detector mounted in the scan head, where a built in heater allowed for temperature regulation using a PID loop controller. The temperature of the scan head was held just above 293 K (up to 10 °C) throughout the experiments discussed in this thesis. The temperature of the scan head in the LT system was held at $\approx$ 4.5 K, cooled with liquid Helium.

The required surface cleanliness for atomic and molecular resolution imaging in STM necessitates pressures comfortably within the UHV vacuum threshold ($< 1 \times 10^{-9}$ mbar). The VT system achieves these pressures using a Agilent ion pump, with periodic sublimation of titanium. The bulk gas

pressure was removed using a TV 301 Navigator turbomolecular pump and an Edwards RV5 oil-seated rotary vane pump. On the LT system, these pressures were achieved using a Gamma Vacuum ion pump, with periodic titanium sublimation. Bulk gas within the chamber was pumped using a Pfieffer HiPace 300 turbomolecular pump, backed by an Edwards nXDS dry scroll pump. All system chambers also underwent three-day bakeouts which helps to remove water and other molecules adsorbed on the inside walls of the chambers, which would otherwise outgas slowly over time. During the experiments described throughout this thesis, the pressures typically ranged between 1×10^{-10} mbar and 5×10^{-11} mbar for the VT system, and 2×10^{-11} mbar and 5×10^{-12} mbar for the LT.

Both chambers were equipped with load-locks for the transfer of samples and tips into and out of the chamber, sputter guns for metal sample cleaning (Omicron ISE 5 for the VT and IS40 on the LT), and adjustable manipulator arms for sample annealing (and cooling in the case of the LT system) and transferring between the preparation and main chambers.

3.1.1 Substrates and Preparation

3.1.1.1 Si$(111) - 7 \times 7$

The Si(111) surface is a prototypical system in surface science, which, due to it well studied structure, is commonly used as a benchmark for testing various techniques in STM. In addition to this, the surface is quick and simple to prepare in UHV. This means makes it an ideal candidate for testing automation scripts.

The general appearance of the Si(111) - 7×7 surface is shown in Figure 3.2a. When imaged with an ideal tip, the surface appears with a unit cell containing 12 silicon surface atoms, with depressions at each of the four corners of the unit cell, known as corner holes.

Silicon is very reactive and so readily oxidises in air, meaning that surfaces have to be well prepared before use and all imaging must be carried out under UHV conditions. Prior to being introduced into the chamber, n-type Si(111) crystals ($0.001 - 0.005\ \Omega$cm) were sonicated sequentially in solutions

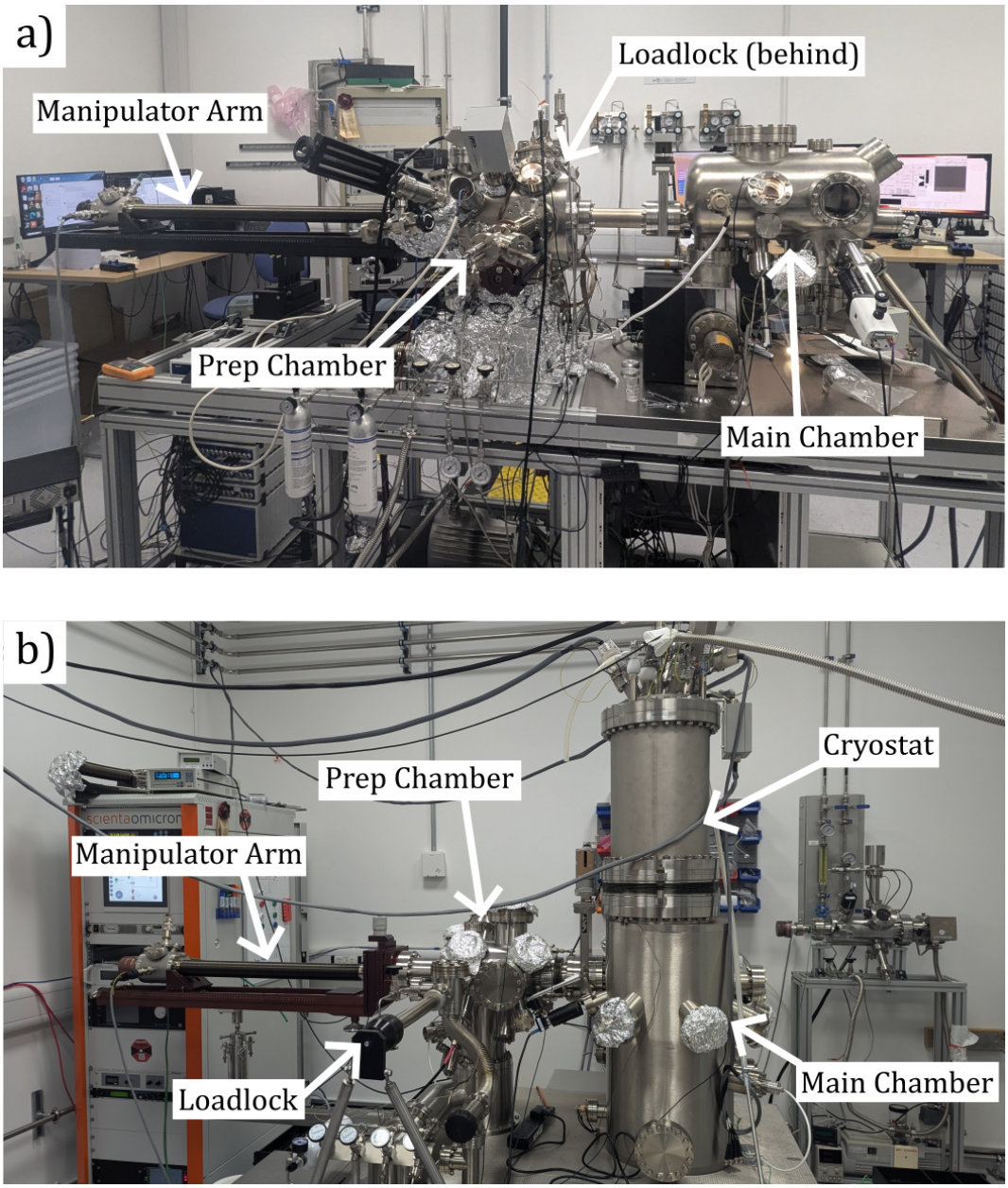


Figure 3.1: a) Scienta Omicron VT combined STM/NC-AFM. b) 3rd generation Scienta Omicron LT AFM. Labelled on both are the main and preparation chambers, load-locks (on the opposite side of the chamber for the VT system) and manipulator arms. Also shown is the cryostat for the LT main chamber, containing the cryogenic liquids used to cool the scan head.

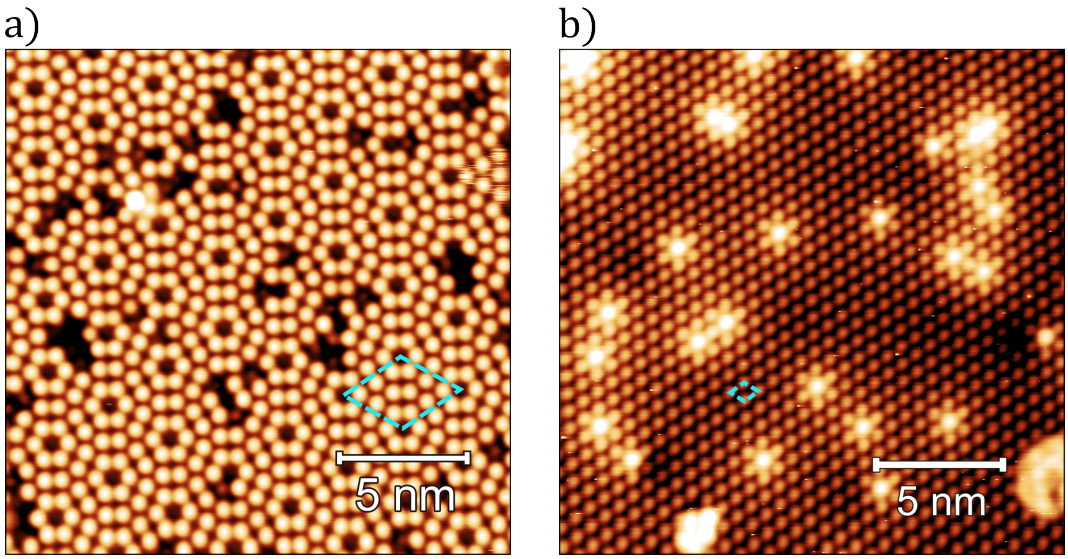


Figure 3.2: Two Si(111) surfaces imaged in STM at room temperature. a) The 7×7 reconstruction at 200 pA, 2 V. b) B:Si - $(\sqrt{3} \times \sqrt{3})R30°$ at 250 pA, 2 V. The unit cells for each substrate are shown in dashed blue.

of deionised water, methanol, acetone and isopropanol for $\sim$ 5 minutes each. Wafers are then introduced into the loadlock, where they are pumped by a turbomolecular pump, and baked at 150 °C before introducing into the chamber. Once in the chamber, the crystals are initially degassed via direct heating of the crystal at 500 °C overnight, before removing the remaining oxide layer by flash annealing the sample to 1200 °C, repeating until the sample can be held at temperature for a full 10 seconds without the chamber pressure exceeding 2×10^{-9} mbar. On the final flash, the sample is cooled quickly to 800 °C, followed by a slow cool down back to room temperature over $\sim$ 10 minutes. This process allows the Si(111) crystal to relax into the ground state 7×7 reconstruction.

Silicon crystals are bought pre-cut along the (111) plane. Cleaving along the (111) plane results in surface silicon atoms with fewer neighboring atoms than is energetically favourable (four in bulk silicon). On initial cleaving, the surface forms the meta stable 2×1 configuration, a state with a high density of unsaturated dangling bonds, which is sub-optimal. On annealing to 800 °C the top layer of the crystal is able to re-orientate itself to minimise the

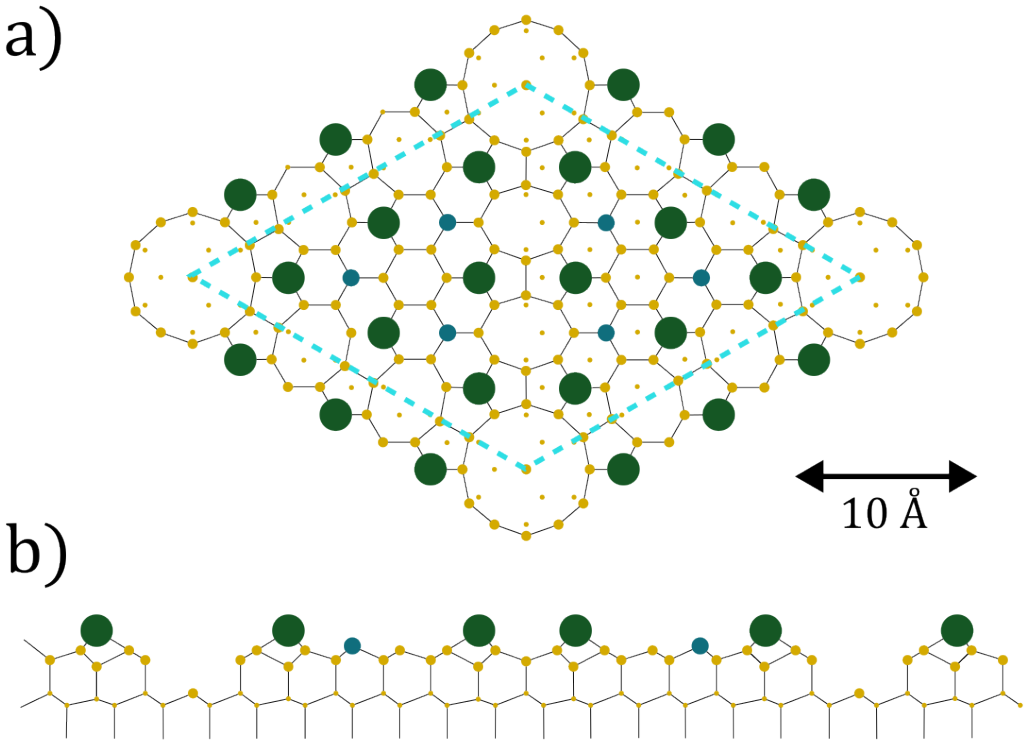


Figure 3.3: a) Top down ball-and-stick model of the Si(111) - 7 × 7 surface showing the 12 surface adatoms visible in the unit cell. b) Side view ball-and-stick model of the same surface. Adatoms and rest atoms are shown by green and blue spheres respectively. Yellow spheres of descending size show silicon atoms in lower layers. The dashed blue line indicated the unit cell of the 7 × 7 reconstruction (see Figure 3.2b for the same unit cell in STM.)

energy, reducing the number of dangling bonds in the unit cell from 49 to 19, including 12 of the adatoms (shown in green in Figure 3.3), 6 rest atoms in the second layer (blue in Figure 3.3) and 1 in the corner hole, resulting in the 7 × 7 reconstruction.

The model shown in Figure 3.3 is the dimer-adatom-stacking fault (DAS) model [1]. This model explains the stacking fault present on one half of the unit cell, called the faulted half (left side in Figure 3.3a), in contrast to the un-faulted half (right hand side of the unit cell).

The most common defect visible on the Si(111) - 7 × 7 surface are adatom

vacancies, which appear are depressions in STM image where a silicon atom would typically be located. These defects occur when silicon atoms do not fully relax in the ground state during surface preparation. Water contamination is also a common occurrence due to the high reactivity of the surface, and the persistent low background level of water in the vacuum chamber. Water on the surface also appears as depressions in STM images. These defects become more prevalent over time, even under good vacuum conditions ($< 1 \times 10^{-10}$ mbar) due to the background pressure. Consequently, Si(111) - 7×7 crystals require re-preparation every three-to-four days.

3.1.1.2 B:Si$(111) - (\sqrt{3} \times \sqrt{3})R30^{\circ}$

The method of preparation used for the B:Si(111) - $(\sqrt{3} \times \sqrt{3})R30^{\circ}$ [B:Si hereafter] was adapted from Spadafora *et al.* [2], in a protocol known to be most consistent in producing a high quality reconstruction with minimal defects. Heavily boron doped p-type Si(111) crystals were sonicated, baked in the loadlock and degassed in the UHV chamber overnight as described above, before being flash annealed (by direct heating) to 1200 °C for up to 10 seconds at a time, repeating until the full 10 seconds could be reached without the pressure exceeding 2×10^{-9} mbar. At which point the crystal is annealed at 900 °C for one hour, and then cooled to room temperature slowly at a rate of ~ 1 °C/s, this process draws boron dopants from the bulk crystal to the surface [3].

Figure 3.2b shows the boron induced $(\sqrt{3} \times \sqrt{3})R30^{\circ}$ reconstruction of the Si(111) surface. Through the extended annealing process described above, the boron atoms within the bulk crystal segregate to the surface of the crystal. On reaching the surface, the boron atoms occupy the substitutional (S_5) sites [4], below the surface silicon atoms in the T_4 sites, as shown in Figure 3.4. Because of this positioning of the boron atoms, the sp^3 dangling bonds of the surface silicon atoms become saturated through charge transfer (from the silicon to the boron atoms), resulting in a much lower reactivity than the clean Si(111) - 7×7 surface [5].

Defects on the B:Si surface arise from incomplete boron substitution into the S_5 sites, resulting in the sp^3 dangling bond being present, which appear

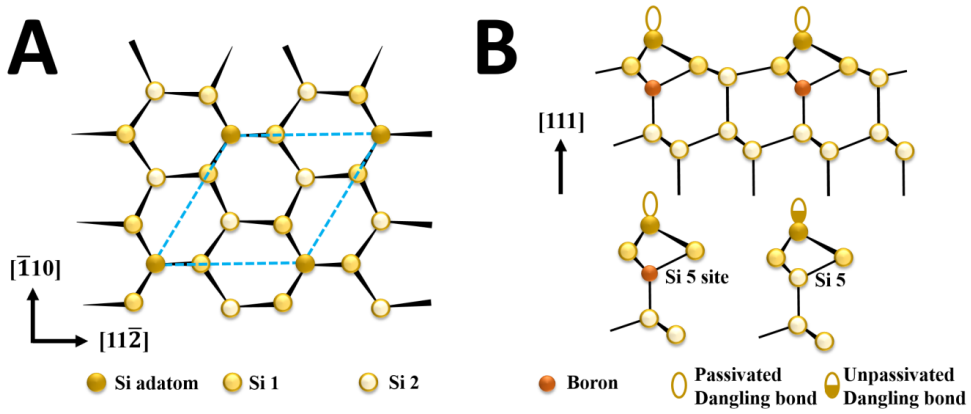


Figure 3.4: A) Top down ball-and-stick model of B:Si(111). The repeating unit cell is shown by the blue dashed line (see Figure 3.2b for the unit cell in STM). B) Side view ball-and-stick models of the complete substitution (top) and a depiction of the complete and incomplete substitutions of the S_5 site (bottom), the latter resulting in the surface defect shown in Figure 3.5a. Image from Brown *et al.* [6].

as bright protrusions due to the higher electron density (Figure 3.5a.) Additional defects on the B:Si surface also shown in Figure 3.5 are: b) Silicon atoms residing in S_5 sites below vacancies in the T_4 site, c) boron atoms in the S_5 site below vacancies in the T_4 site, and d) boron atoms occupying adsorbs between three boron atoms in S_5 sites [7].

3.1.1.3 Cu(111) and Au(111)

Copper and gold are widely studied prototypical metal substrates, whose ease of preparation and well known structures make them a valuable surface for atomic manipulation and the study of organic molecules. This makes both Cu(111) and Au(111) good candidates for studying the automation of tip preparation for surfaces with adsorbed atoms and molecules. Both substrates are prepared in the same way, where repeated sputter-anneal cycles are applied, repeating until the surface appears clean in STM. Sputtering involves introducing a pressure of $\sim 5 \times 10^{-5}$ mbar of Ar into the chamber, with a beam energy of 1.5 kV, directed at the sample in the manipulator

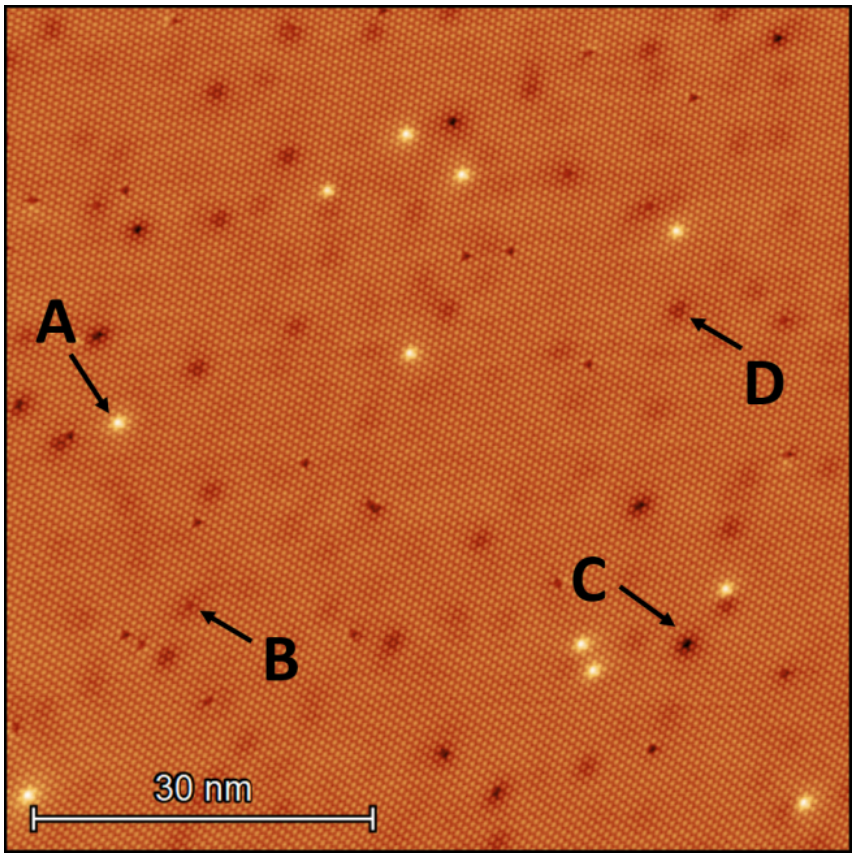


Figure 3.5: Overview STM image of the B:Si surface, images at room temperature, 2 V, 20 pA. The overview shows various common defects on the B:Si substrate: A) Si(S_5) atoms underneath silicon atoms. B) Si(S_5) atoms underneath vacancies. C) Boron atoms occupying S_5 sites underneath vacancies. D) Boron atom occupying sites between three B(S_5) atoms. Image from Brown *et al.* [6].

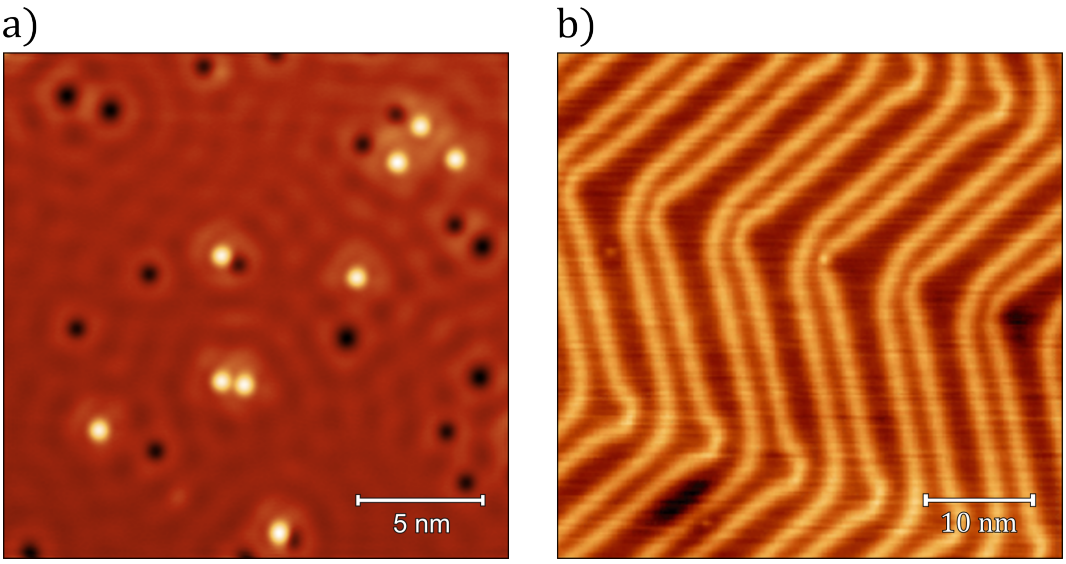


Figure 3.6: Two coinage metal surfaces imaged in STM at low temperature. a) The Cu(111) surface with a low coverage of Cu adatoms (bright protrusions) and CO molecules (depressions), imaged at 100 pA, 100 mV. b) The Au(111) surface showing the characteristic herringbone structure, image from Cogswell *et al.* [8].

arm (shown in Figure 3.8a) for 30 minutes. After sputtering, the sample is annealed at a temperature of 500 °C for 30 minutes via resistive heating in the manipulator arm. Once prepared, the surfaces appear as shown in Figure 3.6.

Both the Cu(111) and Au(111) surfaces have a face-centred cubic crystal (fcc) structure, which leads to similar adsorption sites on both. The specific adsorption sites for a fcc surface are shown in Figure 3.7. The top sites are where an adsorbate sits directly above a surface atom, typically resulting in weaker binding due to the lower coordination number. Bridge site adsorptions occur where an adsorbate sits above a pair of neighbouring atoms, resulting in a slightly stronger binding than on top sites. Hollow sites are characterised by a generally higher adsorption strength, where an adsorbate sits above the intersection of three surface atoms, either with a second layer atom directly below (fcc hollow site) or without (hexagonal close packing (hcp) hollow site).

The atomic structure of the surface is not visible in either of Figures 3.6a or

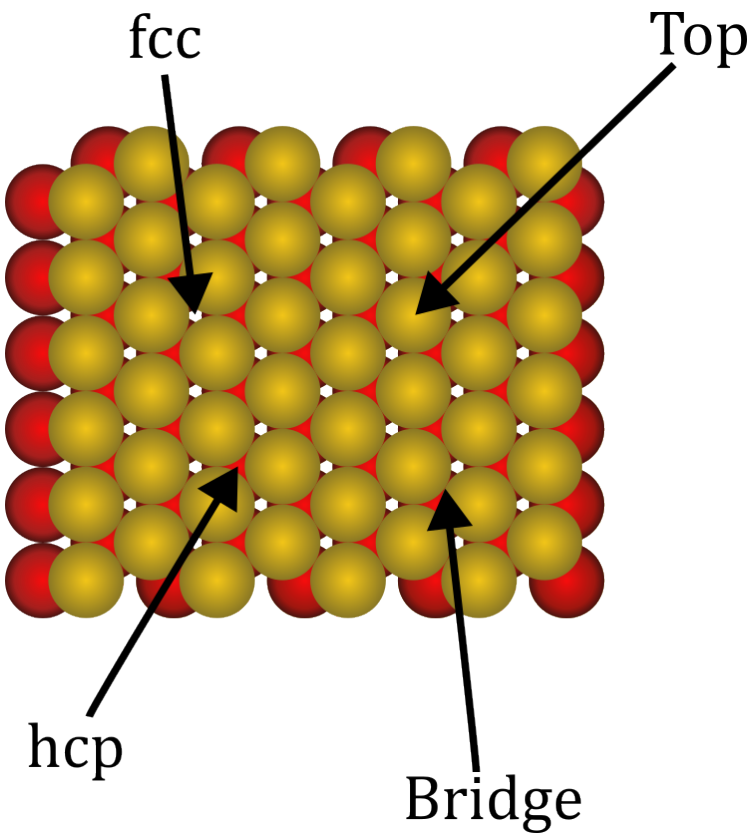


Figure 3.7: Top down ball-and-stick model showing the first two layers of a fcc crystal structure showing the possible adsorption sites. The two hollow sites, fcc and hcp, are dependent on the absence or presence of an atom below to hollow site respectively.

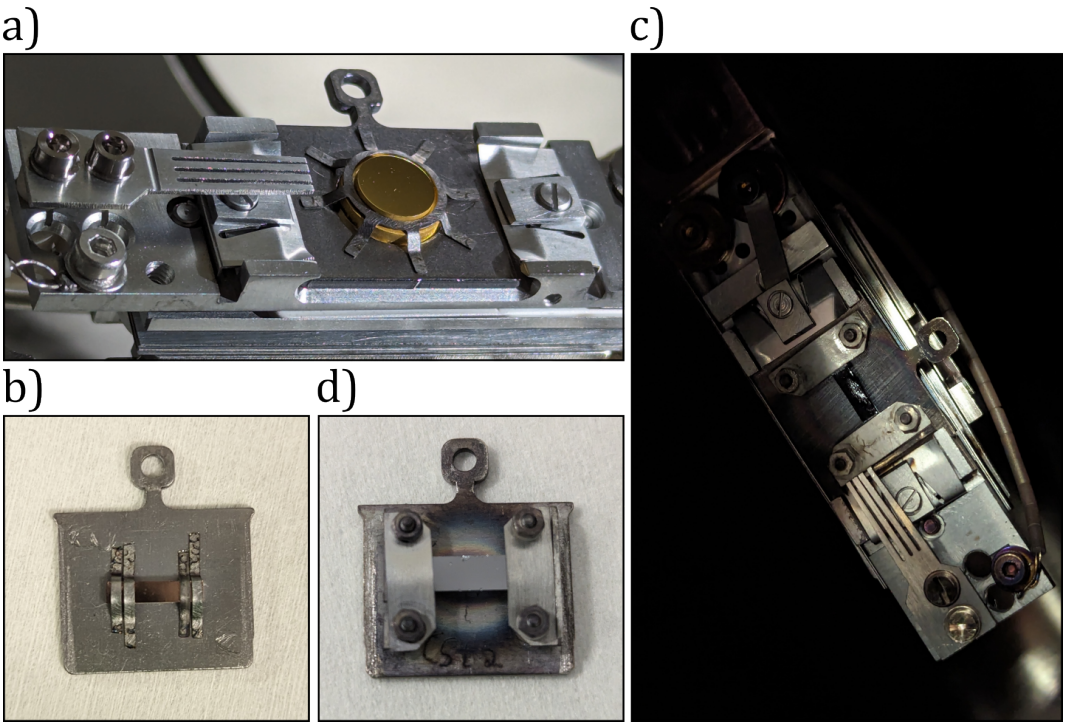


Figure 3.8: a) Au(111) sample in the manipulator arm of the LT system. b) Cu(111) sample mounted on a metal sample plate. c) Si(111) sample mounted on a sample plate. b) Si(111) sample plate in the manipulator arm of the VT system.

b due to the imaging parameters. The Cu(111) surface (Figure 3.6a) shows only a standing wave pattern due to free electrons on the surface, the pattern of which can be disturbed by adsorbates on the surface. The Au(111) - $22 \times \sqrt{3}$ surface (Figure 3.6b) shows what is know as the herringbone reconstruction. This is a unique reconstruction observed on Au(111), which arises due to surface stress and the desire to minimise the surface energy. This reconstruction appears as a slight rearrangement of the surface into a series of peaks and troughs, consisting of alternating fcc and hcp stacking regions. This reconstruction creates a more complex landscape of adsorption sites, where adsorbates preferentially aggregate along the domain boundaries and "elbows" of the herringbone.

3.1.2 Atomic and Molecular Deposition

Two selections of adsorbates on the Cu(111) surface are considered within this thesis, the first has a low coverage of Cu adatoms and CO molecules deposited, and the second has the addition of C_{60} molecules. For the Au(111) surface, a low coverage of Tin Phthalocyanine (SnPc) was prepared.

A low coverage of CO on Cu(111) was achieved by leaking CO (to a pressure around 10^{-8} mbar) into the chamber and opening the cryostat shields for 30 s whilst the sample was kept below 10 K in the head. Cu and C_{60} were deposited by direct sublimation into the scan head from a Focus EFM3T evaporator (Figures 3.9c-d), during which the sample was held at $\sim$ 5 K. When depositing via an electron beam flux monitor (EFM) evaporator, the source material is deposited by heating through bombardment of electrons from a filament. The source material is biased to accelerate the electrons toward it, which heats the source, causing the material to sublime. The adjustable parameters in such a deposition source are the bias applied to the source and the filament current, where the filament current is varied to achieve a target emission current from the source. For the Cu deposition, the source was a copper rod, which was biased at 800 V, aiming for an emission current of 9 mA. Once the emission current was reached, the cryostat shields were opened to allow for direct line of sight onto the sample for 1 minute, before closing the shields and checking the surface in STM. The C_{60} deposition source was a Mo crucible containing C_{60} powder, which was biased at 300 V, aiming for an emission current of $\approx$ 3 nA. At the correct emission current, the cryostat shields were opened for 10 minutes, before closing and again checking the coverage in STM.

On the Cu(111) surface, the CO molecules adsorb preferentially over top sites of the Cu(111) surface, whereas Cu adatoms adsorb over fcc hollow sites, as shown in Figure 3.6a. The C_{60} molecules, however, have been found to adsorb in a variety of different geometries. These different orientations are characterised by the different sections of the C_{60} cage pointing upward; these include: hexagon up, pentagon up, and C=C or C-C bond up. The specific bonding site of C_{60} on Cu(111) depends on the specific geometry of each individual C_{60} molecule. Typically, the C_{60} molecules adsorb with a hexagon of carbon atoms facing up, where they bond to fcc hollow sites on

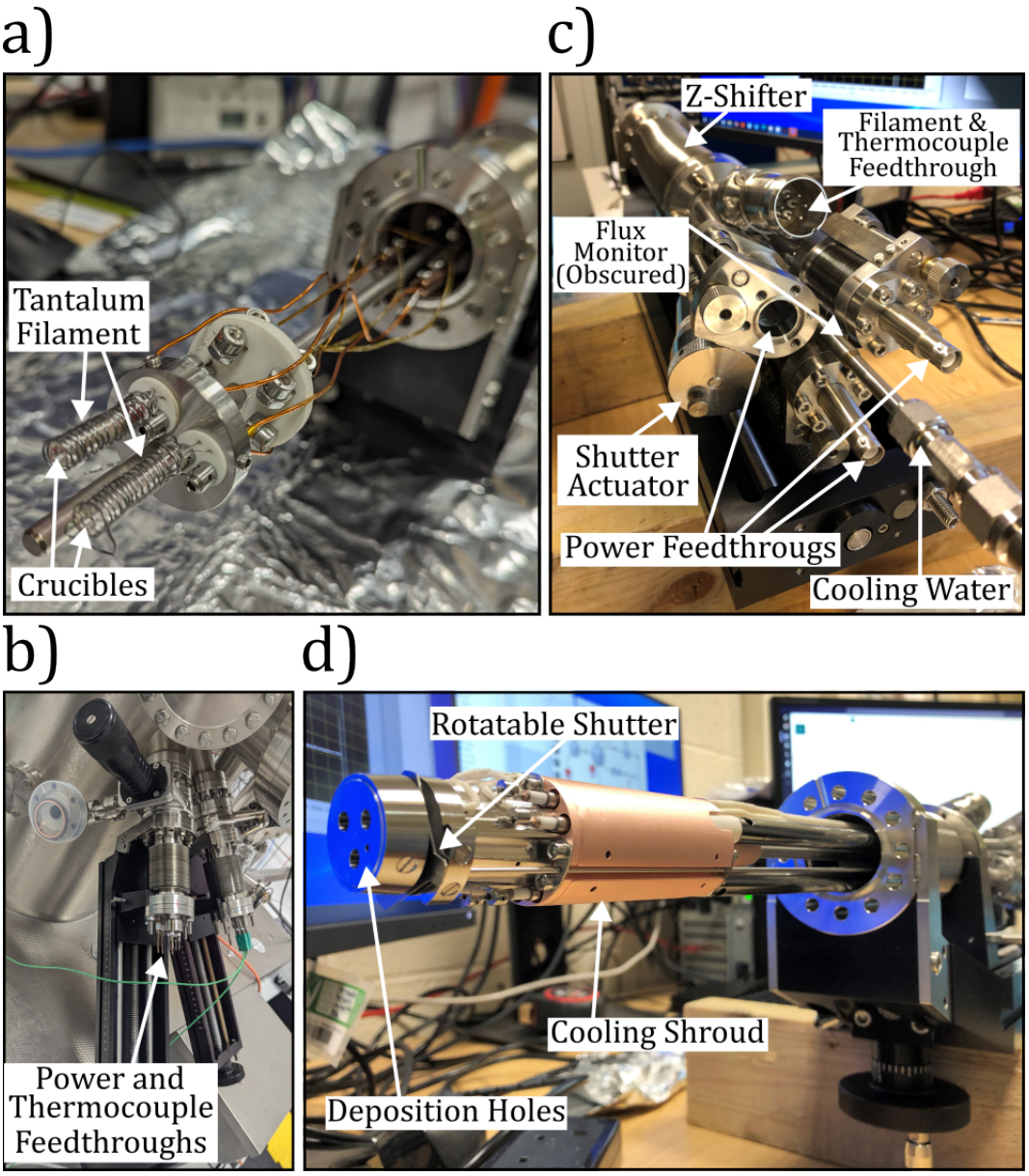


Figure 3.9: a)-b) Home-built double evaporator. c)-d) EFM. Both mounted on a Z-shifter, and mounted on the vacuum chambers behind a valve so it can be pumped down separate from the system.

the Cu(111) surface.

SnPc was deposited onto the Au(111) surface using a home-built evaporator (Figure 3.9a-b), where the powdered source material is contained within a glass crucible using glass wool, around which a coil of tantalum wire is wound, providing a source of heat for the crucible. By passing a current through the coil (through copper feedthroughs), the crucible is heated up to a target deposition temperature, measured through a thermocouple embedded into the bottom of the glass crucible itself. These evaporators are designed to be pumped down and baked behind a gate valve, separate from the rest of the chamber. Once under vacuum, the crucibles are degassed behind the gate valve by heating to slightly above the deposition temperature, before being introduced into the main chamber and degassed once more. The target temperature for SnPc deposition was 360 °C, once reached, the cryostat shields were opened for 1 hour, before closing and checking the coverage in STM. Once deposited, the sample was cold annealed to room temperature, which has the effect of driving the molecules preferentially to the "elbow" sites of the herringbone structure.

3.2 Computational Methods

3.2.1 Image Processing

Typically, before input into any automation scripts, due to the effects described in Section 2.1.2.3, scans require some level of processing in order for the important features within the image to be more apparent, specifically, the images are flattened to reduce effect of distortions in the z-direction arising from thermal drift. An example of post and pre-flattening is shown in Figure 3.10.

Flattening of all input images was carried out automatically in Python by row-wise median subtraction, where the median height of each row is calculated, and subtracted from the height at each pixel. This method assumes that the median values of each row should be the same height, and centres each row based on this assumption. This method of flattening works

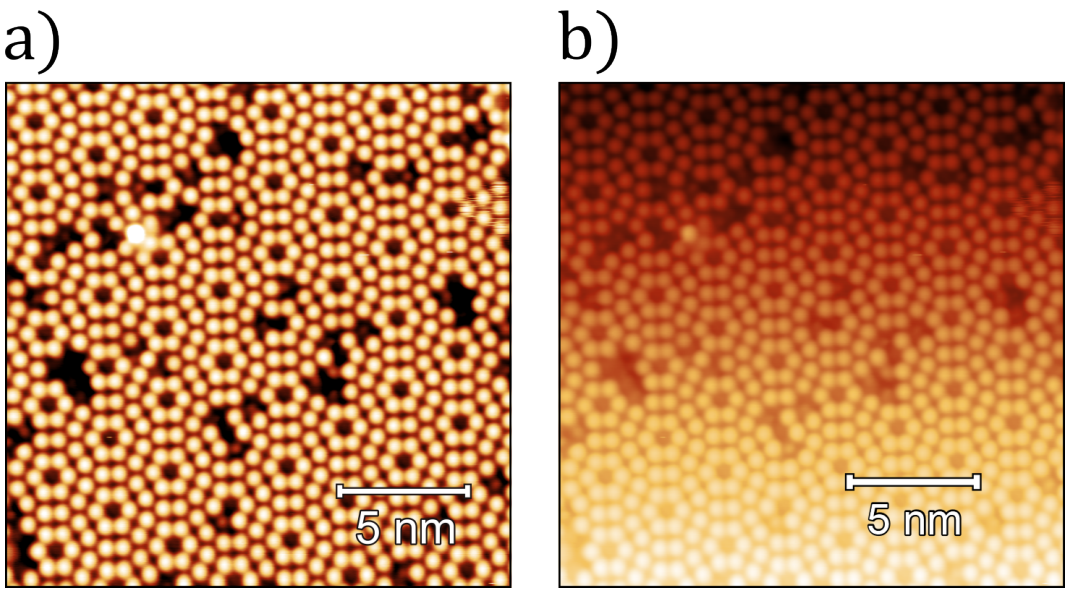


Figure 3.10: Si(111) - 7 × 7 surface, imaged in STM at 2 V, 200 pA at room temperature a) after plane flattening, and b) before plane flattening.

well when applied to images containing no steps, and where the majority of the slope is in the slow scan direction, however if either of these conditions are not met, other flattening methods should be used.

Once flattened, the scans are also thresholded between a set of heights depending on the features which need to be focused on. For a bare, relatively clean surface, where the surface atoms are the features of interest (such as on the silicon based systems presented), automatic mean thresholding was applied. Here, scans are thresholded around a fraction above and below the mean height of the image. This works well to stop large defects (commonly protrusions) from washing out the image, obscuring the majority of the atomic detail within the scan.

For scans containing larger molecules, rather than a simple fraction above and below the mean height, physical distances above and below the mean height were input for thresholding. This focuses the colour scale onto the features of interest, the molecules. An example of this is shown in Figure 3.11, where before thresholding very little detail on the C_{60} molecules can be seen, and after thresholding the shape of the lobes can be observed in

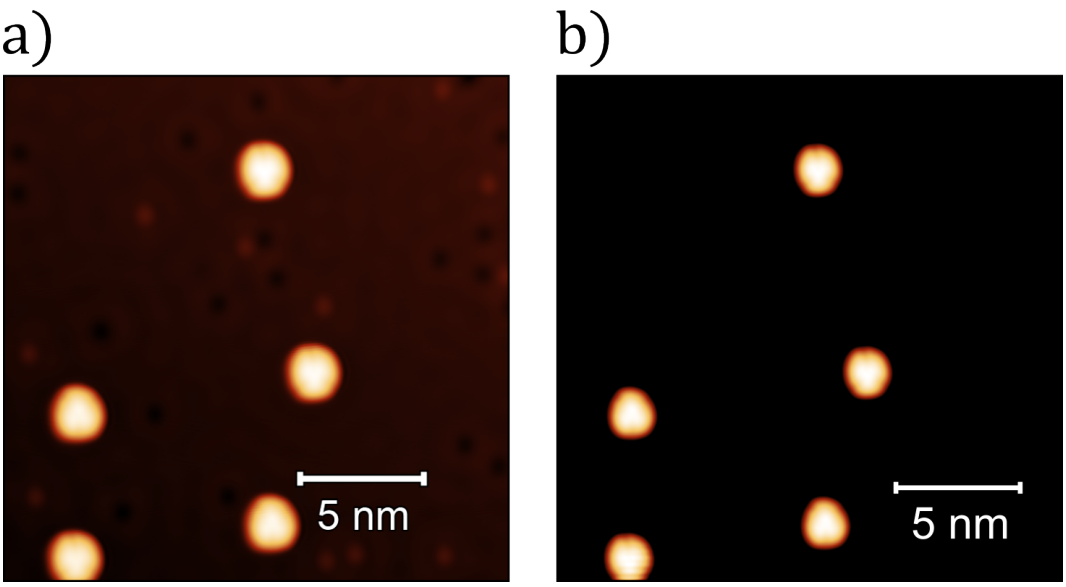


Figure 3.11: Cu(111) surface with a low coverage of C_{60}, CO and Cu, imaged in STM at 100 mV, 100 pA at 5 K before and after thresholding in a) and b) respectively.

more detail.

3.2.2 Python Scripting

Python was used throughout this project for all analysis, image extraction and training of machine learning networks. Final scripts discussed in this thesis run on Python version 3.9.13. Various common python libraries were used in the project, most importantly: numpy, pandas, scipy, TensorFlow and SPIEPy. All machine learning models were trained using TensorFlow 2.10.0, with models trained on an Intel Core i7-8750H CPU, utilising an NVIDIA GeForce GTX 1060 GPU with 16 GB of DDR4 RAM.

3.2.2.1 SPIEPy

One important, less well known, library which was useful in this thesis (specifically in Chapter 4) in the SPIEPy library, a Python port of the SPIW

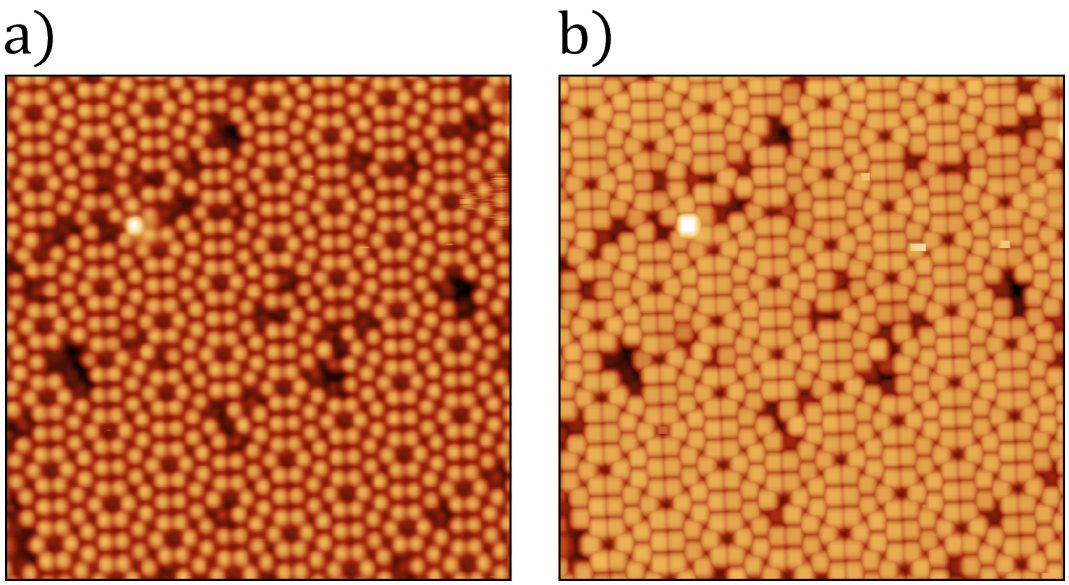


Figure 3.12: Si(111) - 7 × 7 surface, imaged in STM at 2 V, 200 pA at room temperature a) before, and b) after applying the numpy max filtering function with a footprint of 10 pixels.

Matlab library [9]. This library includes a number of functions, the most prominently used of which is the peak finding function.

This function applies the numpy maximum filter function, which takes in two arguments: an input array and a footprint size, and finds the maxima within an input array, setting all pixels in a square footprint around it to that value. The footprint size also determines how close together the maxima can be for this calculation. An example of this function being applied to the Si(111) - 7 × 7 surface is shown in Figure 3.12.

From this maximum filtered image, the points where the original input and the maxima image contain the same pixel value correspond to the peaks in the input (the centres of the maxima in the maxima image).

3.2.3 LabVIEW Scripting

As mentioned previously, the SPM equipment used throughout this thesis are controlled by the Nanonis operating system. The front-end of this is written in LabVIEW, a graphical programming environment which allows for simple production of graphical user interfaces. This being written in LabVIEW means that interfacing with it through custom LabVIEW scripts, in order to directly control the STM, is convenient.

For this reason, all automation scripts used throughout this thesis were written in LabVIEW 2021, also interfacing with Python for specific pieces of analysis and classification, as discussed in more detail in the following sections.

References

(1) Takayanagi, K.; Tanishiro, Y.; Takahashi, M.; Takahashi, S. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **1985**, *3*, 1502–1506.

(2) Spadafora, E. J.; Berger, J.; Mutombo, P.; Telychko, M.; Švec, M.; Majzik, Z.; McLean, A. B.; Jelínek, P. *Journal of Physical Chemistry C* **2014**, *118*, 15744–15753.

(3) Baris, B.; Jeannoutot, J.; Luzet, V.; Palmino, F.; Rochefort, A.; Chérioux, F. *ACS Nano* **2012**, *6*, 6905–6911.

(4) Baumgärtel, P.; Paggel, J. J.; Hasselblatt, M.; Horn, K.; Fernandez, V.; Schaff, O.; Weaver, J. H.; Bradshaw, A. M.; Woodruff, D. P.; Rotenberg, E.; Denlinger, J. *Phys. Rev. B* **1999**, *59*, 13014–13019.

(5) Yates, J. T. *J. Phys. Condens. Matter* **1991**, *3*, 14–156.

(6) Brown, T. Intramolecular Force Mapping at Room Temperature, Ph.D. Thesis, 2022.

(7) Makoudi, Y.; Jeannoutot, J.; Palmino, F.; Chérioux, F.; Copie, G.; Krzeminski, C.; Cleri, F.; Grandidier, B. *Surface Science Reports* **2017**, *72*, 316–349.

(8) Cogswell, M.; Ahmed, F.; Girshick, R.; Zitnick, L.; Batra, D. *4th International Conference on Learning Representations* **2016**.

(9) Stirling, J.; Woolley, R. A.; Moriarty, P. *Review of Scientific Instruments* **2013**, *84*, 113701.

4 Fourier Filtering Based Feature Finding

Whilst the primary purpose of the scanning tunnelling microscope (STM) is atomic resolution imaging, it was soon noted that STM also had the capacity to make changes to the surface structure due to the proximity of the tip whilst imaging [1]. It therefore quickly developed into a method where one is able to intentionally manipulate single atoms and molecules on a surface [2]. This lead to the ability to create artificial structures and patterns with an array of applications in the study of quantum systems [2], data storage [3–5] and nano-electronics [6–8]. Nonetheless, the manual manipulation of adsorbates into a predetermined pattern is a slow and laborious undertaking, even for relatively small structures. The movie "A Boy and His Atom" consisted of 242 individual frames (one of which is shown in Figure 4.1a), containing 65 CO molecules, combined to make a stop motion film. The making of this film manually, took a total of 252 hours to complete [9]. This amount of time being spent on manual manipulation is not feasible for larger scale manipulations, hence, a full, reliable automation procedure would increase efficiency drastically.

Automated manipulation to construct patterns from adsorbates on a surface has been attempted many times, both utilising machine learning (ML) [11, 12] and deterministic techniques [10, 13, 14]. Commonly in manipulation at this scale, adsorbates will be moved on the surface via lateral manipulation [15], where the tip interacts with the atom/molecule, either pushing or pulling it along the allowed atomic sites. Manipulation via pulling is generally preferred as it is much more reliable than pushing; on close-packed surfaces with multiple symmetry equivalent directions, adsorbates can be pushed to the side of the tip, which is not the case when pulling [15]. For automated manipulation, it is critical for the script to have knowledge of

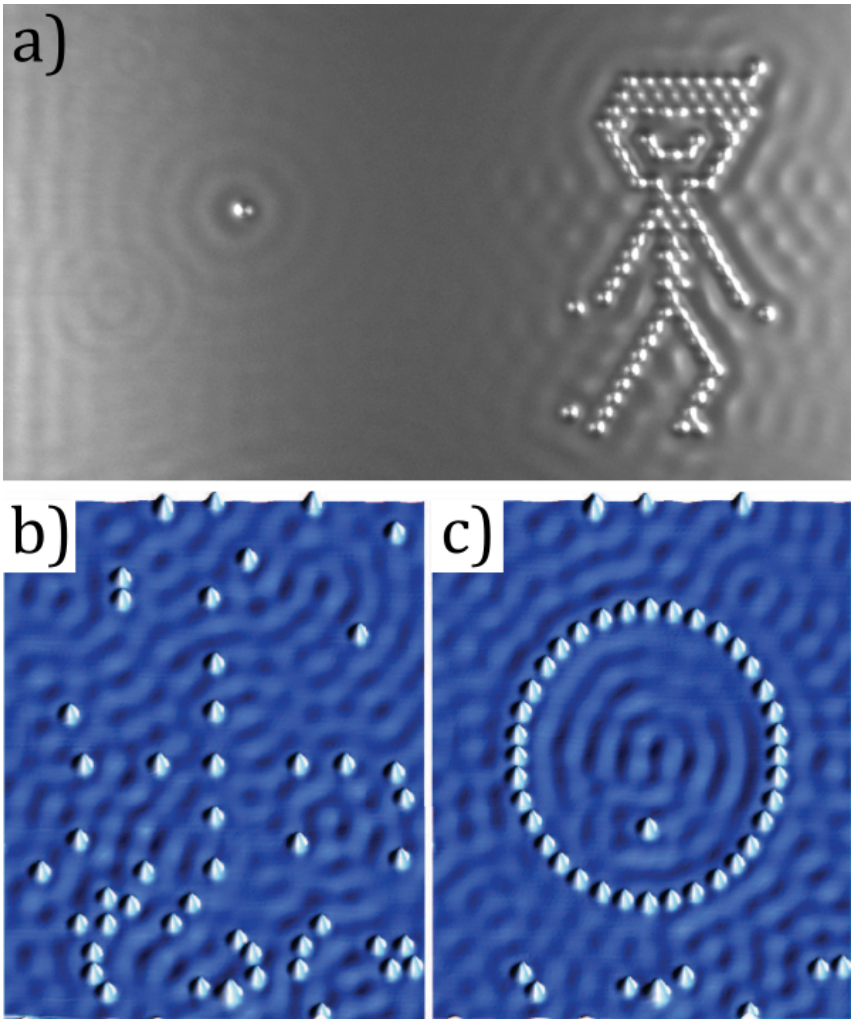


Figure 4.1: a) Frame from "A Boy and His Atom", a short film produced by IBM through manual manipulation of CO molecules on a copper substrate [9]. b)-c) Autonomous assembly of Co atoms into a quantum corral on the Cu(111) surface. a) Initial arrangement of Co atoms in the 20×25 nm scan frame. b) Image after assembly, taking a total of 66 minutes. Images from Celotta *et al.* [10].

the feature space over which manipulation events are possible, meaning knowledge of the atomic sites (both allowed and not-allowed). Without this knowledge, and careful planning, it is possible for mistakes to be made; e.g., incorrect placement of atoms/molecules, or irreversible clusters formation if adsorbates are moved too close to one another. The pattern shown in Figure 4.1c took a total of 66 minutes to assemble autonomously without mistakes, which, whilst this is possibly more time than would be needed by an experienced operator to do this manually, an automated script does not need to take breaks in it's operation, and so is far more efficient.

The manual manipulation procedure begins by taking an initial overview scan to locate the positions of atoms on the surface. From here, a fixed grid overlay of the surface structure can be applied to show the atomic positions throughout the rest of the surface. Each individual manipulation can then be planned based on the location of atoms/molecules in an area, working around defects and steps edges. This process requires a judicious pre-knowledge of the surface itself, including an operator to design the initial overlay, which additionally needs to be manually scaled depending on the scan size.

In this chapter, a method of identifying the atomic positions of the substrate and the automatic identification of defect sites and adsorbates with only minimal pre-knowledge of the surface structure is presented. This is accomplished via analysis of the reciprocal space of the STM image. Specific frequencies in Fourier space are isolated, producing an ideal representation of a specific scan, without the presence of any adsorbates. This method is able to map the positions of clear areas of a given surface, as well as surface atoms obscured by adsorbates appearing as protrusions or depressions in the topograph or missing surface atoms. These positions can then be passed on for further automated experimentation. This method is scale invariant, and completely automated for a given surface, resulting in an automated mapping of the surface. As an example, this method is applied to a live imaging tool, which is able to scan a surface, identify the positions of different features within the scan and automatically take spectroscopy measurements over selected sites of interest.

4.1 Reciprocal Space

Reciprocal space, also known as Fourier space, is a mathematical construct, commonly used in crystallography to describe the periodic structure of crystal lattices. Whereas real space describes the physical location of atoms within a crystal, reciprocal space shows the spatial frequencies of the crystal structure as a whole. The idea of reciprocal space is important in understanding different phenomena in condensed matter physics, specifically areas involving diffraction.

From a surface structure perspective, an understanding of reciprocal space is fundamental in analysis of diffraction patterns from surface-sensitive techniques such as low-energy electron diffraction (LEED), where the crystallographic structure of a surface can be characterised in a non-destructive manner [16, 17]. Here, a beam of low-energy electrons (typically $20 - 600$ eV) is directed at a surface, which interacts with the sample elastically, diffracting depending on the lattice spacing. The resulting interference pattern is imaged on a fluorescent screen, where high intensity diffraction spots can be observed, providing information on the reciprocal lattice of the surface. An example of such an interference pattern is shown in Figure 4.2b. Techniques such as LEED are relatively insensitive to surface defects, showing diffraction patterns based purely on the reciprocal nature of an "ideal" surface.

STM, in contrast, provides a real space mapping of a surface with atomic resolution, the periodic structure of which can be studied by utilising the fast Fourier transform (FFT) of this topograph, converting the spatial information from real space into reciprocal space. This FFT map is analogous to the features observed in a LEED experiment, as is shown in Figure 4.2. Both LEED and the FFT of an STM image reveal peaks in reciprocal space, corresponding to the periodic components of the surface, specifically representing the reciprocal lattice vectors of the surface. Distances in reciprocal space are related to real space via an inverse relationship:

$$Q = \frac{2\pi}{x} \tag{4.1}$$

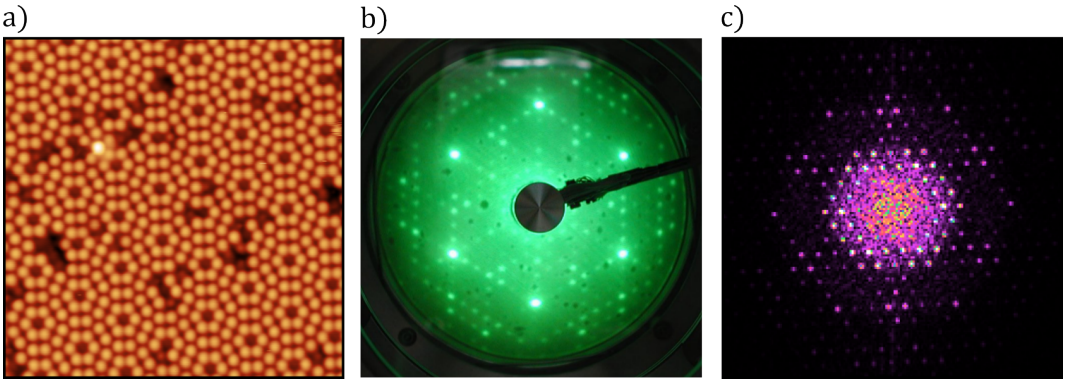


Figure 4.2: a) Si(111) - 7 × 7 surface, imaged by STM at 200 pA, 2 V at room temperature. b) Example of a diffraction pattern of the Si(111) - 7 × 7 surface from a LEED experiment obtained by Aversa *et al.* [17]. c) Fast Fourier transform of the image shown in a), showing the comparable positioning of peaks as in b).

Where Q is the distance in reciprocal space and x is the distance in real space. Images in real space are composed of various patterns, edges, textures and other features which contribute to the overall appearance of the image. Calculating the FFT of a real space image allows for the decomposition of the image into a sum of the sinusoidal patterns of varying frequencies, which correspond to different features and textures within an image. High spatial frequencies correspond to rapid changes of intensity in the input, such as edges or noise, whereas low spatial frequencies represent slow changes in intensities, such as small changes in intensity, or smooth areas within an image. FFTs are commonly used in image processing to filter noise within an image, which usually appears in the frequency domain as variations at well defined frequencies, which can be filtered out using simple low-pass, high-pass, or band-pass filters, depending on the type of noise present, removing these high frequency components from the image.

This means that for a simple grid-like arrangement of atoms in real space, the corresponding Fourier transform will produce peaks in reciprocal space at corresponding frequencies to the distances between features. This is shown pictorially in Figure 4.3, where the individual effect of different isolated peaks in the FFT are demonstrated. Isolating individual peaks results in sinusoidal features in the filtered image in different orientations, and isolating multiple peaks begins to create an overall periodic structure.

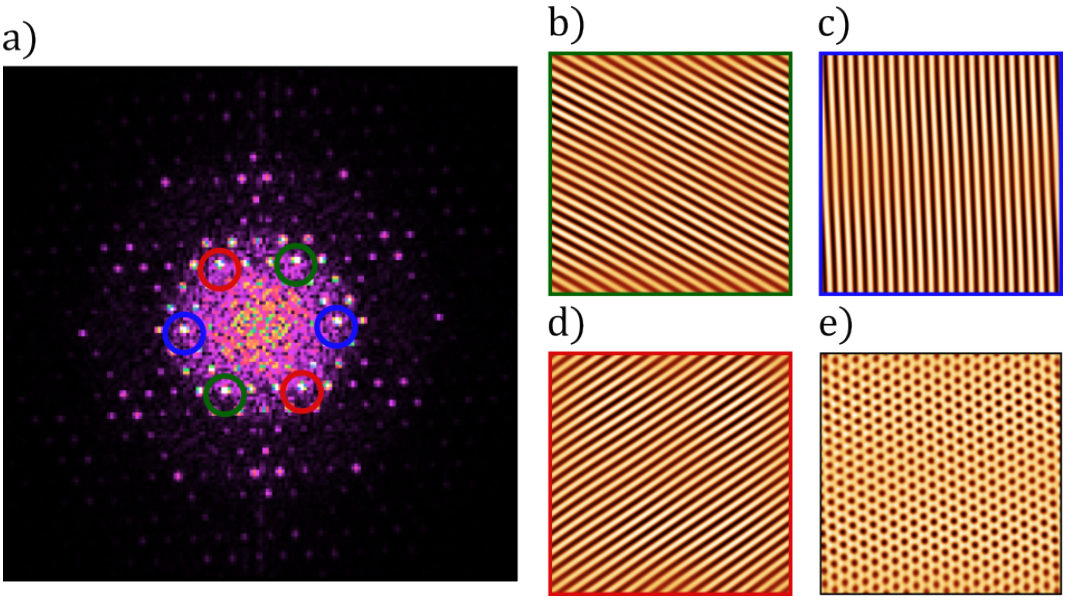


Figure 4.3: a) Fourier transform of Si(111) - 7×7. b)-e) Images resulting from filtering specific frequencies from a) and calculating the inverse Fourier transform. Peaks in a) highlighted in green, blue and red result in images b-c) respectively. e) Result of isolating all the peaks highlighted in a).

Not only is the periodic structure in an STM image visible in reciprocal space, but also, all of the information within the topograph, including the non-periodic information such as defects and adsorbates which are randomly scattered throughout. Features such as these cause diffuse scattering in reciprocal space, where rather than forming well-defined peaks with a high intensity, like those produced by the periodic lattice, the defects show as a diffuse "cloud" around the primary peaks due to disruptions in the long range order of the surface. Using this knowledge, it is possible to reconstruct only the periodic structure of the input topograph by isolating the sharp peaks in reciprocal space and taking the inverse Fourier transform, as will be discussed in the next section.

4.2 Surface Estimation

The automated mapping of a surface begins with an input topograph, as shown in Figure 4.4a, from which the two-dimensional FFT is calculated, resulting in the reciprocal space feature map shown in Figures 4.4b-c. Before input, all images are flattened as described in Section 3.2. As mentioned in Section 4.1, the bright features shown within this feature map correspond to the specific frequencies which make up the repeating units of the surface. The information containing other features in the input image, such as defects and adsorbates which do not repeat, will be present somewhere within the FFT feature map, however these features have a much lower intensity. Therefore, by isolating the specific peaks with the highest relative intensities, an estimation of the purely periodic parts of the surface can be found.

From here, a function within the Spiepy Python library, "locate troughs and peaks", was used. This function, which is described in Section 3.2.2, takes an image input, as well as the pixel distance between features, and locates the peaks present within the image. By inputting the FFT into this function, along with the distance between the peaks in frequency space, the high intensity frequencies within the FFT are accurately located. Once located, these peaks can be isolated, and the inverse of the filtered FFT calculated. This results in a representation of the surface from the input image without

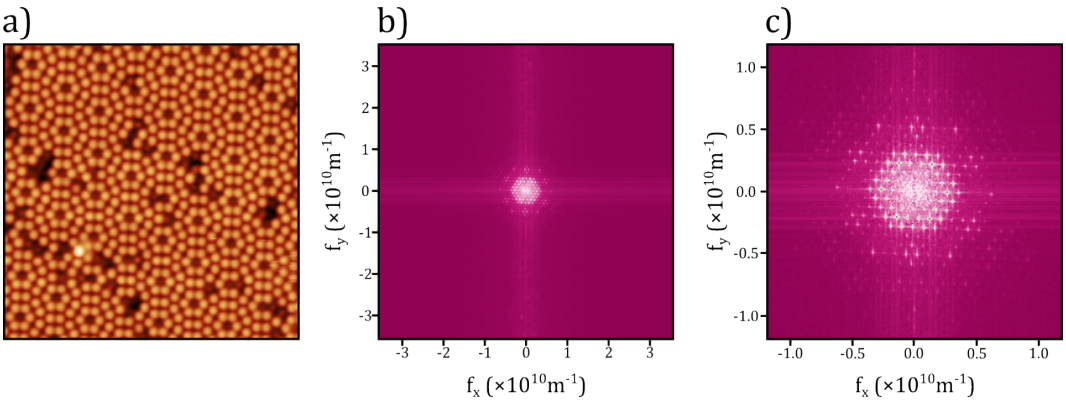


Figure 4.4: a) 20 × 20 nm STM image of Si(111) - 7 × 7 taken at room temperature, 2 V bias, 200 pA setpoint. b) The full range of the two-dimensional Fourier transform of a). c) Cropping of b) showing the important central frequency region.

the presence of defects or adsorbates, essentially an estimate of how an atomically perfect surface taken over the same area would appear.

Once isolated, the filtered FFT (Figure 4.5b) results in the estimated surface structure shown in Figure 4.5e. In addition to this, the sections of the FFT remaining after filtering can be used to find the features within the input image which were removed (image difference), shown in Figures 4.5c and f respectively. As can be seen through comparisons between these, the majority of the obvious, non-repeating structures on the surface have been accurately separated from the bare surface itself.

As mentioned previously, for the underlying surface to be estimated, the pixel-wise distance between features in Fourier space needs to be input into the function. The distance between peaks in frequency space is invariant between different scan sizes, and can be converted into a pixel-wise distance using the physical size of the scan and the pixel density of the image. The frequency range of the FFT can be calculated using Equation 4.1, and dividing this frequency range by the pixel density of the input scan results in the frequency resolution of the FFT (frequency range per pixel). The frequency resolution can then be used to calculate the pixel-wise distance by dividing the distance between peaks in frequency space by this resolution. Additionally, as the peaks are found by finding maxima within any area in the FFT, the method is invariant to rotations.The distance between peaks

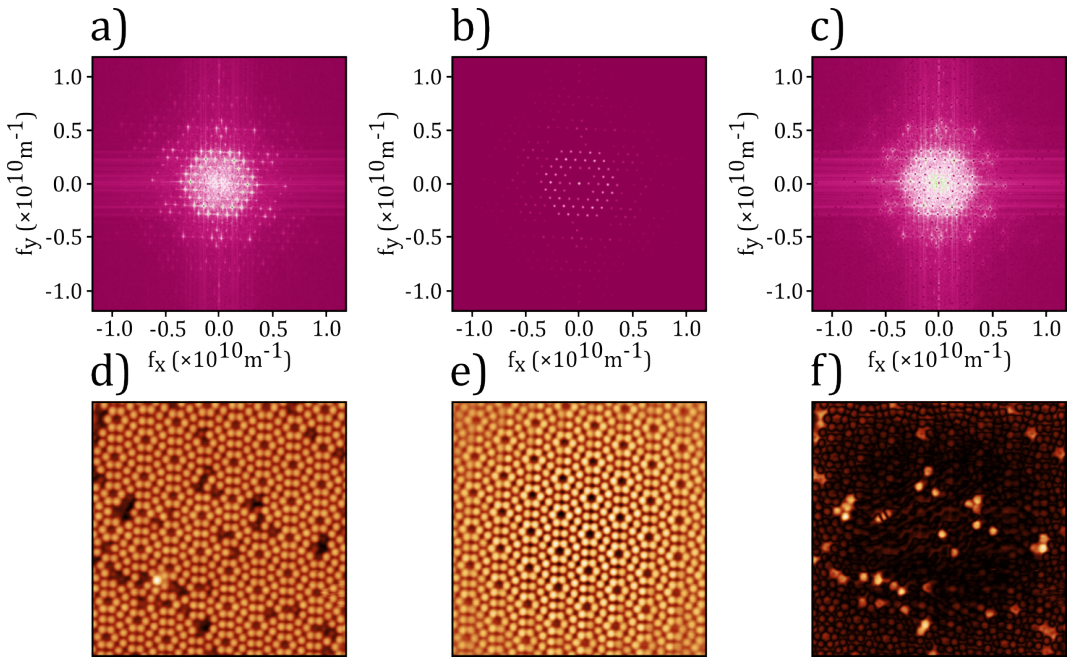


Figure 4.5: a) Original FFT before any filtering steps. b) Filtered FFT containing only the information from the repeating structure of the input image. c) FFT difference between a) and b), containing all information which is filtered out. The inverse Fourier transforms of a-c) result in the images shown in d-f) respectively.

only needs to be calculated once for each surface, after which the this method should work well at automatically estimating the ideal, clean surface representation of any input scan with enough of the repeating structure visible (see Section 4.3.3 for a detailed discussion on the limitations of this technique.)

Once the surface estimation has been obtained, the positions of both the obscured and clean surface atoms can be located. The positions of these atoms are found using the Spiepy function, "locate troughs and peaks". Here, the pixel-wise atomic spacing also needs to be input into the function to accurately find the peaks, which, once input, will work automatically on all images of the same surface by taking information directly from the scan file. These atomic positions are found for both the original input image, and the filtered image. By comparing the locations found in both, any atomic sites located on the filtered image, which are not present on the original image, are assumed to be positions where surface atoms have been obscured by defects or adsorbates. Obscured positions whose topographical height is greater than the mean peak height are labelled as protrusions, and those below the mean are labelled as depressions.

We note that before locating the atomic positions on the input image, it is masked at thresholds above and below the mean height of the image. This helps to avoid peaks being found due to adsorbate protrusions on the topograph which could result in spuriously found atomic positions. An example of this masking is shown in Figure 4.6, where the upper mask (above the mean) is shown in red and the lower (below the mean) is shown in blue. The final positions found on an example input image and filtered image are shown in Figure 4.7.

4.2.1 Fourier Artefacts

The filtered image (Figure 4.5e) shows clear blurring at the edges of the scan frame. This is a common artefact in Fourier filtering resulting from the to sharp discontinuities at the edges of the input image. When converted into Fourier space, these sharp discontinuities result in high frequency components in the frequency domain, which are then filtered out during

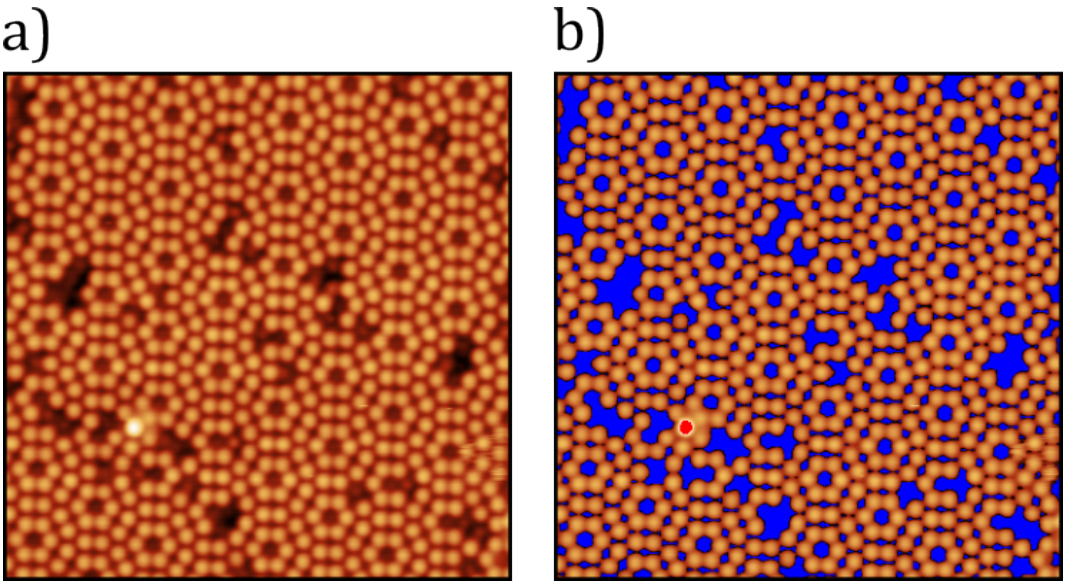


Figure 4.6: a) Original input image, which is masked, b), to determine the presence of protrusions and depressions. Higher mask is shown in red and the lower mask is shown in blue.

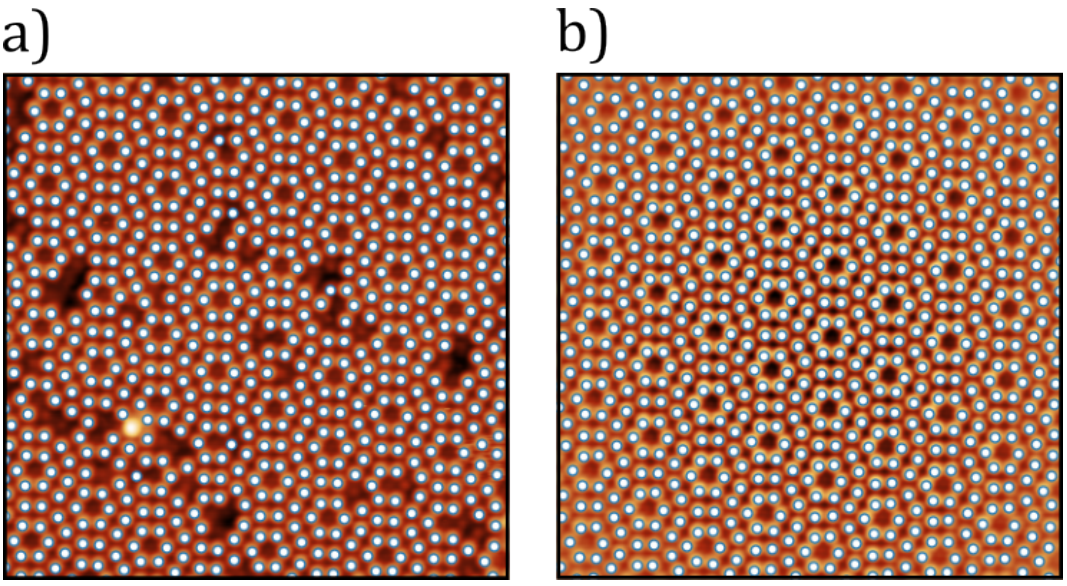


Figure 4.7: a) Input image with the initial points located on the surface plotted in white. b) filtered image with the estimated points plotted in white.

the filtering process. These discontinuities are present on all feature edges throughout the image, however, they are much sharper at the scan edges and so the resulting effects in the filtered image are much more evident at the scan edges. The effects of this can also be seen, to a much lesser extent, in more central parts of the filtered image, where the atoms appear somewhat blurred when compared to the input image.

To account for these effects, a number of pre-processing steps were tested. One common step to reduce the effect imaging artefacts is to pad the input image. Padding the edges of an input image is often used in frequency filtering to reduce the presence of the high frequency features within the FFT by softening the harsh boundaries. Image padding involves adding extra information to the edges of the image data, with the added information taking different forms, each having different overall effects on the final filtered image. A series of padding methods were tested, the most relevant of which were mean-padding, max-padding, tile-padding and reflection-padding. Examples of these padding methods are shown in Figure 4.9. As a point of reference, Figure 4.8 shows the result of the automated Fourier filtering method applied to the un-padded input. This filtered image estimates the repeating structure of the underlying surface well for central regions of the image, however toward the edges the atomic positions are less well defined and would not result in an accurate estimation of the real atomic positions.

The first of the padding methods tested was mean-padding, which involves padding the edges of the input with the mean of each row on the horizontal plane, and the mean of the columns on the vertical, as shown in Figure 4.9a and e. This method clearly improves the surface estimation, however still results in minor blurring toward the edges of the image frame. Second is max-padding, where the image is padded with the maximal pixel value from each row is used in the horizontal plane, and the maximum from each column in the vertical. The results of this padding are shown in Figure 4.9 b and f, where similarly to mean padding, the overall structural appearance of the image has been improved, however the corners of the image are now affected significantly, giving the reconstruction an obvious difference in the apparent height of the atoms. Next, is the tile-padding method, where for the specified pad length, the input image is tiled across as if repeating, as shown in Figure 4.9c and g. This method shows less blurring at the edges

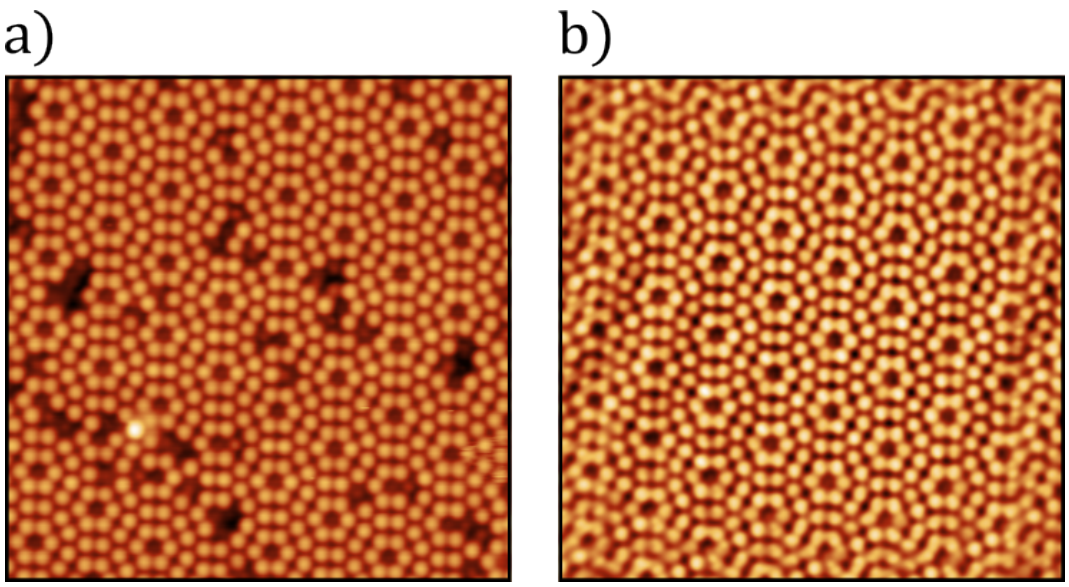


Figure 4.8: a) The original input image with no applied padding, which results in the filtered image shown in b).

of the scan frame, but with an overall worse estimation of the structure itself due to small misalignments of tiles at the boundaries, causing slightly different frequencies to be found in the FFT, and thus in the frequencies used to calculate the filtered image. Finally, the reflection-padding method, where the edges of the images are reflected to the pad length, is shown in Figure 4.9d and h. This method works very poorly overall, with the reflection causing a similar problem to that seen in the tile-padding, but with a much larger effect, meaning that frequencies which previously would have been found in the FFT are not as prevalent and so not found by the script. Overall, the mean-padding method was chosen as the optimal padding method and so was used throughout all results within this chapter, including the figures shown in the previous section.

In addition to padding, the effect of a Gaussian filter applied before calculation of the FFT was also trialed. This has the effect of softening the edges, therefore reducing the harshness of the discontinuities throughout the entire image, and as a result, reducing the presence of high frequency components in the FFT. A comparison the obtained filtered image with and without the

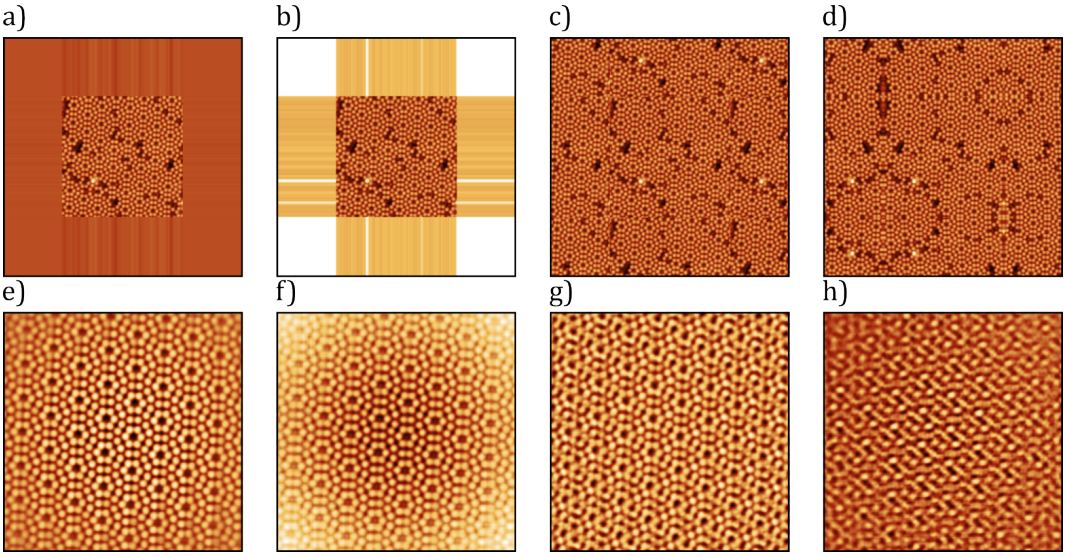


Figure 4.9: a)-d) show the input images with different methods of padding applied (mean, max, tiling and reflection), which result in the filtered images shown in e)-h) respectively.

Gaussian filter step is shown in Figure 4.10.

The effect of applying a Gaussian filter seems to add very little to the sharpness of the resultant filtered image, and appears only to produce an overall more blurry representation (Figure 4.10d) when compared to using no filter (Figure 4.10c). The magnitude of the Gaussian applied was varied, producing similar results. As a result of this, no pre-filter is applied in the final script.

The final addition which can improve the accuracy of locating the atomic positions found on the filtered image is to remove points found close to the edges of the image frame, as it is close to the edges that the reconstruction becomes less reliable due to the residual FFT artefacts. To this end, for the final script, positions found within the edge 10% of the image are removed, i.e., the script only finds the points in the central 80% of the image.

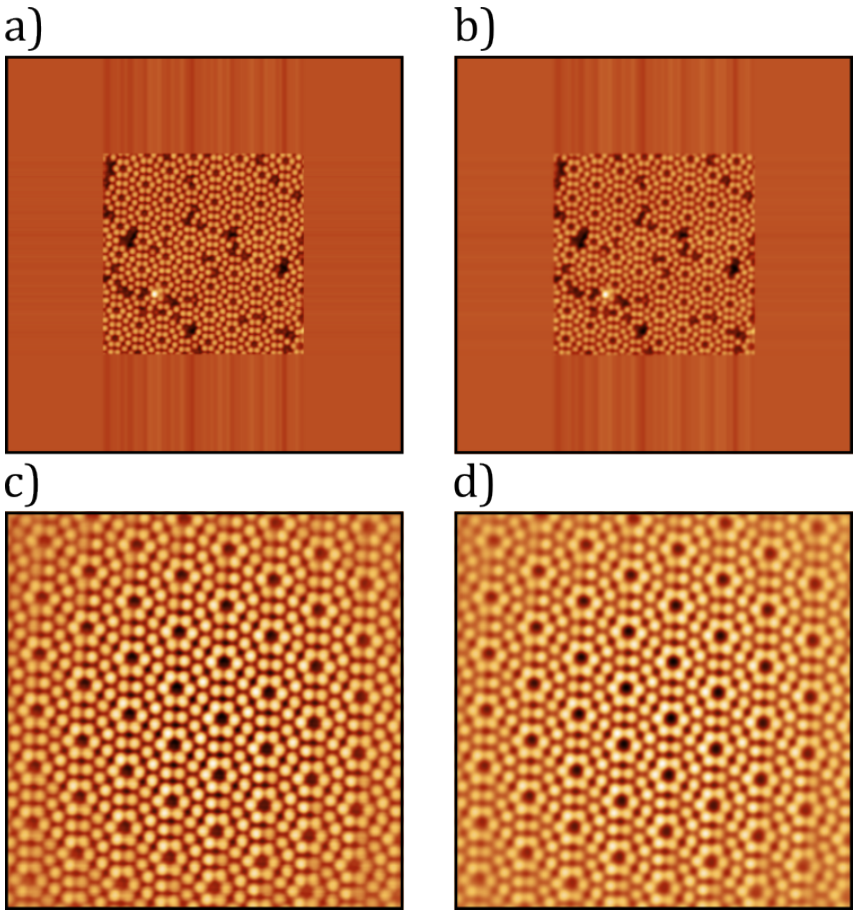


Figure 4.10: a) Input topograph with mean padding. b) shows the same image from a), with a Gaussian filter applied. c)-d) filtered surface estimates using the input images a)-b) respectively.

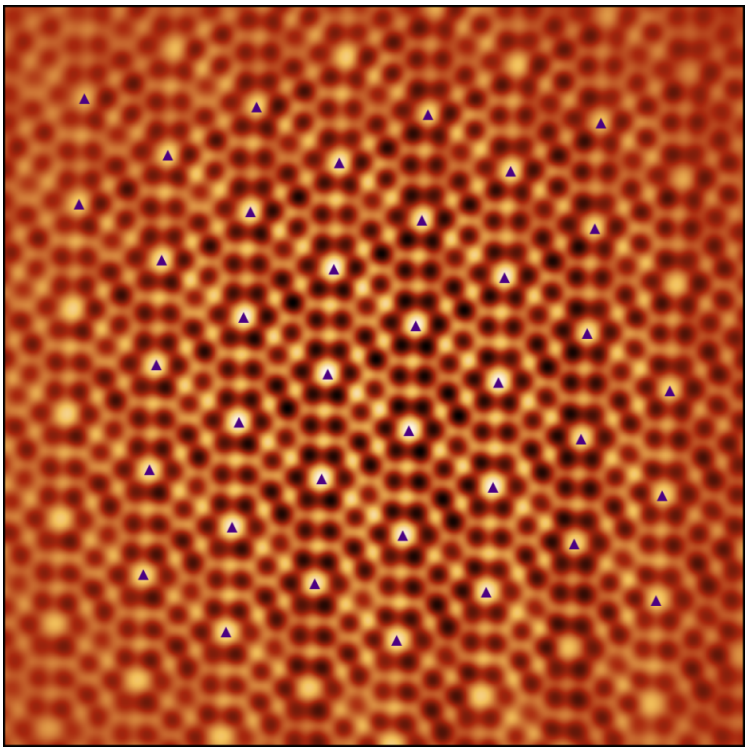


Figure 4.11: Filtered image obtained from an input Si(111) - 7×7 scan after inversion. Located corner holes features are plotted as triangles.

4.2.2 Corner Hole Identifier

Due to the unusual nature of the Si(111) - 7×7 surface reconstruction, an additional feature which is beneficial to locate automatically are the corner holes. The addition of this to the script is trivial, and involves only inverting the filtered image, and running it through the same function used throughout to locate peaks within the image. After the inversion, a slight Gaussian is also applied to the image to avoid spurious peaks in locations other than the corner hole sites. At this point, the corner holes appear as the only protrusions in the image, and so are easily located, as shown in Figure 4.11.

4.3 Results

4.3.1 Si(111)

The Si(111) - 7×7 substrate was chosen as our group has a large amount of historical data on this substrate, both in AFM and STM imaging. Atomic imaging on this surface is also fairly simple to achieve on the room temperature system, which was in use often throughout this project. In addition, due to the reactivity of the surface, it is very common for defects and adsorbates to be present, showing the applicability of this method to substrates containing larger amounts of contamination. Using the Fourier filtering method, the adatom locations on the surface, along with those obscured by defects and adsorbates were estimated as shown in Figure 4.12.

This figure highlights that the script is able to accurately identify the locations of the targeted features accurately, making very few errors in the classifications. When using a suitable input image, the script achieves an estimated accuracy of 99% in feature location and distinction, failing only by identifying clean adatoms as defects toward the edges of the scan.

4.3.2 B:Si

The B:Si substrate was also tested to demonstrate that this method can be applied to another surface with a completely different reciprocal structure. Using the Fourier filtering method, the feature locations are found accurately as shown in Figure 4.13, where an accuracy of 99% is achieved on suitable images, where misclassifications were only made in identifying clean adatoms as defects toward the edge of the scan.

4.3.3 Discussion and limitations

The two examples shown (Si(111) - 7×7 and B:Si) highlight the ability of this method to locate repeating features on a surface automatically. After locating the features, the positions can be easily input into an additional

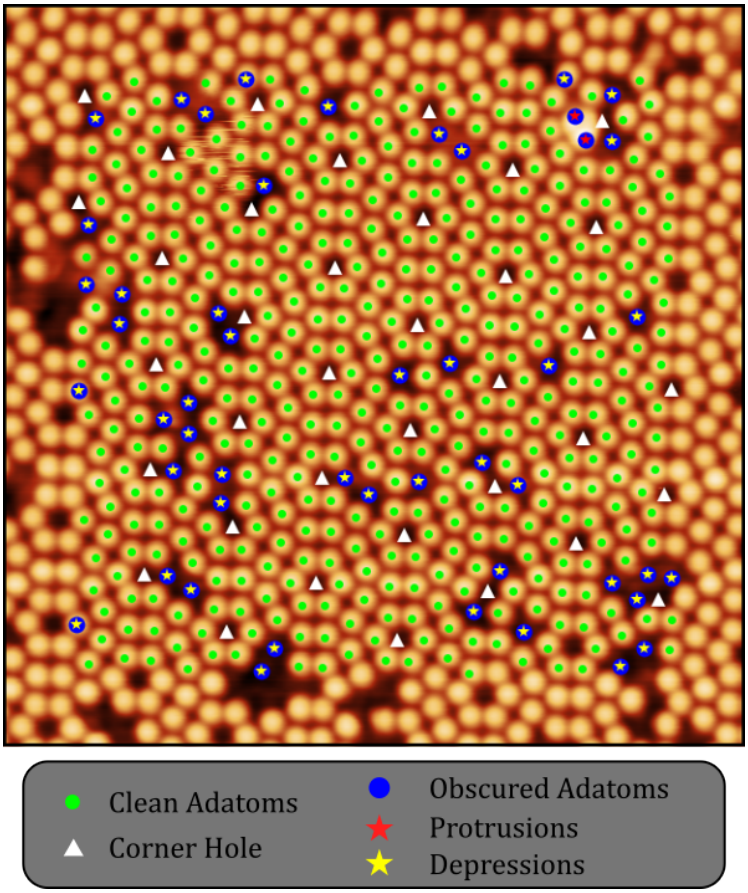


Figure 4.12: Si(111) - 7×7 topograph with automatically located surface features plotted. The features found are the clean surface atoms as green circles, corner holes as white triangles and obscured surface atoms as blue circles shown as either depressions with a yellow star or protrusions as a red star.

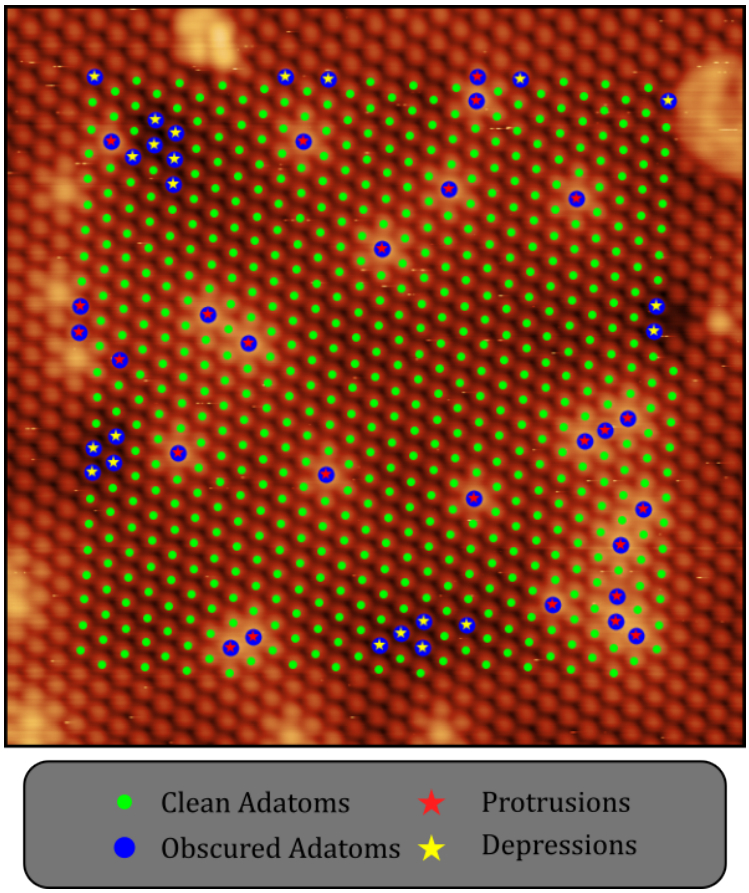


Figure 4.13: B:Si topograph with automatically located surface features plotted. The features found are the clean surface atoms as green circles and obscured surface atoms as blue circles shown as either depressions with a yellow star or protrusions as a red star.

automated script to carry out an automated experiment, such as taking spectroscopy measurements as will be discussed in Section 4.3.4.

Whilst the method works well in the examples shown here, and for any similar images, there are some important limitations to the technique. An essential component for this method to work is the presence of a repeating structure, which is clearly visible in the real space image. Without this, the peaks located in the FFT would not correspond to anything meaningful, and so the final output would not represent the initial image as intended. This need for a repeating structure, however, should not effect the use of this method in general as most atomic surfaces show a reciprocal structure which, in theory, should be found very easily.

There are some additional circumstances in which the input image may not be suitable to recreate the surface accurately, and so the positions of the features found could not be relied upon confidently; these situations are discussed in the following sections.

4.3.3.1 High Contamination

A high level of contamination on a substrate can obscure a large amount of the periodic structure of a surface. This can result in the specific frequencies, which correspond to the underlying periodic structure of the surface, also becoming obscured. In this situation, a much less accurate surface estimation is obtained. The input image shown in Figure 4.14a is a topograph of the Si(111) - 7×7 surface which is heavily obscured by adsorbates, resulting in the reconstruction shown in Figure 4.14b being much less refined.

The filtered image shown in Figure 4.14b clearly shows a structure close to that of the Si(111) - 7×7, however in the absence of some of the main frequencies needed, some atomic positions are missed or not found correctly as shown in Figure 4.14c. It should be noted however, that the majority of the positions are still found, even on a highly contaminated surface, and correctly labelled. However, due to some spurious features being included, an automated experiment on a surface with this much contamination would still not be ideal.

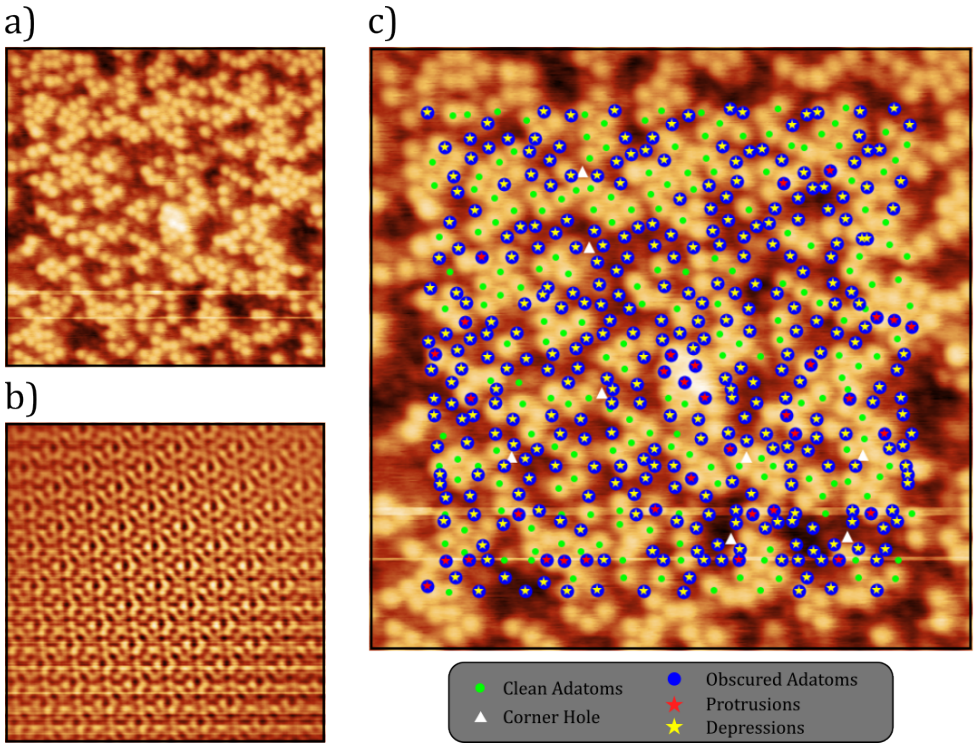


Figure 4.14: a) Si(111) - 7 × 7 topograph with a high defect density, obscuring the majority of the repeating structure of the surface. b) Automatically filtered image resulting from using a) as an input. c) The surface shown in a), with the automatically located features plotted.

Practically however, we note that it is unlikely that atomic resolution spectroscopy would be carried out on a surface as dirty as the one shown here and it is worth noting that if much of the surface structure is obscured, even a human operator would struggle to identify the obscured positions with high precision. Locating positions over the adsorbates themselves could be done automatically via a different method, such as cross-correlation (CC), as will be discussed later in Chapter 5.

4.3.3.2 Creep and Hysteresis

One common image "defect" in STM, as discussed previously in Section 2.1.2.3, is image distortion due to piezoelectric creep and hysteresis, where the scan appears to drift non-linearly over time. This causes the repeating structure of the surface to vary over the scan, and also therefore, in Fourier space. The peaks resulting from the repeating structure are consequently "smeared" in the y-direction, as shown in Figure 4.15. This effect is more likely to impact larger area scans, or those taken at slow scan speeds.

It is possible to account somewhat for this smearing of the peaks by including larger areas around the features in the FFT filtering, or including frequencies directly in a vertical line with the peaks, however this only results in better reconstruction of the image input into the script. This means that all features found from this filtered image will not be accurate to the true physical positions of the features. This is the same with any other imaging distortion, where the positions can only be estimated relative to their position in the image. If the image is, in itself, not accurate, then it is not possible subsequently, to locate the features accurately for spectroscopy or manipulation purposes.

4.3.4 Live Tool

The described automated method of locating atomic positions and features on a surface was additionally built into a tool to interface with the Nanonis scanning software. This tool was coded in LabVIEW, as with the other automation tools which will be discussed throughout this thesis. Before

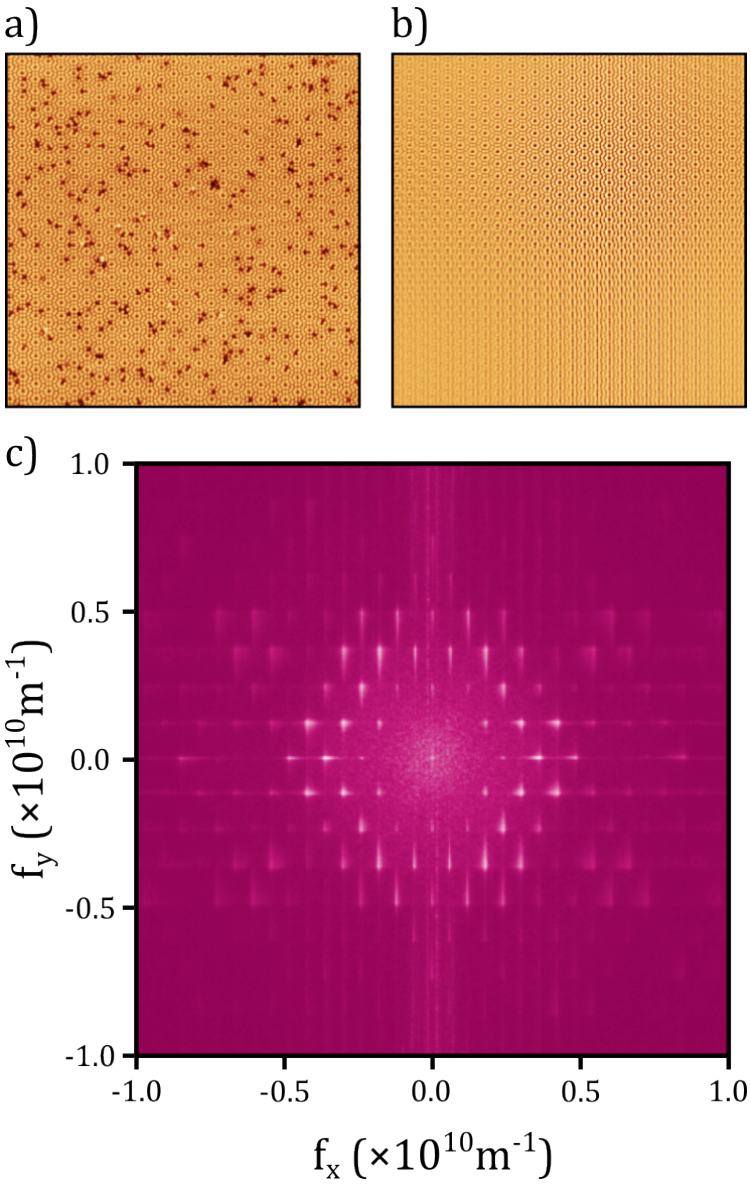


Figure 4.15: a) 50 × 50 nm STM topograph of Si(111) - 7 × 7 with slight creep in the long scan direction. b) Filtered image using the input image shown in a), where the method is not able to reconstruct differing spacing between atoms. c) The FFT resulting from a), which shows the smearing of the peaks due to the creep present in the scan.

starting, the script requires a well shaped tip, such that the atoms on the surface are clearly visible. From here, the script takes an image using the parameters set by the user, and then passes the scan to Python for analysis. The script then locates the desired features within the scan. Specifically for the Si(111) - 7×7 surface, which was the only substrate the tool was tested on, the tool locates the corner holes, clean adatoms, and obscured atoms which appear as protrusions and depressions.

Before starting the LabVIEW script, the user is able to choose which type of spectroscopy measurement is desired (e.g., I(z) or I(V)), and which features the script should carry out the measurements over. After locating all the features in the scan, the script will carry out spectroscopy measurements over the desired locations. In trial runs, the script has worked well, however, the testing was limited as work shifted to the CC method for tip analysis, discussed in Chapters 5 and 6.

4.4 Conclusion and Outlook

The presented Fourier filtering method effectively identifies the positions of surface atoms in multiple periodic structures, demonstrated on Si(111) - 7×7 and B:Si. This method is able to map the positions of clean and obscured/missing surface atoms from a given input image, accurately distinguishing between the two, with minimal knowledge of the surface. The method works by calculating the FFT of the input image, and isolating the high intensity peaks, corresponding to the repeating structure within the image. Inverting back into real space results in a representation of a perfect atomic surface, taken over the same imaging areas as in the input. By comparing this reconstruction to the input image, the positions of clean and obscured surface atoms are determined.

This method was also utilised in an automated tool, which is able to scan an area, and take spectroscopy measurements over the atomic positions found on the surface. The mapping produced by this method could further assist in automated manipulation scripts, where an initial surface overlay is required before the experiment can begin. This method however, is limited by the image input into the script. Common imaging artefacts in scanning

probe microscopy (SPM) will be propagated into the FFT, leading to errors in the filtered image and, consequently, less accurate determination of atomic positions.

Additionally, this method relies heavily on the quality of the surface in the input image. If the surface is heavily contaminated, such that much of its periodic structure is not visible, the accuracy is greatly diminished. In later chapters, a method called CC for finding positions within an input image that most closely correlate with a reference image is discussed in detail (see chapters 5 and 6.) This method excels at locating specific features within an input image, such as adsorbates or specific features of a substrate. Applying the CC method to a heavily contaminated surface, such as the one shown in Figure 4.14, with a reference image containing a section of the unit cell, could potentially locate the position of a unit cell within the image. From this, the reference could be tiled across the entire image, facilitating surface mapping even with a highly contaminated surface.

Further development of this method could include the development of a fully automated manipulation experiment using the STM at low temperatures. The Fourier filtering method would be used to map the underlying surface on a substrate such as Cu(111), creating an overlay to plan the positioning of individual atoms/molecules. From here, CC could be used to locate adsorbates present, followed by path planning and automated manipulation routines.

References

(1) Avouris, P. *Accounts of Chemical Research* **1995**, *28*, 95–102.

(2) Crommie, M. F.; Lutz, C. P.; Eigler, D. M. *Science* **1993**, *262*, 218–220.

(3) Kalff, F. E.; Rebergen, M. P.; Fahrenfort, E.; Girovsky, J.; Toskovic, R.; Lado, J. L.; Fernández-Rossier, J.; Otte, A. F. *Nature Nanotechnology* **2016**, *11*, 926–929.

(4) Jonathon Mamin, H.; Ried, R. P.; Terris, B. D.; Rugar, D. *Proceedings of the IEEE* **1999**, *87*, 1014–1027.

(5) Achal, R.; Rashidi, M.; Croshaw, J.; Churchill, D.; Taucer, M.; Huff, T.; Cloutier, M.; Pitters, J.; Wolkow, R. A. *Nature Communications 2018 9:1* **2018**, *9*, 1–8.

(6) Fuechsle, M.; Miwa, J. A.; Mahapatra, S.; Ryu, H.; Lee, S.; Warschkow, O.; Hollenberg, L. C.; Klimeck, G.; Simmons, M. Y. *Nature Nanotechnology* **2012**, *7*, 242–246.

(7) Weber, B.; Mahapatra, S.; Ryu, H.; Lee, S.; Fuhrer, A.; Reusch, T. C.; Thompson, D. L.; Lee, W. C.; Klimeck, G.; Hollenberg, L. C.; Simmons, M. Y. *Science* **2012**, *335*, 64–67.

(8) Kolmer, M.; Godlewski, S.; Lis, J.; Such, B.; Kantorovich, L.; Szymonski, M. *Microelectronic Engineering* **2013**, *109*, 262–265.

(9) IBM Research, A Boy and His Atom https://www.youtube.com/watch?v=oSCX78-8-q0.

(10) Celotta, R. J.; Balakirsky, S. B.; Fein, A. P.; Hess, F. M.; Rutter, G. M.; Stroscio, J. A. *Review of Scientific Instruments* **2014**, *85*.

(11) Leinen, P.; Esders, M.; Schütt, K. T.; Wagner, C.; Müller, K. R.; Stefan Tautz, F. *Science Advances* **2020**, *6*.

(12) Chen, I. J.; Aapro, M.; Kipnis, A.; Ilin, A.; Liljeroth, P.; Foster, A. S. *Nature Communications 2022 13:1* **2022**, *13*, 1–8.

(13) Skeini, T.; Steiner, J. F.; Hla, S. W. In *2006 6th IEEE Conference on Nanotechnology, IEEE-NANO 2006*, Institute of Electrical and Electronics Engineers Inc.: 2006; Vol. 2, pp 610–612.

(14) Møller, M.; Jarvis, S. P.; Guérinet, L.; Sharp, P.; Woolley, R.; Rahe, P.; Moriarty, P. *Nanotechnology* **2017**, *28*, 075302.

(15) Morgenstern, K.; Lorente, N.; Rieder, K.-H. *Phys. Status Solidi B* **2013**, *250*, 1671–1751.

(16) Heinz, K.; Hammer, L. *Journal of Physical Chemistry B* **2004**, *108*, 14579–14584.

(17) Aversa, L.; Taioli, S.; Nardi, M. V.; Tatti, R.; Verucchi, R.; Iannotta, S. *Frontiers in Materials* **2015**, *2*, 46.

5 Automated Image Based Tip State Classification

This chapter is adapted from a recent paper published in ACS Nano: Barker 2024 [1].

Scanning probe microscopy (SPM) has allowed for great advancements in the investigation of nanoscale phenomena. Central to the success of SPM techniques is the quality and sharpness of the probe tip, which directly influences the resolution, sensitivity and reliability of measurements. In addition to this, tips can be functionalised, where an additional defined molecule or atom is picked up by the tip in order to further increase the resolution. One, well known example of which is the work by Gross *et al.*, where the functionalisation of a probe by picking up a CO molecule was used to drastically increase the resolution of a pentacene molecule in NC-AFM [2].

Despite this being such an important aspect of imaging in SPM, the preparation of usable tips has not been fully perfected when fine tuning the shape *in situ*. Before being used in scanning, tips can be prepared to be as sharp as possible using methods such as electrochemical etching and e-beam heating (as discussed in Section 2.1.2.5), however following these treatments, the atomic scale structure of the tip apex is still unknown, leading to uncertainty in the interpretation of the sample structure. Maintaining a probe tip via *in situ* tip preparation is a very time consuming process, where an operator will attempt to coax the tip into a more suitable shape, through tip crashing and/or voltage pulses. Although time-consuming, this method works within a fixed parameter space, resulting in an ideal candidate for automation.

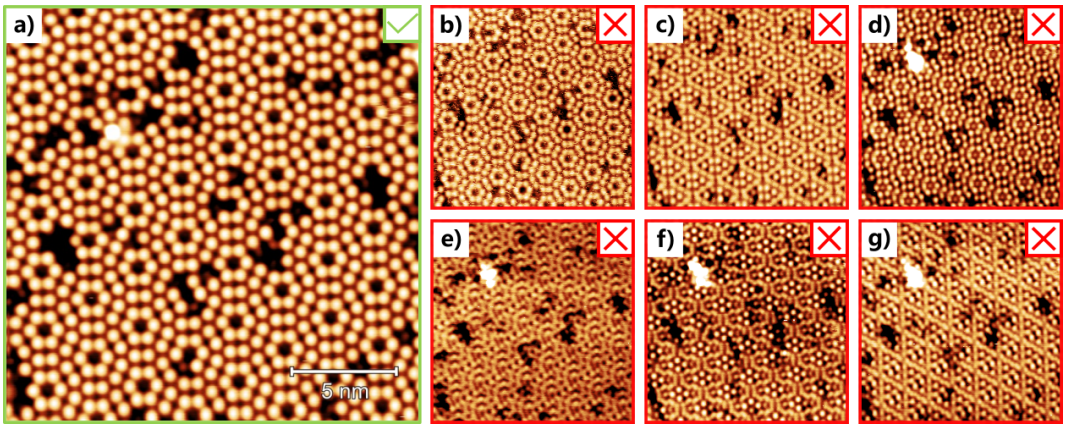


Figure 5.1: Constant current scanning tunnelling microscopy images of the same region of a clean Si(111) - 7 × 7 surface taken at 2 V, 200 pA. Image a) shows a topograph taken using a "good" tip, images b)-g) show topographs obtained using various "bad" tips.

Unfortunately, there is currently no commercially available tool to prepare a probe tip. This is primarily due to the fact that this would first require a method of classifying the state of the tip in the first place. Unfortunately, this task is not a trivial one. One reason for this difficulty in classification is the variety of surfaces which can be imaged, with vastly different apparent structures; a general protocol for classification of a sharp probe tip is therefore difficult to define, even for an experienced operator. In addition to this is the problem of ambiguity in classification. For example, we can ask the question: which image in Figure 5.1 is the correct image? It has been well documented that different probe terminations result in probes which are more suited for different tasks; e.g. varying the tip state may give different desirable surface resolutions as seen on H:Si(100) [3, 4], vary the probability of successful hydrogen extraction from the same surface [5], or manipulate complex molecules on a surface, or even alter the quality of STS data [6]. It is therefore a requirement that before we prepare our tip, we know exactly its intended purpose, and define metrics for its classification.

The idea of automating the process of tip optimisation is not new, with discussions of easing tip preparation dating back to the 1990s [7]. In the intervening time, there have been a number of attempts to automate this process [4, 8, 9], the majority of which use machine learning (ML) based

methods, these attempts were discussed in detail in Section 2.3.1. Unfortunately the process has not yet been fully realised. The idea of a general tip state classifier, able to work on any surface is sadly not likely to be viable; the amount of variation in the appearance of different surfaces would mean that the dataset needed to train a ML model would be enormous, which is a particularly difficult problem to overcome in nanoscience [10].

Currently, considering the amount of time and data required to make accurate classifications on a new surface, an alternative approach is needed. This chapter will discuss our efforts to make these classifications without the use of ML, using deterministic image analysis methods. We find that final classifications can be made with these methods, using only a single image of a surface taken with a probe in the desired state, in combination with some prior knowledge of the structure of the surface itself. This deterministic method is compared to commonly used ML methods, and tested on the same isolated sample set. Once this classification is made, it is possible to move on to acting on the classification, towards reshaping the probe itself.

5.1 Datasets

In looking to develop a classifier, be it ML based or otherwise, it is important to consider the data being used carefully. Specifically in the case of ML, learning will only be effective if the dataset is representative of the entire feature space. In our case, if we want our model to learn what a "bad" tip looks like, we will need to show it as many examples of a these tips as possible, capturing the majority of possible appearances. Similarly, when evaluating classifications made using non-ML based methods, it is ideal to have a large dataset which adequately shows the variety in the dataset, as this will allow for the most accurate final analysis.

In addition to this, the labelling of the dataset is extremely important; ambiguity in labelling may be propagated into the model itself. For this reason, it is ideal to plan labelling strategies carefully, as will be discussed later.

For this work, three different surfaces were considered: Si(111) - 7×7, B:Si, and Cu(111). The methods described in this chapter were applied to the latter surface in two different deposition configurations, the first with a low coverage of both Cu adatoms and CO molecules, and the second with the addition of C_{60} molecules. All systems considered are described in more detail in Sections 3.1.1 and 3.1.2.

5.1.1 Automated Dataset Generation

Ideally, when beginning to build a classification model, the dataset used will be comprised of historical data from the group which needs only to be labelled and tidied up.

However, in normal operation of an scanning tunnelling microscope (STM), a lot of time will be spent on the initial *in situ* preparation of the probe, where it will be clear to an operator within the first few lines of a scan whether the tip is in an ideal state for imaging, and if not, the scan will be stopped for more preparation. The result of this is that it is unlikely that historical datasets will contain a large amount of full scans obtained using a probe in a non-ideal state. As is often the case for many other SPM groups, the data present in our datasets contained either incomplete scans using a non-ideal tip or full scans with an ideal tip. If used for training, this would mean that our training dataset would be hugely biased toward the "good" tip state. In addition to this, historical datasets generally contain varied scan sizes, pixel densities and scan speeds, which whilst not a problem for usual operation, would add some complexity to classification models. We therefore needed to find an efficient way of producing a large number of scans with constant imaging parameters and different tip states

As a solution to this problem, an automated dataset generation script was written in LabVIEW, which is able to scan the same imaging area repeatedly, whilst performing tip preparation between scans. The tip preparation methods and parameters used for each surface varied depending on what was found to reliably produce a significant change in the tip state, without making such a drastic change that it needed manual attention. A flow chart describing the general order of events in a dataset generation run is shown

in Figure 5.2a-d. Before starting the script, the imaging parameters (bias, tunnel current setpoint, scan speed, scan size and pixel density) are chosen such that imaging is in the desired state. The script then begins by taking an initial image, before moving away to perform a tip preparation routine. The tip moves around a larger preparation frame (50 nm away from the imaging area) to ensure the preparation steps are carried out over different areas of the surface, far enough away from the imaging area so as to reduce the effect of the preparation on imaging. A large overview scan showing this preparation frame is shown in Figure 5.2f, where indentations around the frame have affected the surface in these areas. After the tip has been changed, the tip is moved back to the imaging area and another scan is taken. These steps are repeated for a set number of scans. Between two successive scans, the drift between the two is calculated as described in Section 5.1.1.1.

In this way, large, varied datasets could be produced in a few days, without the need for constant operator monitoring. Creating a dataset in this way also has the advantage that we are able to constrain the usual variables which would arise from using historical data, such as varied scan sizes, pixel densities and imaging parameters. Producing a dataset from scratch allowed us to fix all of these factors as well as reduce the bias in the dataset itself. Fixing the imaging parameters does have the effect of producing a less generalisable model, but ideally, should improve the overall performance as the ML model is not required to learn these extra variations.

Due to this, for the purposes of dataset generation and evaluation of the different methods, we selected a standardised size, and parameter set for each surface used. All topographical images used for training were acquired in constant current mode with a scan frame of size 20×20 nm^2 and a resolution of 720×720 pixels. Si(111) - 7×7 scans were obtained using a tunnel current setpoint of 200 pA and a bias voltage of 2 V, whist scanning at a speed of 277.8 nm/s, with proportional and integral gain values of 10 pm and 120 nm/s respectively. B:Si(111) scans were obtained using a tunnel current setpoint of 250 pA and a bias voltage of 2 V, scanning at a speed of 277.8 nm/s, with proportional gain of 20 pm and an integral gain of 90 nm. Cu(111) scans were obtained using a tunnel current setpoint of 100 pA and a bias voltage of 100 mV, scanning at a speed of 185.2 nm/s, with a proportional gain of 1 pm and an integral gain of 100 nm/s.

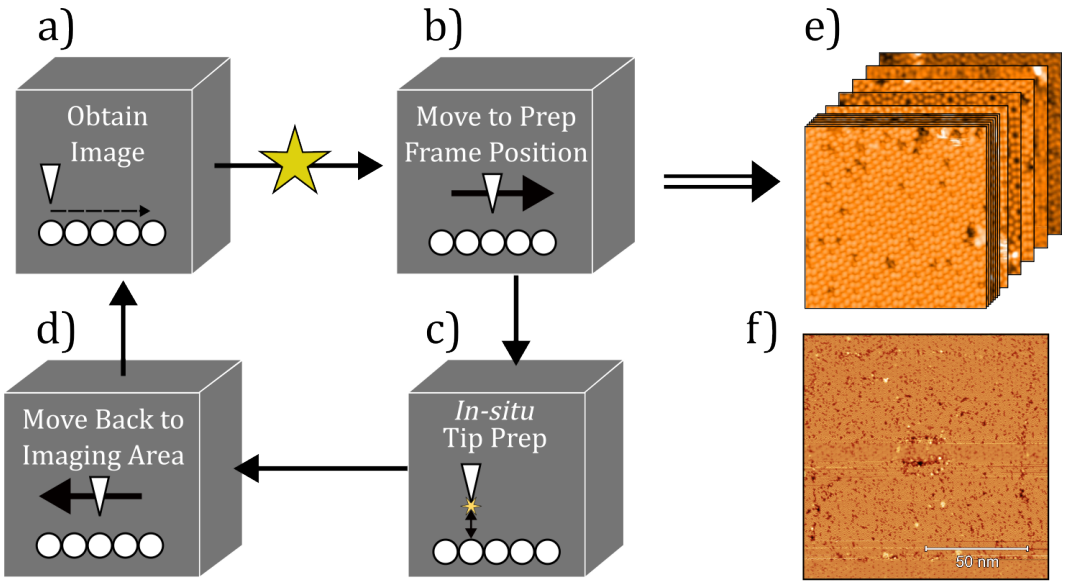


Figure 5.2: a)-d) Flow diagram of the data gathering procedure. a) the script starts by taking a scan of the imaging area, after which the tip is moved 50 nm away to the next position on the preparation frame, b), where a tip preparation routine changes the shape of the tip, c). Following this, the tip is moved back to the imaging area, d), and another scan is taken. This is repeated until a set number of scans are obtained. The star between a) and b) indicates drift compensation, which is applied between two successive scans. e) Final large dataset of images of the same area is produced. f) 120 × 120 nm STM (2 V, 200 pA) image after a data gathering run, showing damage to the surface in the central imaging area and in the shaping frame around it. The damage to the central region is due to tip dropping over long generation runs.

5.1.1.1 Cross-correlation Based Drift Compensation

When acquiring data for surfaces at room temperature, it is important to address the issue of thermal drift, which can result in significant distortions to the apparent surface structure [11, 12]. One method to compensate this drift is by using atom tracking [13], which works by locking the tip onto a surface feature and tracking its displacement over time and hence apply a feed-forward correction vector to the piezo scan tube to compensate. This compensation needs to be regularly updated as the drift between the tip and sample varies over time. However, this method is less suitable for an automated script which deliberately induces large tip changes, as the initial tracking position needs to be manually chosen for the displacement to be measured, and any significant tip change will alter the position of the tunnelling apex, changing the relative position of features on the surface.

To overcome this issue, the image acquisition script also included a cross-correlation (CC) based drift detector, which identifies the physical drift between two consecutive images and updates the drift vector in the control software. This drift detector estimates the physical drift of the scan frame by first calculating the CC of two consecutive images, and assuming the highest correlated position corresponds to the position which the centre of the first image has drifted to after the second scan is taken. Figure 5.3 shows the measured drift between an input image, Figure 5.3a, and a series of other images. From this highest correlated position, it is possible to determine the physical drift which has occurred between the two frames, and using the time taken between frames the drift vector (in m/s) can be calculated and the compensation updated in the scanning software. This update to the feed-forward vector compensates the residual thermal drift.

We found the CC based drift detector is able to detect the residual drift accurately even after a tip change has occurred, this is possible as the CC algorithm will find the drift based on features which are still present in both images such as defects and adsorbates, even if the exact contrast on the atoms has changed. To avoid applying erroneous drift compensation, the drift detector outputs the CC metric of its highest correlated position. It was found that by disregarding drift vectors with an CC value of less than 0.5, erroneous updates could be avoided.

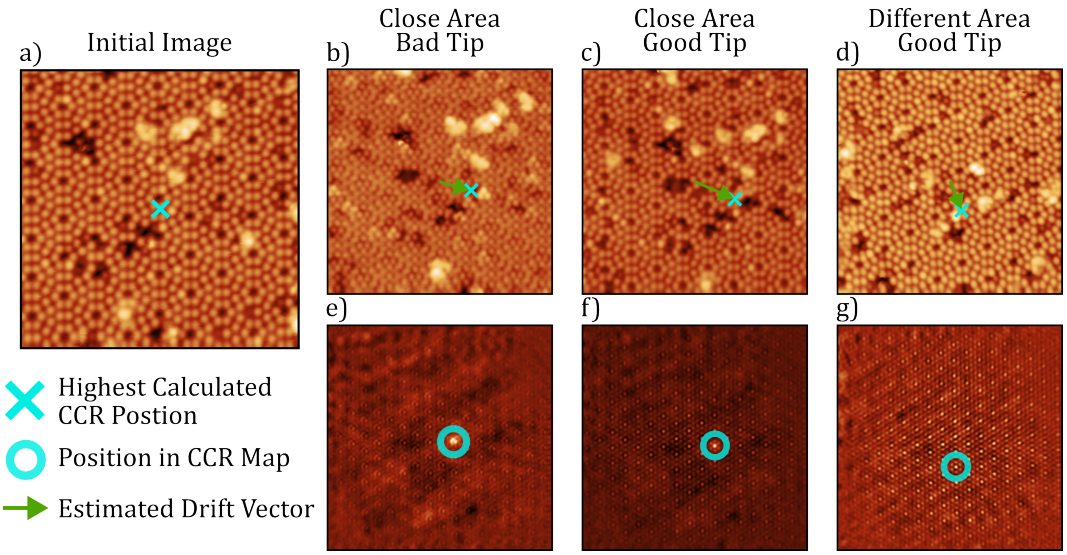


Figure 5.3: a) Initial scan of the Si(111) - 7 × 7 surface, the centre of the image is marked with a blue cross. b) Scan taken close to the a) with a "bad" tip. Due to other features within the scan (adsorbates/defects) the script is able to find the drift well even when scanning with a bad tip. c) Scan taken close to a) with a "good" tip, again finding the drift well. d) Scan taken in a completely different area to a), with a "good" tip. Here, the script finds a drift which is not representative of a real drift, but is the highest CCR found between images a) and d). e)-g) show the CCR maps calculated between a) and b)-d) respectively. The peaks found in e) and f), where the imaging area is the same, are clearly highest in magnitude compared to the other peaks found, whereas the highest peak in g) is of similar magnitude to the surrounding peaks. Calculated CCR values for each image compared to a) are b): 0.48, c): 0.71 and d): 0.34.

In this way, the drift compensation can be updated between successive images. Maintaining the same scan region also reduces the chance of the tip drifting over an area which will produce less suitable images, such as a step edge or disordered area of the surface. We therefore note that all data considered in this paper were acquired in the absence of significant distortion due to thermal drift.

5.1.1.2 Changing the Probe Tip

As mentioned previously, when aiming to generate a large, representative dataset, changing the tip state automatically between successive images is essential. Before starting the automation script, various bias pulse and tip indent parameters were tested during imaging to identify the optimal values for causing a significant change to the resolution of the image, without damaging either surface or tip too drastically. These parameters would vary depending on the surface being used, as well as the initial state of the tip and so this testing step is needed before each data capture run.

Once the specific tip shaping parameters were chosen, these are entered into dataset generation script, and it will use these throughout. Between successive scans, the tip is pulled back on the piezo (≈ 1 µm), feedback controller is turned off and the tip is moved away from the imaging area for tip preparation. To do this, the built in Nanonis tip shaping tool is used, pictured in Figure 5.4, which applies a set bias whilst indenting the tip into the surface. Once complete, the tip is moved back to the imaging site and re-approached to take the next scan.

Generally it was found that applying a small bias ($0.5 - 1$ V) whilst indenting the tip into the surface a small amount ($600 - 1000$ pm below the setpoint height) was sufficient on the silicon surfaces, with slightly larger indents needed on the copper sample.

5.1.2 Data Labelling

Prior to use in labelling, training, and evaluating, the obtained images underwent minor augmentations as would be done in normal processing by

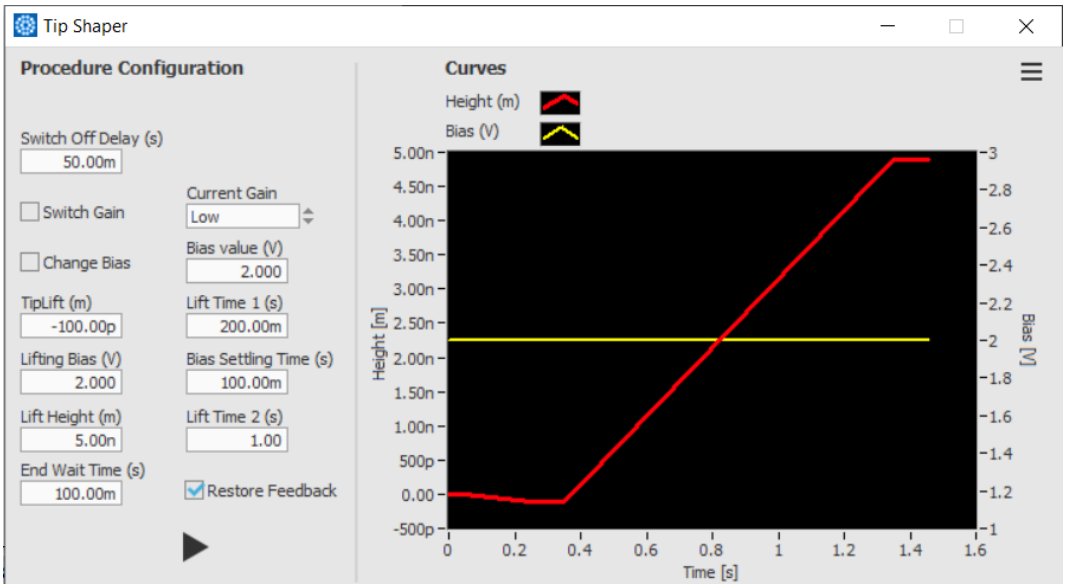


Figure 5.4: Built in tip-shaper tool from the Nanonis software used for tip preparation. The main parameters varied here are the distance the tip is indented into the surface (tip lift) and the bias applied to the tip during the indentation (lifting bias).

a human operator. These augmentations include plane flattening the images (to reduce the effect of a sample misalignment on the appearance of the topograph), median subtraction by row, thresholding, and removing the first 2.7% of the scan lines, in an attempt to remove the majority of visible creep artifacts caused by the prolonged relaxation of the piezoelectric scanner after moving large distances for tip treatment. After these augmentations, the final images were 700×700 pixels (19.4×19.4 nm^2).

"Good" and "bad" tips were defined on a case-by-case basis, depending on the system being studied and the expected appearance of the surface when scanning using a "good" tip. For the Si(111) - 7×7 surface, the main qualifiers of a "good" tip were; the surface adatoms appearing as well defined, with a good contrast between the adatoms and the corner-holes, and the appearance of the overall surface being that of the 7×7 structure with a diamond shaped unit cell containing 12 atoms and corner-holes at the corners, as shown in Figure 5.5a. In contrast to this, the B:Si(111) and the Cu(111) surfaces were categorised primarily by specific protrusions in the

topograph (dangling bond defects (DBs) in the case of the B:Si(111) surface and adsorbed Cu adatoms on Cu(111)) as these features demonstrated a higher selectivity to small changes in the tip apex. These features appear misshapen when imaged with a "bad" tip as opposed to their usual round appearance when imaged with a "good" tip and are hence highly sensitive tip classification points which allow for clear "inverse imaging" of the probe tip. Finally, on the Cu(111) surface with a low coverage of C_{60} molecules, the molecules themselves are used to classify the state of the probe tip, with a "good" tip showing the molecules with three lobes (as shown in 5.5e). These molecules protrude relatively far from the surface (an apparent height of $\sim$ 600 pm in STM at 0.1 V, 100 pA, and an actual height of $\sim$ 1 nm), therefore the imaging is much more sensitive to the "nanoscale" shape of the tip apex (see Figure 2.10, as tip defects further up the shaft are more evident during a scan, which can result in the appearance of double/multi tip features. Because of this, we only classify the primary apex of the tip rather than the tip as a whole.

When labelling a dataset for a classification scheme, it is key to consider the amount of ambiguity which could be introduced into the training set. In general, when labelling a dataset with binary labels, such as labelling a set of images of animals "dog" or "not a dog", it is assumed that the labels being used are completely accurate and obvious (i.e., the distinction between the two should be easy to make). However, when labelling a set of images based on an individual's opinion of the features present, as is the case here, it is inevitable that some ambiguous images remain due to lack of agreement between labellers. The level of ambiguity in labelling itself can be reduced somewhat by setting out a clear classification scheme, which each labeller is to follow, including examples of each category and descriptions of specific features which should be present in each, this is shown in Figure 5.6.

For the Si(111) - 7 $\times$ 7 dataset, labelling was carried out by four members of our group who are all familiar with the surface itself, as well as general atomic resolution imaging at ultra-high vacuum (UHV). Each image was to be classified into one of four categories: "Excellent", "good", "bad" and "tip-change". Examples of each of these four categories were given in the labelling scheme provided to the labellers beforehand. Initially, the total number of images obtained was 1308, with 873 remaining after the removal

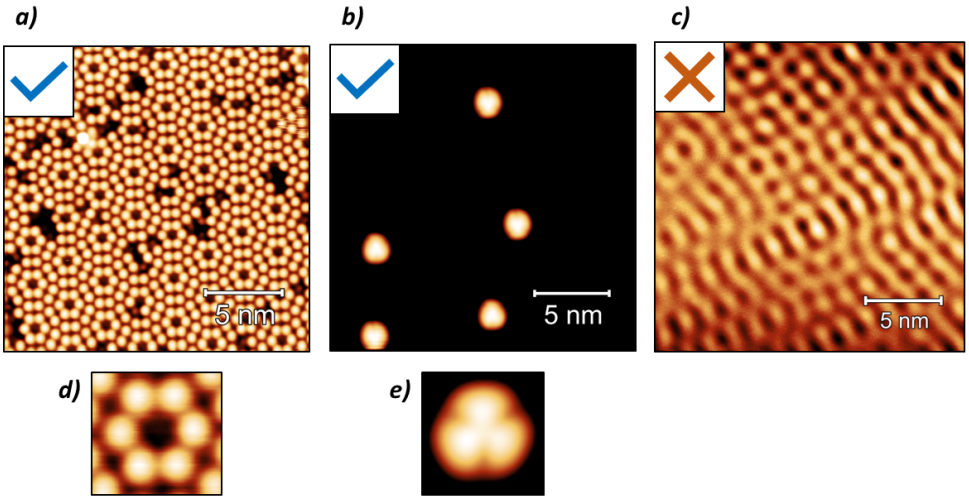


Figure 5.5: Examples STM images of systems which are suitable, a-b), and not suitable (discussed later in Section 5.5), c), for classification via cross-correlation. d)-e) show example reference images to be used in the CC classification on the surfaces shown in a)-b) respectively. a) Si(111) - 7×7 surface imaged at 2 V, 200 pA. b) Cu(111) with a low coverage of C_{60} molecules imaged at 5 K, 100 mV, 100 pA. c) Bare Cu(111) surface imaged at 5 K at 1 mV, 1 nA. This surface is not be suitable for CC based classification, as no common repeating features are visible on the surface, only the standing wave pattern of free electrons. d) Corner-hole feature with six surrounding silicon surface atoms on Si(111) - 7×7. f) Single C_{60} molecule on Cu(111).

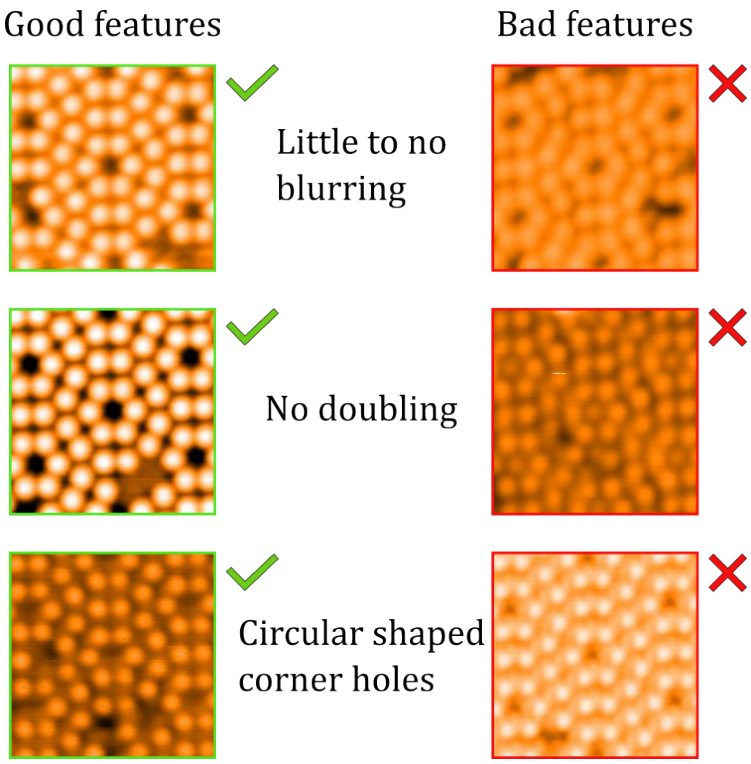


Figure 5.6: Example of the main classification metrics used within our labelling scheme for the Si(111) - 7 × 7 substrate.

of ambiguously labelled images (discussed later). Of these, 265 were labelled as "good" and 608 as "bad". An analysis carried out on this group labelled dataset (discussed in more detail in Subsection 5.1.2.2) found that the overall accuracy of the labels was not significantly reduced when labelled by only a single operator (myself), and so the remaining datasets were labelled in this way, with the same four categories as before. The B:Si(111) dataset initially contained 1701 images, with 1296 remaining after the removal of ambiguous images. After labelling, this dataset contained 408 "good" images and 888 "bad". Finally for the Cu(111) dataset with a low coverage of CO and Cu adatoms, the initial dataset contained a total of 2036 images, with 40 classified as "good", and 1996 as "bad" with no images needing to be removed due to ambiguity. For this last dataset, rather than the four categories used previously, three were decided on; combining the "excellent" and "good" categories into one "good" category, and keeping the "bad" and "tip-change" categories as previously. This dataset resulted in a very imbalanced split between classes, which will be discussed in more detail later.

Post-labelling, a script was created to detect changes in the probe from an input topograph, meaning images which were initially classified as tip-change could simply be removed, avoiding adding this complexity to the automated classification models. This script is able to detect tip changes during a scan by taking a line scan from the slow scan direction, and looking for drastic changes in the height from line-to-line which would not be present in a normal scan. It was also decided that for ease of model training, the "excellent" and "good" categories should be combined into a single "good" class (as was done before labelling for the Cu(111) dataset). As a whole, this reduces the complexity needed in the automated classification scheme.

Any labelling script produced will only be able to label a set of images as accurately as the manual labels given to it. Therefore, if a labeller in unsure on any specific label, use of such an ambiguous label would propagate this uncertainty into the model itself. Therefore an additional step to reduce the effect of this in the classifications was to remove the ambiguous images from the dataset. These images were defined differently depending on the number of labellers: for the Si(111) - 7×7 dataset, ambiguous images were defined as those which were not agreed upon by the majority of labellers.

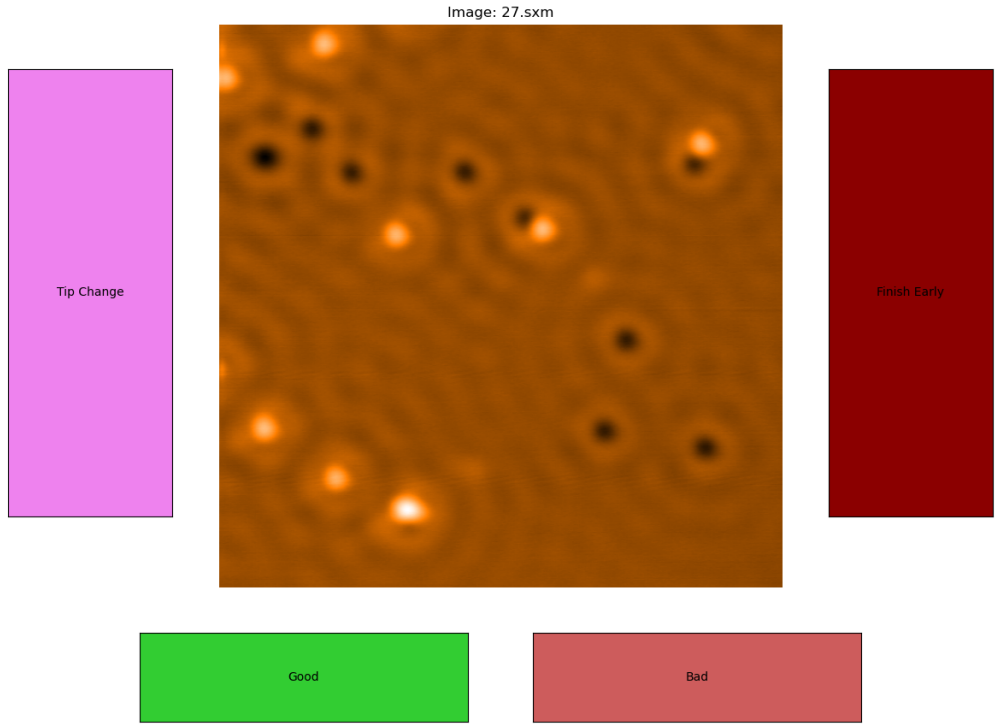


Figure 5.7: User interface for the script used to label each dataset, here showing an image of the Cu(111) surface with Cu adatoms (protrusions) and CO molecules (depressions).

For the datasets labelled by a single person, an additional class was included in the labelling script for scans which did not clearly fall into any single label.

To ease the labelling process, a simple python program was written with a graphical interface which showed the labeller a single image at a time, with a button for each possible label, the interface is shown in Figure 5.7.

5.1.2.1 Final Datasets

At the end of the labelling process, four total datasets were available to be used, three of which were fully labelled (Si(111) - 7×7, B:Si, and Cu(111)

with a low coverage of CO and Cu adatoms), and one final dataset which was unlabelled, containing scans of the Cu(111) surface with a low coverage of C_{60} molecules.

Of the labelled datasets, the two silicon based sets resulted in well balanced, binary categories of adequate size to be used in a ML based classification model, with roughly 30% "good" images, and 70% "bad" images. The labelled copper dataset, on the other hand, resulted in an extremely biased "bad" tip category, which contained 98% of the total images within the set.

From an initial rough analysis of the final Cu(111):C_{60} dataset it was evident that a similar, if not worse, imbalance in this dataset would be present. For this reason, the final dataset remained unlabelled and so no ML training was attempted using these images.

5.1.2.2 Labelling Analysis

For the Si(111) - 7×7 dataset, an additional 10% of repeated images were included, allowing for each labellers self-agreement to be calculated. From this it was found that the labellers agreed with themselves on average 92% of the time. Additionally, by comparing each operators labels to the majority voted labels, assumed to be the ground truth, their accuracies and precisions could be calculated; on average the accuracy of the whole group was 92% with a precision of 84%. In the work of Gordon *et al.* [4], when labelling a small subset of H:Si(100) data into 6 categories, between 9 microscopists the total agreement was 73%.

As mentioned previously, after labelling as a group, it was determined that the difference between labelling in this way and taking the majority voted label, compared with a single labeller (myself) was negligible. This was found in part as the solo labeller achieved an individual accuracy (again, based on the majority voted labels) of 94%, showing that they almost always completely agreed with the majority. In addition to this, a deterministic classifier (discussed in detail in Subsection 5.2) was evaluated both on the majority voted labels and the individual labels, results from this evaluation are shown in Table 5.1, where comparable results were observed.

	Majority Voting	Solo Labelling
Accuracy	90.2%	90.8%
Precision	94.6%	97.3%

Table 5.1: Table comparing the results of two deterministic classifiers trained on labels obtained from the majority vote of four human operators and labelling carried out by a single individual.

5.2 Deterministic Based Classifier

The deterministic classifier was developed in an attempt to overcome the need for a large labelled dataset, a key problem with ML methods. The aim of the deterministic classifier is to have a model which is able to make independent classifications of the state of a scanning probe tip, using set algorithms which can be applied to images without the need for any training.

5.2.1 Analysis Methods

The deterministic classifier developed here uses standard image analysis techniques; specifically cross-correlation and a measure of circularity, both of which will be discussed in depth. We found that these methods are sufficient to classify the state of a probe tip using only a single image when applied to the systems being shown here. An additional image analysis method, Fourier ring correlation, was also trialled, however, it was found not to increase the classifier's accuracy or precision and so is not used in the final classifier.

5.2.1.1 Cross-Correlation

Cross-correlation is a fundamental technique used in image processing and computer vision to analyse the similarity between different parts of an image. It plays a crucial role in tasks such as object recognition, image registration, and feature extraction. By measuring the similarity between two images or comparing a template with an image, cross-correlation enables

us to identify patterns, locate objects, and align images. At its core, cross-correlation involves scanning a reference (or kernel) over an image and computing a similarity measure at each position (known here as the cross correlation ratio (CCR)).

Calculation of the CCR first requires a single reference image, taken from a topograph where the scanning probe tip is in an ideal state. The principle of this method is to scan this reference image over an input image and measure how closely the reference image resembles the area of the input underneath it at each point. Because of this, care must be taken in choosing the reference image, as a poor choice could lead to inaccurate classifications, this is discussed more in Subsection 5.2.1.1.1. The chosen reference image should contain enough information so as to be able to capture a commonly appearing structure, such as a molecule adsorbed on the surface or a unit cell, whilst being as small as possible. A smaller reference image both reduces the chance of defects in the input image being contained within a highly correlated position, and increases the likelihood of multiple instances of the feature being found in the input image. For example, the reference image chosen for the B:Si(111) surface is an image of a single Si DB surrounded by six surface atoms (shown in Figure 5.9a). This structure appears as a bright feature around which six spheres are arranged in a hexagon. This was chosen for the B:Si(111) surface as the defect is a common feature, and was found to provide higher selectivity for identifying tip quality than a unit cell of the pristine surface. In cases where the tip is not ideal, the surface atoms can appear similar to when scanning with a "good" tip, whereas the DB defects often appear misshapen or doubled with the same tip. On the other hand, on the Si(111) - 7×7 surface we found the reference image of the six atoms surrounding a corner-hole was suitable (shown in Figure 5.8a). Using the chosen reference image, a cross-correlation feature map can be calculated using Equation 5.1 [14].

$$\gamma(i,j) = \frac{\sum_{x,y}[f(x,y) - \bar{f}_{i,j}][t(x-i,y-j) - \bar{t}]}{\sqrt{\sum_{x,y}[f(x,y) - \bar{f}_{i,j}]^2[t(x-i,y-j) - \bar{t}]^2}} \tag{5.1}$$

Where $\gamma(i,j)$ is the cross-correlation ratio at position (i,j), $f(x,y)$ is the input image, $t(x-i,y-j)$ is the reference image at position (i,j), $\bar{t}$ is the mean of

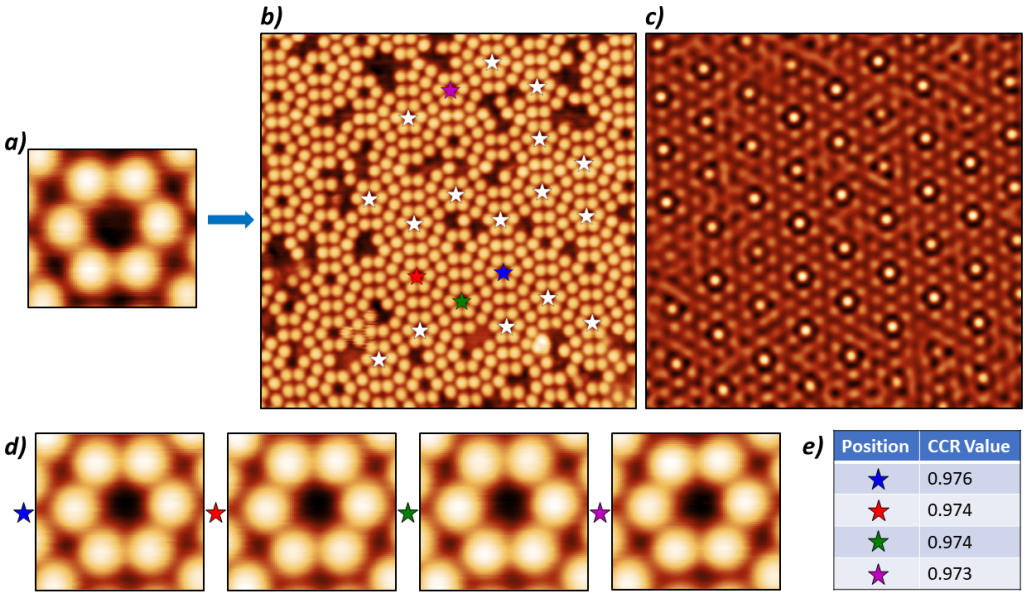


Figure 5.8: Cross-correlation method as applied to Si(111) - 7×7. a) shows the reference image used, in this case the chosen image is a tight square image surrounding a corner hole feature. b) shows an input image, over which the reference image will be scanned. Centered over each pixel, the reference image outputs a number between 0 and 1 describing how similar the area is to the reference image. The result of this is shown in the cross-correlation feature map in c). The stars overlaid on b) show the top 20 highest correlated positions, which correspond to the peaks in c). d) shows the top 4 highest correlated positions with the coloured stars corresponding to the same colours in b). e) shows the CCR values obtained for the areas shown in d).

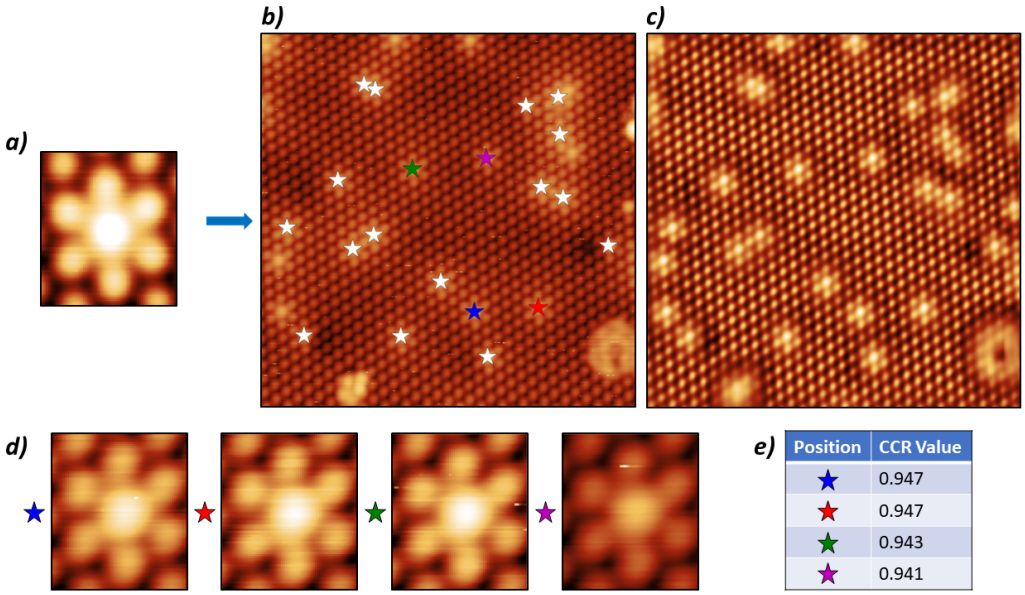


Figure 5.9: Cross-correlation method as applied to B:Si(111). a) shows the reference image used, in this case a dangling bond defect was chosen. b) shows an input image, over which the reference image will be scanned. The CC feature map is shown in c). The stars overlaid on b) show the top 20 highest correlated positions, which correspond to the peaks in c). d) shows the top 4 highest correlated positions with the coloured stars corresponding to the same colours in b). e) shows the CCR values obtained for the areas shown in d).

the reference image and $\bar{f}_{i,j}$ is the mean of the area of $f(x,y)$ underneath the reference image at position (i,j). This was used to generate the feature map, an example of which can be seen in Figure 5.8c. The reference image is scanned over the input image at every position with the reference centred on each pixel in the input image. The feature map produced will be of the same size as the input image, and comprises of pixels with values between 0 and 1, with 0 being no correlation at that position, and 1 corresponding to an exact match. Bright peaks can be seen in the feature map, corresponding to positions with high input image correlation to the reference image, i.e., areas which appear similar to the centre of a corner-hole in the example shown in Figure 5.8a. Figure 5.9 shows the same method as applied to the B:Si(111) surface.

The highest correlated positions can then be extracted from the feature map, each of which will have a corresponding value for the CCR at that position. By then taking an average of these, an overall value for the CCR can be obtained from the input image, which shows overall how closely it resembles the features present in the reference image. The number of values, N, can be chosen depending on how commonly the reference image feature appears in the system in question. Although a larger value of N avoids the chance of a spurious high correlation, for the datasets investigated here, $N = 1$ produced similar results to $N = 5$ and gives the advantage that only one instance of the feature needs to be present in the scan, which could be useful when evaluating surfaces where a high defect density could be present (in the case of the Si(111) - 7×7 with adsorbed water or other small gas molecules, for example).

Using the technique described above, one obtains a single numeric metric which describes how closely the image matches one taken with an ideal tip, and can therefore be used as a measure of the quality of the probe tip. For a given sample system, the CCR threshold for a "good" tip was defined empirically by running a small test set (around 20 images is sufficient) through the CC evaluation. This process only needs to be performed once for a given sample system, and can be easily modified later if it is practically found that the threshold is too strict or too lenient. We discuss the effect of the chosen threshold in more detail in Section 5.2.1.1.1.

5.2.1.1.1 Choice of Reference Image The choice of reference image is extremely important for optimal classification via CC, both in the choice of the image (and so specific tip state) to take a reference from, and the specific area chosen. To ensure we were able to choose the optimal reference for Si(111) - 7×7, three different reference image areas were trailed from three different "good" images. The three areas chosen were the corner hole feature including the surrounding six surface atoms, the complete unit cell containing twelve surface atoms and the right hand side of the unit cell containing six atoms. The images used and their histograms are shown in Figure 5.10.

Each histogram shown in Figure 5.10 is the result of calculating the CC of a section of the labelled training dataset, and plotting the number of counts for each CC value for manually labelled "good" and "bad" images in green and red respectively. Histograms such as these were used within this project to choose an overall threshold for classification, and as such, the reference image which appeared to show the best distinction between "good" and "bad" tips past a defined threshold was chosen as the optimal reference to be used in final classification.

This was analysed in depth for the Si(111) - 7×7 surface. Qualitatively, the variance in the tip state (between the three "good" tips considered) seemed to have little effect on the final classification. The different areas chosen for the reference have a larger effect, with the full unit cell showing significantly more "good" images at lower CC values, however the difference between the corner hole reference and half unit cell appears minimal. The final choice of reference was created from the image which, in the opinion of multiple operators, appeared to be taken with a sharper tip. The area chosen for the final reference was the corner hole feature with six surrounding surface atoms as this was the smallest of the two images and would be slightly more computationally efficient. The histogram for this is shown in Figure 5.10f and from this histogram a threshold of > 0.92 was chosen for "good" tips, balancing accuracy and precision. Final results of this will be discussed in Section 5.2.2.1.

5.2.1.1.2 Cross-correlation on Cu(111) Of the two adsorbate systems on Cu(111) attempted, we will first discuss the C_{60} system. The CC classification

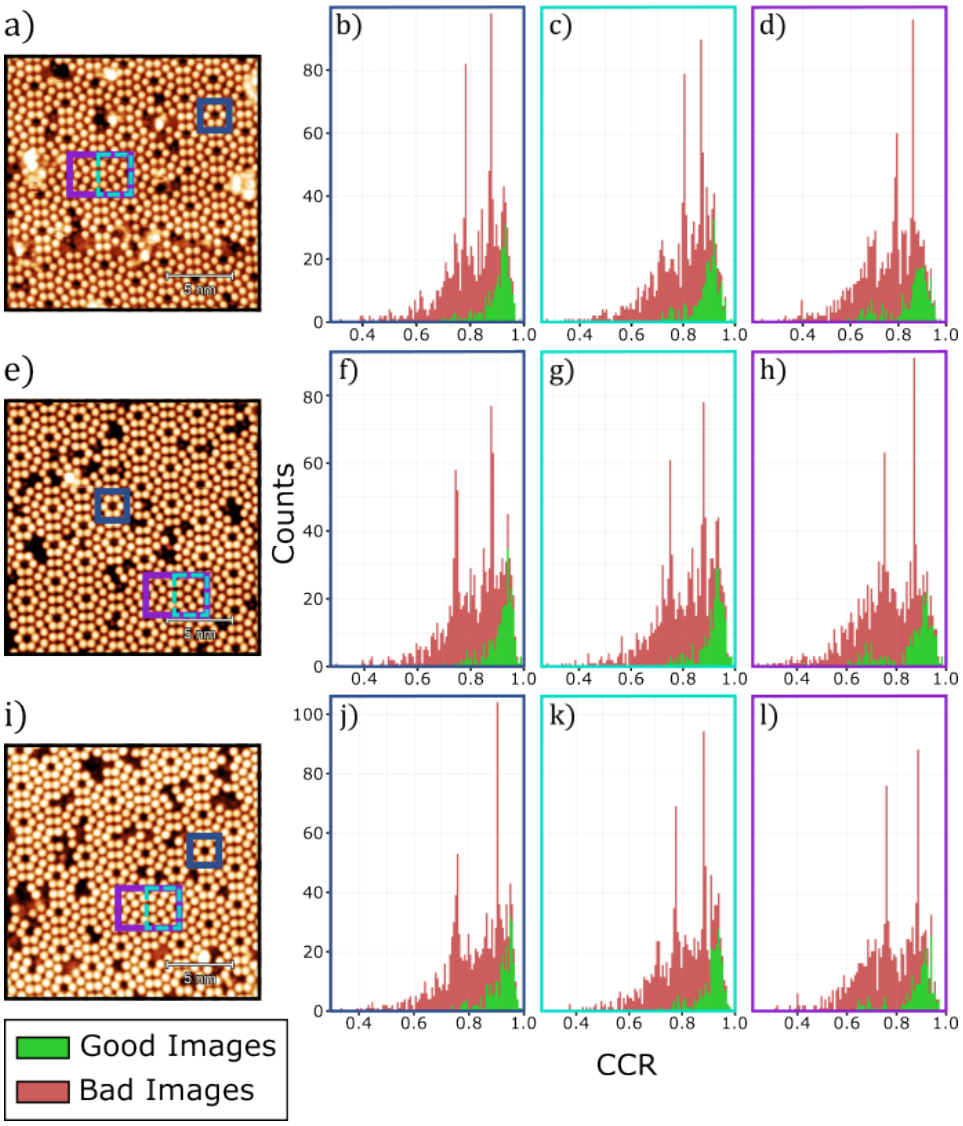


Figure 5.10: Results of the analysis of various options for CC reference image on a set of labelled images. a), e) and i) show different choices of scan, which were taken using different "good" tips, with coloured boxes corresponding to the different areas chosen as reference: dark blue shows a corner hole feature, purple shows the unit cell and light blue half the unit cell. These colours correspond to the histograms made from the reference images. References from a) resulted in histograms b)-d) and similarly f)-h) for e) and j)-l) for i).

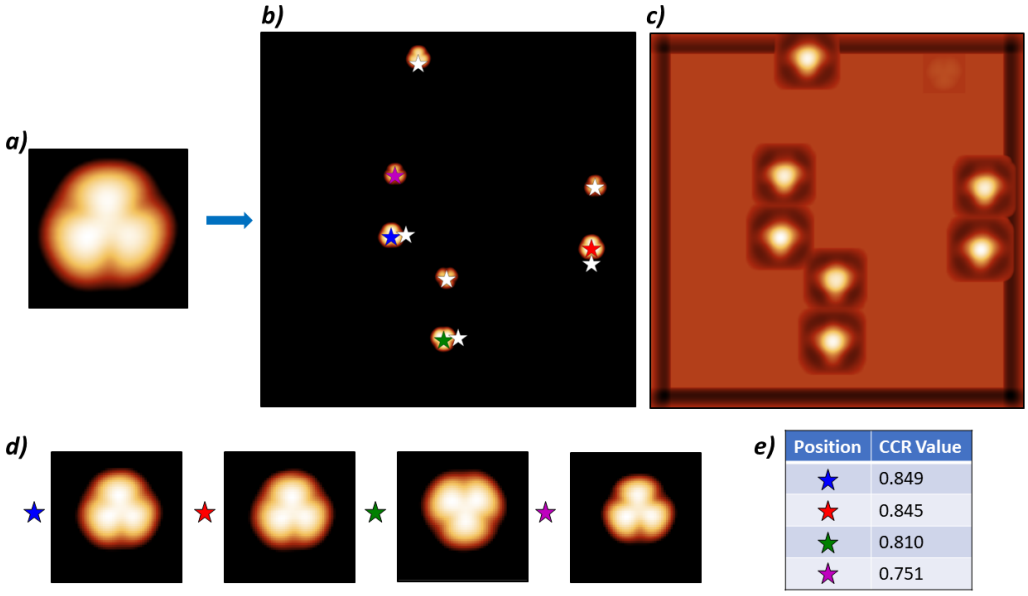


Figure 5.11: Cross-correlation method as applied to Cu(111) with a coverage of C_{60}. a) shows the reference image used, in this case cropping of a C_{60} molecule was used. b) shows an input image, over which the reference image will be scanned. The CC feature map is shown in c). The stars overlaid on b) show the top 10 highest correlated positions, which correspond to the peaks in c). d) shows the top 4 highest correlated positions with the coloured stars corresponding to the same colours in b). e) shows the CCR values obtained for the areas shown in d). The purple star indicated a C_{60} shadow which is found in addition to those scanned with the primary tip.

method can be applied well to this system by using a cropped image of a C_{60} molecules as a reference image as shown in Figure 5.11.

Unfortunately, due to the high aspect ratio the C_{60} molecules, the resultant topographs are more sensitive to the shape of the probe. This means that the height of the molecule can easily result in widely spaced multi-tip features due to tunnelling occurring with secondary apexes further up the shaft than might occur during imaging of an atomically flat surface or lower aspect ratio molecules. Features resulting from these secondary apexes are known as "ghost" features or "shadows". These features are also difficult to separate from the "true" features, as they can often appear similar to the main feature imaged by the primary tip, as can be seen in Figure 5.12. It is

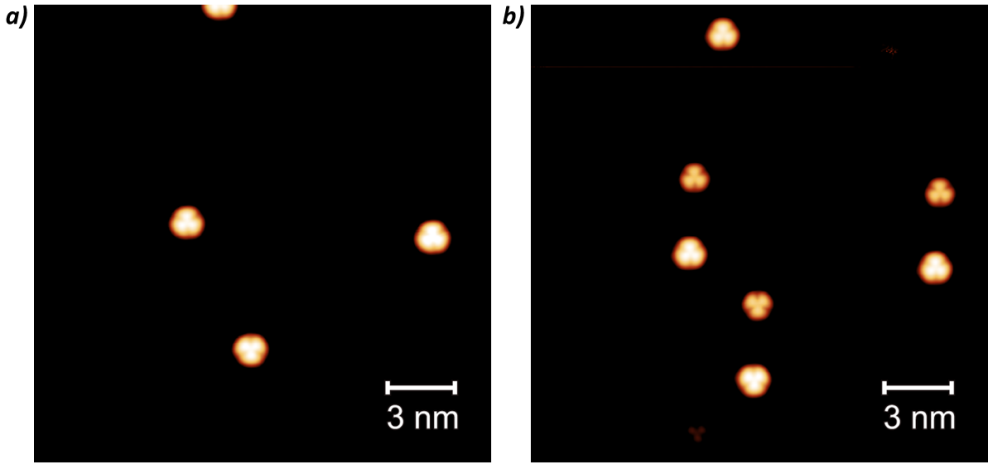


Figure 5.12: STM scans of the Cu(111) surface with a low coverage of C_{60}. The image was taken at 100 mV imaging bias and 100 pA setpoint. a) and b) show the same imaging area before, a), and after, b), a tip change, where "ghost" C_{60} molecules with a lower apparent height appear.

worth noting here than this problem may also occur in other classification models (even ML based) if trained on cropped images of individual high aspect ratio molecules.

The result is that when using the CCR method it is only possible to classify the state of the primary apex; secondary apex features may also be present in an image classified as "good" as shown with the shadows obtaining comparable CC values to primary tip features in Figure 5.11

The second Cu(111) adsorbate system used had a low coverage of both Cu adatoms and CO molecules. Figure 5.13a shows the appearance of this system, with the main tip shape identifying feature being the individual Cu adatoms. When the tip is sharp, these features appear as perfect circles, with differences in the shape of the tip resulting in small deviations in the roundness of this circular feature.

The CC based classification method was trialled on this system by using a

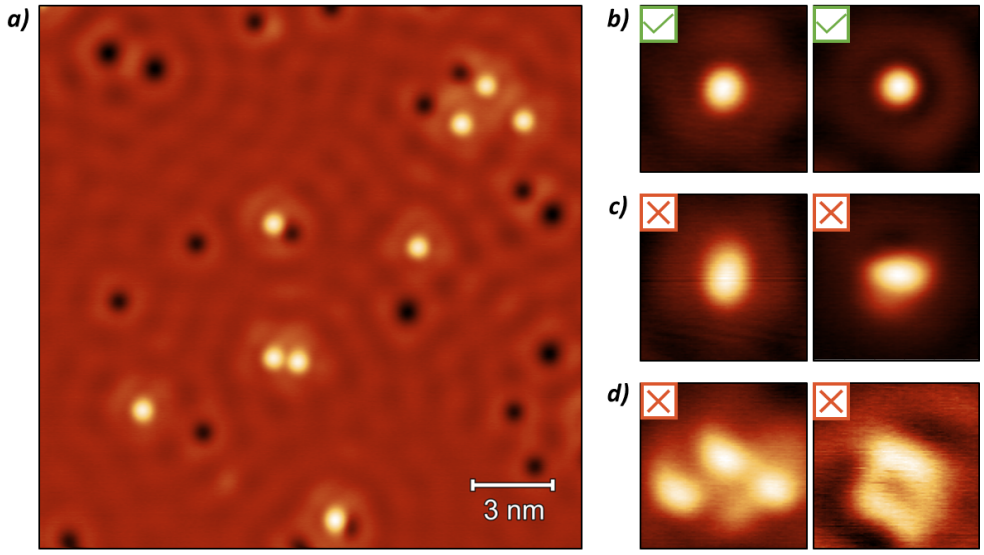


Figure 5.13: a) STM image of Cu(111) with a low coverage of Cu adatoms (bright protrusions) and CO molecules (depressions) taken at 5 K with an imaging bias of 100 mV and a 100 pA setpoint. b) two examples of Cu adatoms imaged with a "good" tip, showing a round appearance. c) two examples of Cu adatoms imaged with a slightly misshapen tip which would be classified as "bad". d) two examples of Cu adatom images attributed to extremely misshapen tips or tips with multiple apexes.

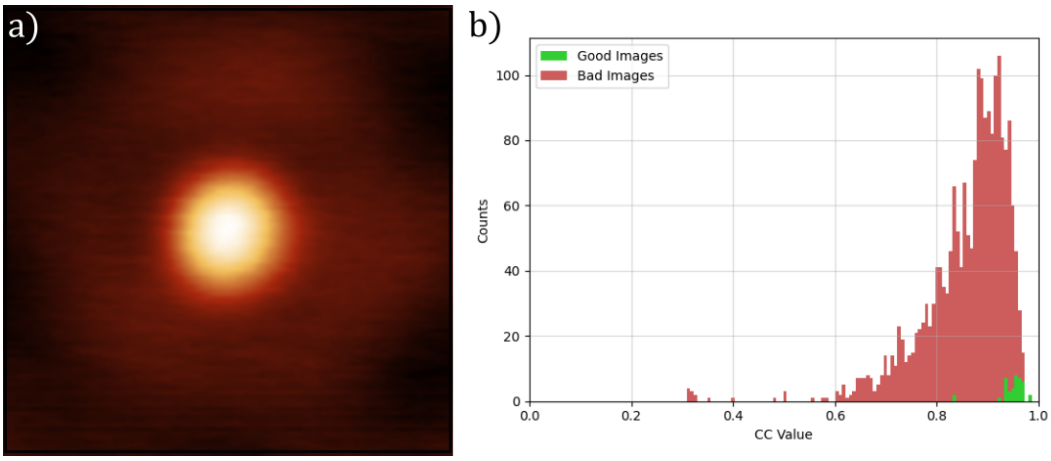


Figure 5.14: a) Cu adatom used as a reference image for the CC classification on the Cu(111) surface with a low coverage of Cu adatoms and CO molecules. The histogram showing the CC values obtained on a set of labelled images is shown in b).

small area around a Cu adatoms as a reference image (such as those shown in Figure 5.13b). The results from the CC attempt on this system are shown in Figure 5.14. This shows that chosen any threshold would result in a bad classification accuracy. Unfortunately, the CC method is not sensitive to small variations in the circular shape of the single adatom, and so an additional deterministic method was needed for classification of this system which will be discussed in the following section.

5.2.1.2 Circularity Measurement

An additional deterministic method was developed in order to allow for classifications of the Cu(111) surface using deposited copper adatoms as the comparison point. This method starts by using the CC algorithm to obtain the positions of the adatoms on the surface. Although this locates adatoms successfully, as reported above, we found that due to the lack of distinct features within in the adatom, CC was not able to make accurate classifications. We therefore introduced an additional image classification stage using the measured circularity of the adatoms.

The appearance of the adatoms is highly dependent on the shape of the

probe apex, with any irregularities in the tip causing the spherical shape of the adatom to appear deformed. Hence, we find it effective to measure the circularity of the adatoms for use as a metric in our classifier.

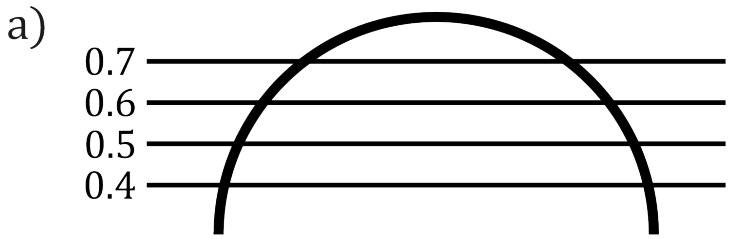
Once the adatom has been located on the surface using cross-correlation, the image is normalised to values between 0 and 1, and then thresholded to binarise the image, with all values above a specific value having a pixel value of 1 and below a value of 0. This is repeated for a four image thresholds (0.4, 0.5, 0.6 and 0.7), resulting in four output images for each adatom (see Figure 5.15). A range of image thresholds is chosen as each binarised image corresponds to the shape of the feature at different radii from the centre, so by taking an average of a range it is possible to check how spherical the feature appears. From here, the circularity is measured using Equation 5.2 for each image.

$$C = \frac{\sigma(r)}{\bar{r}} \tag{5.2}$$

Where $\sigma(r)$ is the standard deviation of the radius and $\bar{r}$ is the mean. This results in an output which measures how similar the feature is to a perfect circle, with a perfect circle measuring 0. The python library PyDip was used to measure the radius of the feature in each binarised image at various rotations. The average of the four circularity measurements is taken and used as the final metric used in the deterministic classification. Similar to the CCR metric, an image is classified as "good" based on a threshold which is discussed further in Section 5.2.2.1, the histogram used to define the threshold is shown in Figure 5.16

5.2.1.3 Fourier Ring Correlation

An additional technique trialled for tip state classification was Fourier ring correlation (FRC) [15], a method which aims to estimate the spatial resolution of an image, commonly used in electron microscopy. Typically, resolution is estimated by measuring the minimum resolvable distance between two features, however this requires manual input in choosing and measuring multiple features in the same image to improve the estimate.



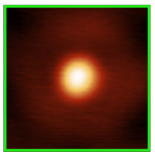
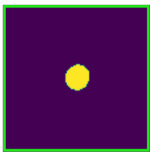
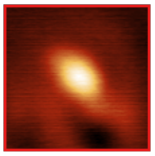
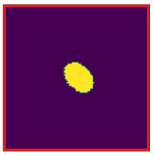
Threshold →		0.4	0.5	0.6	0.7
b) Good Tip					
Circularity	Avg = 0.030	0.029	0.034	0.036	0.021
c) Bad Tip					
Circularity	Avg = 0.44	0.214	0.378	0.516	0.658

Figure 5.15: Schematic showing the circularity measurement method. a) Example thresholding at different heights on a circular feature. b) and c) show examples of Cu adatoms on Cu(111), observed in STM with a "good" and a "bad" tip respectively. Both features are shown at the various binary thresholds used to calculate the circularity, with the output circularity values shown below each. The average value (below the leftmost image in a) and b)) shows the final value obtained for each tip.

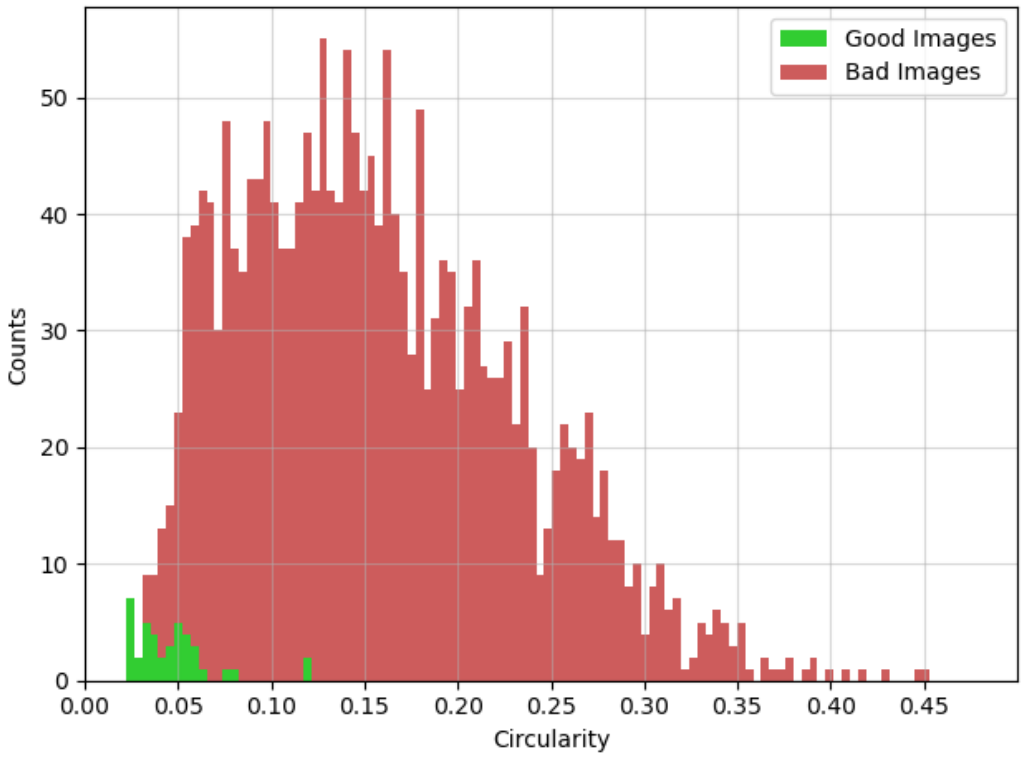


Figure 5.16: Stepped histogram obtained through deterministic circularity calculation of Cu adatoms on Cu(111).

FRC has also been used [16] to estimate image resolution in electron cryomicroscopy and optical nanoscopy, however the method is applicable to a wide range of methods including scanning probe microscopy (SPM). The method is based on calculating a cross-correlation histogram in frequency space of two images of the same subject area with independent noise realisations. The spatial frequency spectra of two images are split into bins of different frequencies, which in 2D frequency space corresponds to a series of concentric rings with frequency corresponding to the radius of the ring. The FRC histogram for each frequency bin is calculated using Equation 5.3 [16].

$$FRC(r_i) = \frac{\sum_{r \in r_i} F_1(r) \cdot F_2(r)^*}{\sqrt{\sum_{r \in r_i} F_1^2(r) \cdot \sum_{r \in r_i} F_2^2(r)}} \tag{5.3}$$

Where F_1 and F_2 are the Fourier transforms of two images and r_i is the ith frequency bin. The image resolution is defined from the obtained histogram as the inverse of the spatial frequency at which the CC curve drops below a pre-determined threshold value. In practise, the threshold value used can vary, however we found that the commonly used $1/7$ (0.143) threshold [16] works well in predicting a resolution which corresponds well to a rough qualitative analysis of the images.

Originally, FRC analysis would use two images of the same area, however in this use case it is not viable due to the time taken per scan. Koho *et al.* [16] introduced a method of calculating the FRC resolution from a single image. This method involves first splitting the input image into two pairs, shown in Figure 5.17a). The first pair is formed by taking every pixel with (even, even) and (odd, odd) row/column indexes, with the second pair formed from (even, odd) and (odd, even) indexes. The resultant sub-images will appear almost identical, with a size exactly half that of the input image. The FRC can then be calculated for both pairs of sub-images, and an average resolution calculated for the final FRC value of the image.

As can be seen in Figure 5.18, a weak correlation can be seen between the FRC estimated resolution and the quality of the probe tip, with no "good" tips appearing above an FRC resolution of around 120 pm. However, below this 120 pm boundary, numerous "bad" tips also appear. FRC was

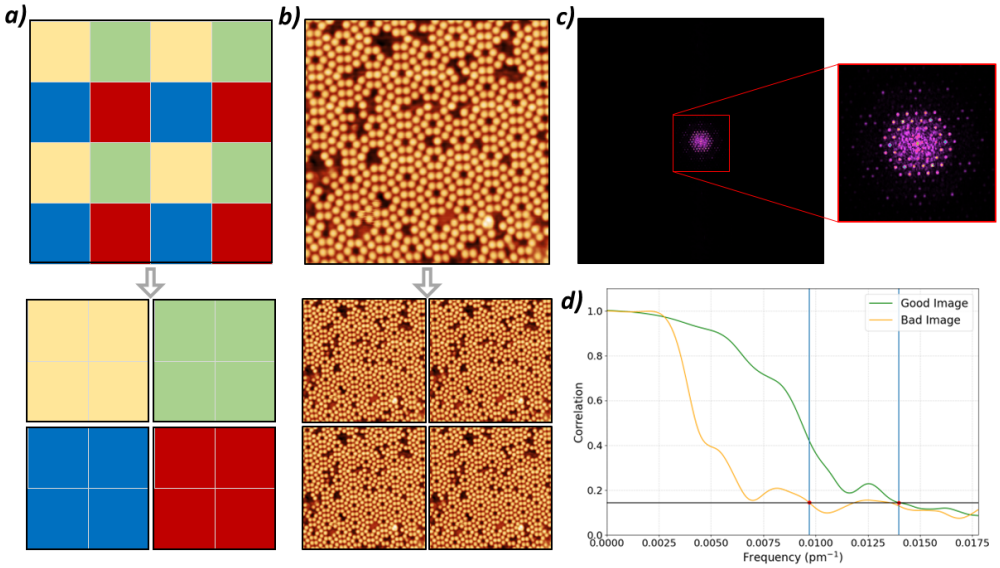


Figure 5.17: Figure showing the process of estimating the resolution of an image using FRC. a) First the input image is split into four sub-images by taking every other pixel on each axis. An example of this process, applied to a 4 x 4 pixel image of coloured squares, is shown. b) Shows the input image for a Si(111) - 7×7 sample with the 4 sub-images it is split into. For each of the sub-images, the Fourier transform of the image is obtained, an example of which is shown in c). The inset of c) shows a zoomed in section of the central portion where most of the detail is contained. From here, the cross-correlation is calculated between rings of frequencies in Fourier space between two of the sub images, this produces the plot shown in d). The FRC resolution is determined as the reciprocal of the point at which the resolution drops below a pre-defined threshold (vertical blue lines), this threshold is shown by a horizontal black line, here we use the commonly used 1/7 threshold. The green line in d) shows the FRC curve of a good image, and the amber shows a bad image.

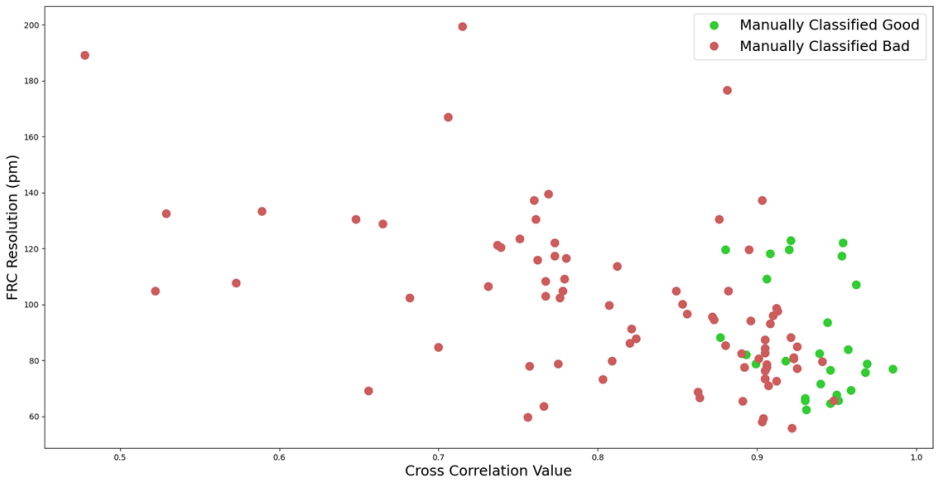


Figure 5.18: Scatter plot plotting the FRC resolution (y-axis) and CCR (x-axis) for various images taken with "good" (green) and "bad" (red) tips.

therefore discarded as a metric, as it offered worse selectivity than the CCR metric. The reasoning for its poor performance is that in SPM, a good spatial resolution is not always a good descriptor of a "good" tip. Specifically in the case of a multi-tip, it is possible to have multiple sharp apexes, meaning the resolution of the image will appear to be sharp, however the structure in the scan will not be representative of the true atomic or molecular structure of the sample.

5.2.2 Results

5.2.2.1 Personal Group Datasets

Using the CC based method described in Subsection 5.2.1.1, both the Silicon datasets were classified using suitable references images. The specific reference templates used for these datasets were a small cropping of the six surface atoms surrounding a corner-hole feature on the Si(111) - 7×7

dataset, and a cropping of the six surface atoms surrounding a dangling-bond defect on the B:Si dataset. To achieve a "good" classification, both datasets used a threshold of > 0.92, meaning any image with a CCR value of greater than this would be classified as a "good" tip.

Using this method, for the Si(111) - 7×7 dataset, our classifier was able to achieve an overall accuracy of 90%, with a precision of 97%. On the B:Si(111) dataset, the same classifier was able to achieve an accuracy of 89%, and a precision of 95%.

As mentioned in Subsection 5.2.1.1.2, the CC based method was not able to distinguish between "good" and "bad" probes on the Cu(111):CO/Cu dataset. However, using the circularity measurement method, classification of the probe state was still possible. Here, the circularity value a DB feature would need to be considered "good" was < 0.035. Using this threshold, the classifier was able to achieve an accuracy of 99%, and a precision of 81%. It should be noted here that for highly imbalanced datasets, such as this one, the accuracy metric is a poor measure, as simply classifying all images as the majority class would result in a high overall accuracy, therefore the precision provides a much more robust metric by which to assess the performance of the classifier. For example, the results here were obtained using the entire dataset of 2036 images, of which 1996 were classified by a human operator as "bad" and 40 as "good". Using this dataset, if all images in the set were classified simply as "bad", the final accuracy would still be 98%.

The final dataset obtained for use in classification was that of the Cu(111) surface with a low coverage of C_{60} molecules. As mentioned, this surface remained unlabelled, due to the fact that the dataset generation script, which relied on random tip changes resulting from *in situ* tip preparation techniques, also produced a very skewed dataset. Therefore, whilst the CC based classification scheme was able to make classifications on this set, a few caveats should be noted. Firstly, as discussed in Section 5.2.1.1.2, due to the high aspect ratio of the C_{60} molecules, only the primary apex could be classified. The second consideration to make is that due to this dataset not being fully labelled, it was not possible to obtain a complete overall accuracy in classification. It was however, possible to calculate the precision of the model in classifications made using the CC classifier, by manually labelling the images which the CC method classified as "good" for a given

	Si-7 × 7	B:Si	COCu	C_{60}
Accuracy	90%	89%	99%	—
Precision	97%	95%	81%	87%

Table 5.2: Accuracies and precisions obtained by the deterministic classifier on multiple datasets: Si(111) - 7 × 7, B:Si, Cu(111):CO/Cu, Cu(111):C_{60}.

threshold. For this dataset, it was found that a CCR threshold of > 0.99 produced the best results, obtaining a precision of 87% on primary apex identification. The reference image used here was a simple cropping of a C_{60} molecule as shown in Figure 5.11a.

All accuracies and precisions obtained using the deterministic method are shown in Table 5.2.

5.2.2.2 Open-source Datasets

We also considered the applicability of these methods to other surfaces which have previously been classified by ML methods in the literature, in each case commenting on whether the surfaces would suit CC based methods and, where possible, by applying our deterministic classifier to publicly available datasets.

In the work carried out by Alldritt *et al.* [17] ML was used to classify images of CO molecules to assess the quality of a CO functionalised tip, prepared using an automated tool. Using a convolutional neural network (CNN) the authors were able to achieve an overall binary accuracy of 95% and a precision of 90%. When imaging a CO molecule with a "good" CO functionalised tip, the molecule appears as a "sombrero" like feature, with a protrusion centred inside a ring-shaped depression. With this knowledge, and using the publicly available dataset used for this work, a "good" image of this "sombrero" feature was extracted and was chosen as a reference image. This reference image was then used to create a CC based classifier, which was able to achieve an accuracy of 62% and a precision of 99%. These results show that whilst the CC based method would disregard a larger portion of the "good" tips than ML, it would be more precise in its final

classification of a "good" tip which follows the trend we observed in our own data and supports the robustness of the method.

In the work of Krull *et al.* [9] ML was used to classify images of magnesium phthalocyanine (MgPc) on Ag(100) as part of an automated imaging tool. MgPc should be an ideal candidate for classification via CC, as when imaged with a "good" tip, the MgPc molecule has a distinct cross shaped appearance which would, in theory, allow for the CCR metric to distinguish between tip states. It should be noted that at higher coverages, the appearance of the molecule may change, with (for example) the formation of molecular islands. It would therefore be necessary to select a reference image that best reflects the appearance of the molecule on the surface for a given coverage. An attempt was made to classify the open-source data used by Krull *et al.* [9], however the very low resolution of the images in the available online data [9] for training prevented accurate classification due to a lack of detail.

In the work carried out by Rashidi *et al.* [8] images of DB on the H:Si(100) surface were used to train a ML based classifier to determine the state of the probe tip between two states: "sharp" and "double". The state of the probe tip on the H:Si(100) surface can be characterised in multiple ways, depending on the desired tip mode. The H:Si(100) surface appears as rows and the state of the tip is classified based on a DB defect in the structure, which appears as a diamond-shaped protrusion. It is possible that the CCR metric would be able to distinguish between a "good" and a "bad" tip using a cropped image of this defect as a reference image, however since the aim of their work was to distinguish specifically between "sharp" and "double" tip states, other, much simpler methods could be used to make this classification, such as thresholding of the image combined with a way of counting features present in the scan. Unfortunately, the data used in this paper is not open-source and so was not usable for a trail here.

5.3 Machine Learning Based Classifier

A ML based classifier was trained to be used as a comparison to the deterministic approach. A CNN classifier was chosen as they are commonly

used in image classification tasks, due to their ability to extract high level patterns from the input image.

To improve the performance of the training, images were augmented further using horizontal and vertical flips, as well as 90° rotations, increasing the size of the dataset by a multiple of 8. This helps to increase the amount of variance in the training dataset, which reduces the level of overfitting during the training process. Overfitting causes the training accuracy to increase, at the expense of the accuracy obtained on unseen data, and was discussed in detail in Section 2.2.2.6.

5.3.1 Results

5.3.1.1 Personal Group Datasets

For use in ML based classifiers, three datasets were obtained: Si(111) - 7×7, B:Si(111), and Cu(111) with a low coverage of CO and Cu adatoms.

For both silicon datasets, multiple CNNs were trained in order to find the optimal hyper-parameters for our specific dataset. In both cases, initially 30 models were trained varying the number of convolutional layers and kernels per layer ($1 - 5$ layers, and various kernal configurations within those). At this point, the configuration with the highest performance was used for additional hyperparameter training where the number of dense training layers and neurons per layer was varied from $1 - 3$ layers, and starting neurons of 32, 64, 128, 256, 512, 1024 doubling in successive layers.

The optimal architectures found were the same for both the B:Si(111) and the Si(111) - 7×7 classifiers. The structure used a total of five 3×3 convolutional layers (with 20, 40, 60, 80 and 100 feature maps respectively) with ReLU activation functions, each separated by 2×2 max pooling layers. The convolutional layers were followed by three dense training layers (32, 64, 128 neurons per layer in that order) using ReLU activation functions and a final binary output layer using a sigmoid activation function. The training layers were each separated by dropout layers (with probabilities 0.5, 0.3, 0.3 respectively) to reduce overfitting. The input to each network consisted of

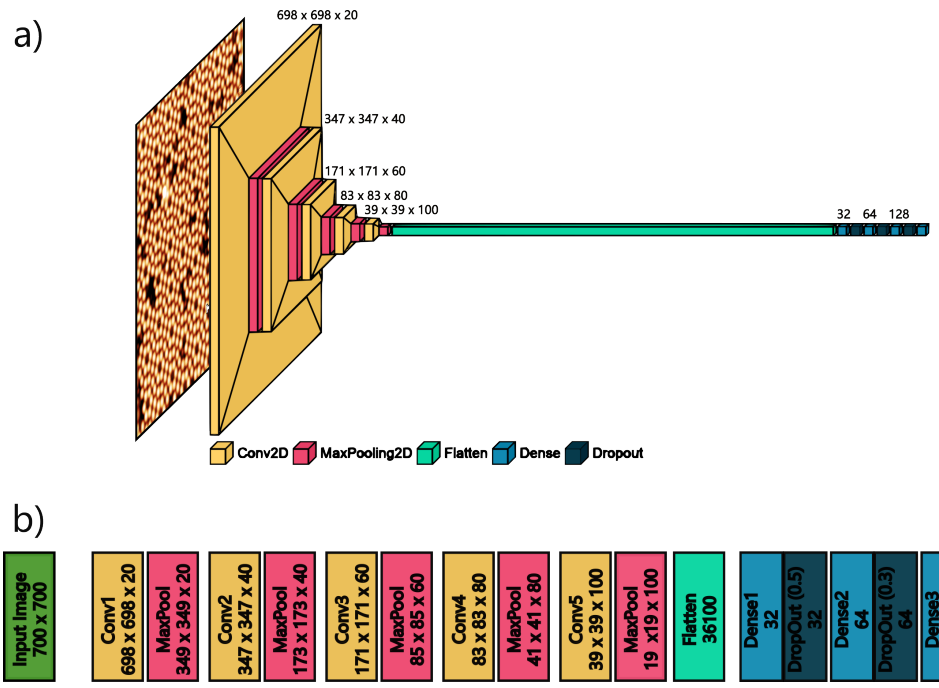


Figure 5.19: Architecture of the CNN model used for both the Si(111) - 7 × 7 and B:Si(111) datasets. It consists of a total of 18 layers: 5 consecutive convolutional-pooling layers, a flattening layer, 3 training-dropout layers, and ending with a single dense output layer.

700 × 700 pixel (19.4 × 19.4 nm^2) constant-current STM topography images. This model architecture is shown schematically in Figure 5.19.

Training was undertaken for a total of 100 epochs each, with the final weights and biases chosen being those which produced the highest overall precision. Figures 5.20 and 5.21 show the training progress for both datasets using the same model configuration. Figure 5.20 shows the accuracy (left) and precision (right) obtained on the training set (used to update the weights and biases during training) and the validation set (used to obtain a rough accuracy and precision) for the Si(111) - 7 × 7 dataset. These accuracy and precision graphs are used primarily to assess whether the models have overfit to the training data; if overfitting has occurred, the validation curves will be significantly lower than the training curves, indicating that the training has allowed the model to correctly classify the training data but not

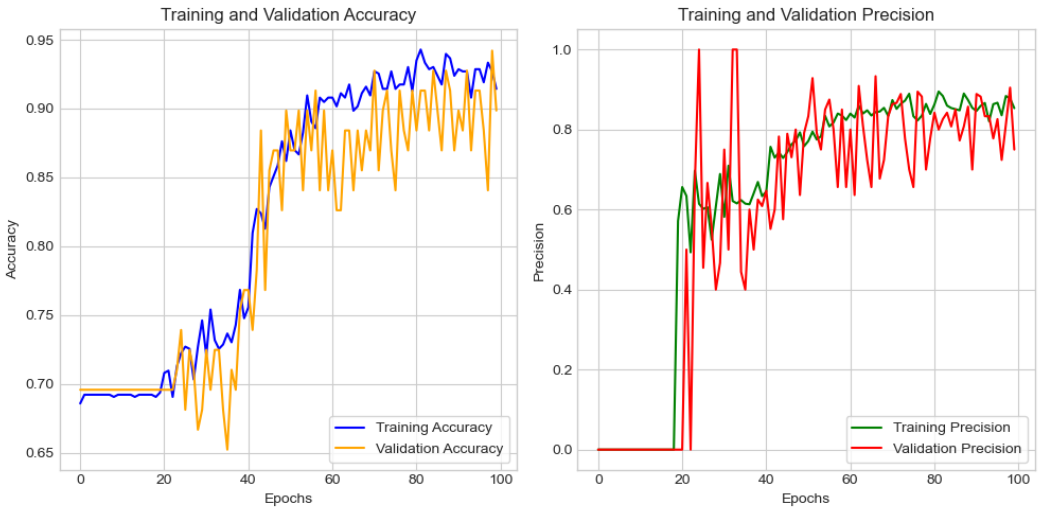


Figure 5.20: Graphs showing the progression of the accuracy (left) and precision (right) during 100 epochs of training on the Si(111) - 7×7 dataset. Results on the training dataset are shown in blue and green, and validation curves are shown in yellow and red.

the validation.

Figure 5.20 shows that both validation and training curves are following the same trend for the accuracy and precision, meaning our model has fit correctly. Values shown on these curves are only estimates, the final accuracy and precision for each model is obtained by evaluating the model over a separate dataset known as the test set, which here consisted of 20% of the entire dataset separated out before training.

Similarly for the B:Si dataset, Figure 5.21 shows the accuracy (left) and precision (right) curves over 100 epochs. By comparing the training and validation curves for each, it can be seen that the model shows little overfitting.

Final accuracies and precisions for each model were obtained, as mentioned previously, using the separated test set, which achieved the results shown in Table 5.3.

The final dataset considered was the Cu(111) surface with a low coverage

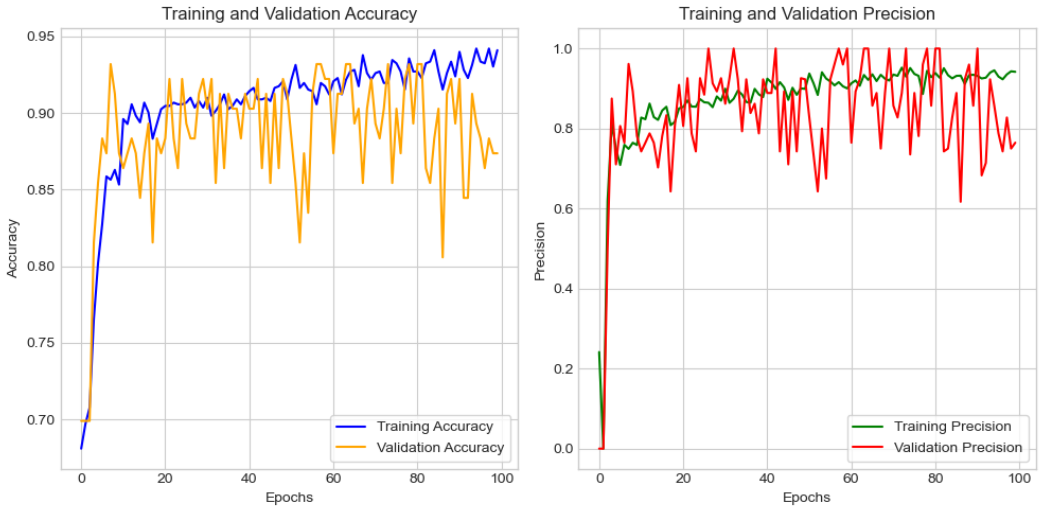


Figure 5.21: Graphs showing the progression of the accuracy (left) and precision (right) during 100 epochs of training on the B:Si dataset. Results on the training dataset are shown in blue and green, and validation curves are shown in yellow and red.

	CNN	
	Si-7 × 7	B:Si
Accuracy	96%	90%
Precision	**92%**	**97%**

Table 5.3: Table showing the accuracy and precision obtained using CNNs on the silicon datasets at room temperature.

	Class Weighting	Down-sampling	Combination
Accuracy	Did not train%	99%	52%
Precision	Did not train%	0%	1%

Table 5.4: Results of trained models on the Cu(111) dataset with different augmentations to the dataset in order to attempt to train a model.

of CO molecules and Cu adatoms. As mentioned, the dataset itself was very highly biased toward the "bad" tip state category, with 98% of the images being in this class. Due to this, training was not as straight forward; additional methods which were not used for the other datasets were needed to train any model at all. The first method attempted was to add class weighting to the model, which results in specific classes having a larger effect on the loss function in training. The "good" class was therefore weighted at a factor of 49 compared to the "bad" class. The second method attempted was to down-sample the dataset, which involves removing a portion of the most frequent class in order to balance out training set to some extent. In the silicon dataset, our split was roughly 70% "bad" images, with 30% classified as "good", which seemed to train well. Therefore in down-sampling, we attempted to replicate this split for the copper dataset, reducing the total number of "bad" classified images from 1592 to 86, which obviously reduces the overall size of the training set enormously. Three runs were attempted in total, using the same model architecture as was used for silicon, which utilised class weighting, down-sampling and a combination of the two, resulting in the accuracies and precisions shown in Table 5.4.

The results clearly show that no sensible training was completed during any of the runs attempted, with the model utilising only class weighting not being able to train at all. The model using solely down-sampling appears to predict all images as "bad", whereas the final model using a combination of the two made some attempt at learning, however the results show that this dataset was clearly not sufficient to train a ML based classifier.

Additional analysis can be carried out on trained ML models in order to determine the overall performance, mainly in calculating the receiver operator characteristic (ROC) curve. The ROC curve is calculated by using a model to make classifications on a test set, which, for a binary classifier, result in predictions in the form of confidence values between $0 - 1$. By then

varying the confidence threshold at which a classification is assigned to a specific class (if 1 is "good" and 0 is "bad", commonly a simply rounding threshold would be used, meaning anything above 0.5 would result in a "good" classification) and calculating the true positive and false positive rates at each threshold, the ROC curve is produced. These curves are shown in Figure 5.22, where the area under the ROC curve (AUROC) corresponds to the performance. A perfect classifier will result in an AUROC of 1, whereas random guessing between the two classes would results in a value of 0.5 which is indicated by the dashed diagonal in the figure. The test sets used to calculate these curves were of sizes 174 for the Si(111) 7×7 model, 259 for the B:Si model and 407 for the Cu(111) model. Unfortunately, for the latter test set, the total number of "good" images was only 3, resulting in a poorly calculated curve with only four distinct steps, resulting in a poor AUROC calculation.

Even with this dubiously calculated AUROC value of 0.65, the results point toward the training having had little effect. The AUROC values calculated for both the Si(111) - 7×7 and B:Si models, on the other hand, show a value of 0.98, suggesting a robust predictive performance.

5.3.1.2 Open-source Datasets

Similarly to when analysing the deterministic classifier (Subsection 5.2.2.2), we can compare our ML based models to those trained by other groups in the literature [9, 17], who have also attempted to tackle the problem of tip state recognition, and have allowed for their training data to be open-access. Using these datasets, we can attempt to train similar models to our own using their data, with the same model architecture as was used on the silicon datasets discussed in Subsection 5.3.1.1.

The dataset used by Krull *et al.* [9] contains a total of 7382 images, classified into binary "good" (1761) or "bad" (5621) tip states. Initial image sizes here were 64×64 pixel images which meant that to allow for us to use our model architecture, the images needed to up-scaled to 700×700 (as in our models) so as to not completely reduce the data before training in the convolutional and max-pooling layers which both reduce the image size. Results from both the training and validation set are shown in Figure 5.23 with final

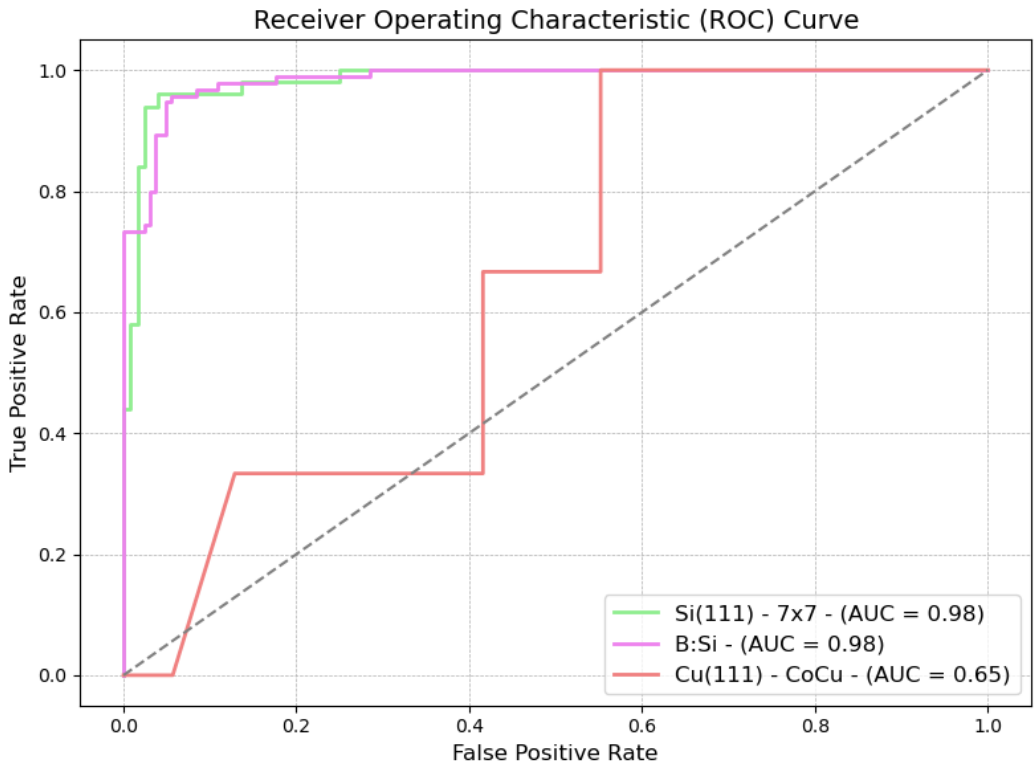


Figure 5.22: ROC curves for three CNN trained models on green: Si(111) - 7×7, pink: B:Si, red: Cu(111) - COCu and grey dashed: random guessing. AUROC values are given for each, with an perfect value being 1 and random guessing achieving 0.5.

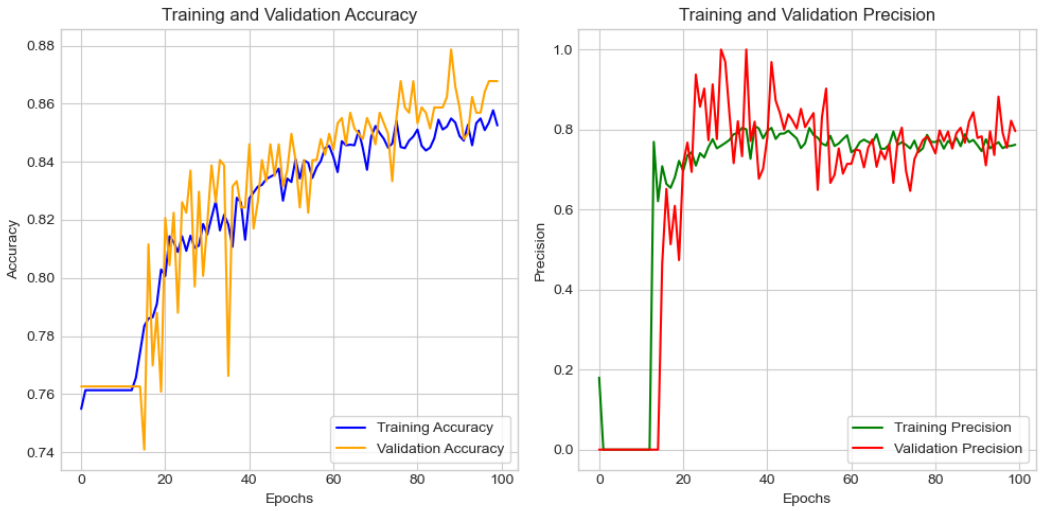


Figure 5.23: Graphs showing the progression of the accuracy (left) and precision (right) during 100 epochs of training on the dataset from Krull *et al* [9]. Results on the training dataset are shown in blue and green, and validation curves are shown in yellow and red.

accuracies obtained from validation datasets resulting in a final accuracy of 85% and a precision of roughly 69%.

The dataset used by Alldritt *et al.* [17] contained a total of 391 images of individual CO molecules imaged with a CO functionalised tip, taken from a total of 66 overview scans (and therefore 66 different tip functionalisations). These overviews were classified, again, into binary "good" (45) and "bad" (21), from which the individual CO molecule images were taken, resulting in a final dataset of cropped images containing 230 "good" and 161 "bad" images. The raw dataset consisted of images of sizes 21×21 pixels, meaning that we needed to up-scale these to 700×700 pixels for use with our network. Training and validation curves for the accuracy and precision are shown in Figure 5.24, which resulted in a final accuracy of around 95% and a precision of 95%

Similarly to the analysis completed Subsection 5.3.1.1, an ROC curve can be can calculated for the two models described above, shown in Figure 5.25.

These curves result in a calculated AUROC of 0.98 for the Alldritt model, and

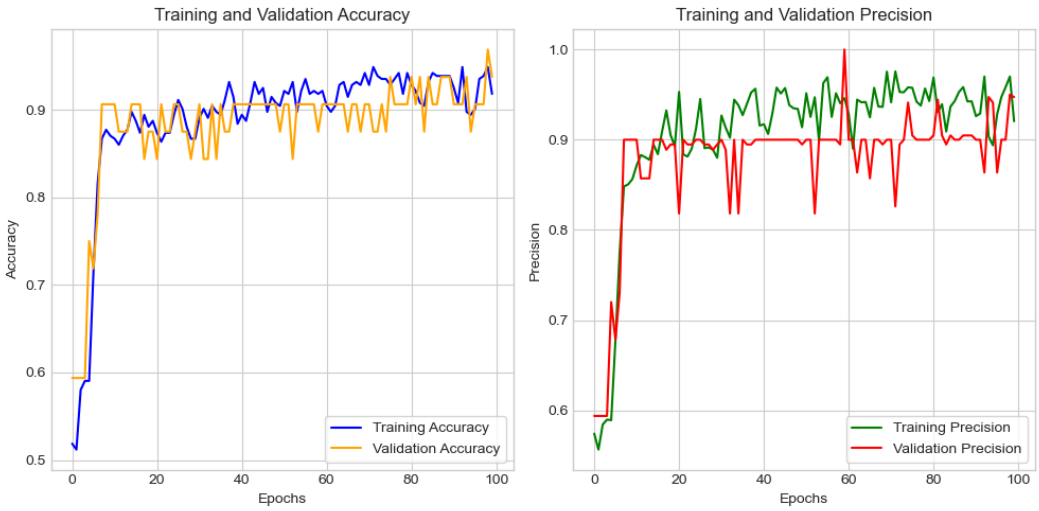


Figure 5.24: Graphs showing the progression of the accuracy (left) and precision (right) during 100 epochs of training on the dataset from Alldritt *et al.* [17]. Results on the training dataset are shown in blue and green, and validation curves are shown in yellow and red.

0.85 for the Krull model. Both models are therefore fairly robust, however the predictive capability of the model trained on the Krull (85% accuracy) is not as good as the same model trained on our datasets, with an average accuracy of 93%.

The dataset of Gordon *et al.* [4] contains a total of 18931 images, classified into a total of six categories of image resolution, which depend on the sharpness of the probe: atoms, asymmetries, dimers, rows, tip-change and generic defect. This dataset would have been another good example of training a different dataset using our model architecture, however, the labels attached to these data were not given in a clear format, and with no accompanying description of the labels, the dataset was not usable so we were not able to train a ML model here.

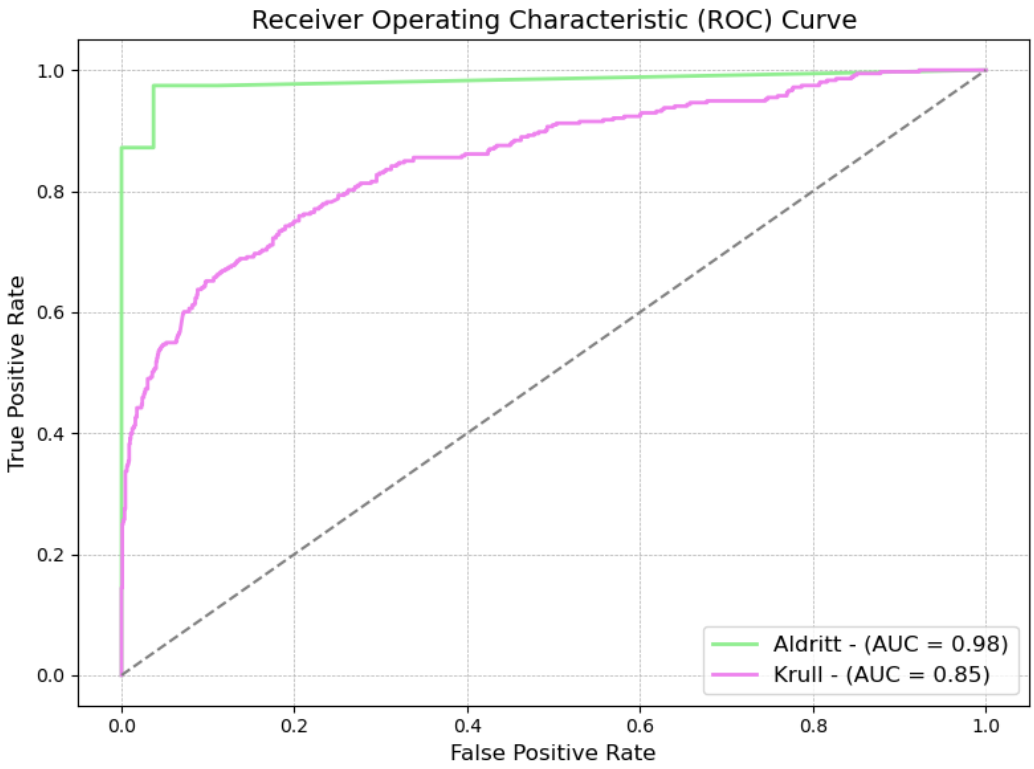


Figure 5.25: ROC curves for two CNN trained models trained on the open-source datasets: green: Alldritt *et al.* [17], pink: Krull *et al.* [9] and grey dashed: random guessing. AUROC values are given for each, with an perfect value being 1 and random guessing achieving 0.5.

	CC		CNN		Operator	
	Si-7 × 7	B:Si	Si-7 × 7	B:Si	Si-7 × 7	B:Si
Accuracy	90%	89%	96%	90%	92%	95%
Precision	**97%**	**95%**	**92%**	**97%**	**93%**	**88%**

Table 5.5: Table showing the accuracy and precision for multiple tip state classification methods: CC classifier, ML based CNN and manual classifications carried out by an operator.

5.4 Comparison of Results

The results for the various methods of classification for the two silicon datasets are shown in Table 5.5. The CC based classifier was able to achieve comparable results between the two silicon surfaces, with an overall slightly higher precision compared to the accuracy. This slightly lower overall accuracy incurred by the CC classifier is almost entirely due to the misclassification of "good" tips as "bad" (as is shown by the high precision) and so, when applied to an automated tip preparation tool would only contribute to slowing the overall process of exiting with a "good" tip. The ML model shows very similar results, with comparable accuracies and precisions for each surface. When compared to a human operators classification, which should be considered to be the ground truth of any classification, we again see very similar results. For the Si(111) surface, the standard deviation between the three classification modes is 2.5% in accuracy and 2.2% in precision, with similar results being found for the B:Si surface (acc: 2.6%, prec: 3.9%).

Given the similarity in performance between both models, we note the main apparent advantage to using deterministic image analysis techniques for classification versus ML methods is the significantly reduced overhead in their creation. As noted, sufficiently large datasets required for ML are not always available, as was the case for this study, which necessitated the development of an automated dataset generation script. In addition to the large datasets, manual labelling needs to be carried out on these datasets which is a very time consuming process, and requires careful forethought and trained microscopists to obtain adequate labels. When compared to large labelled datasets used in other fields, it is not possible to outsource the

	RW [8]	GM [4]		Krull [9]	Alldritt [17]
	H:Si(100)	H:Si(100)	Au(111)	MgPc/Ag(100)	CO/Cu(111)
Acc	97%	93%	91%	94%	95%
Prec	Not Given	96%	97%	87%	90%

Table 5.6: Table showing the accuracy and precision of multiple ML based binary classification attempts in the literature on various surfaces. The Rashidi-Wolkow (RW) [8], Gordon-Moriarty (GM) [4], Krull [9] and Alldritt [17] networks all use convolutional neural networks.

process of labelling a dataset, primarily due to the instrumental expertise and physical understanding required to make the distinction between "good" and "bad" tips. Conversely, the CC based methods require a much smaller dataset, comprising a single "good" sample image, and no time is needed for training. A result of this is that when trying to make classifications on a different system, or even the same system with different imaging parameters (which can result in a different appearance of the surface), much less time and effort is needed when using CC based methods. Whereas ML would require an entirely new dataset to be trained and labelled, CC would still require only a single image.

Additionally, the results from the presented ML networks show comparable accuracy and precision to previous attempts published in the literature, as shown in Table 5.6, the model architectures of which are shown in Table 5.7.

The average accuracy and precision obtained in previous work (shown in Table 5.6) was 94% and 93% respectively, compared to the averages of 93% and 95% obtained in our ML attempts. We note here that direct comparisons between ML networks trained on entirely different datasets are difficult to make due to the variability in the datasets themselves (differences include size, variability of features, data preprocessing). Therefore caution should be taken when making quantitative comparisons between different ML based classifier's results. Nevertheless, the results indicate that the ML method presented here is able to classify tip states with high accuracy and precision, and performs similarly to other ML SPM image classifiers [4, 8, 9, 17].

From table 5.8 it is clear that using our model architecture was not universally successful. The model trained using the dataset from Krull in our

Model	Input Shape	Conv Layers	Dense Layers
RW [8]	(28, 28, 1)	Conv(30, 5×5, ReLU) Conv(40, 5×5, ReLU) MaxPool(2×2)	Dense(128, ReLU) Out(1, Softmax)
Krull [9]	(64, 64, 1)	Conv(64, 3×3, ReLU) Conv(128, 3×3, ReLU) Conv(256, 3×3, ReLU) Conv(512, 3×3, ReLU) MaxPool(2×2) Conv(64, 3×3, ReLU) Conv(128, 3×3, ReLU) Conv(256, 3×3, ReLU) Conv(512, 3×3, ReLU) MaxPool(2×2) Conv(64, 3×3, ReLU) Conv(128, 3×3, ReLU) Conv(256, 3×3, ReLU) Conv(512, 3×3, ReLU) Conv(64, 3×3, ReLU) Conv(128, 3×3, ReLU) Conv(256, $3x \times 3$, ReLU) Conv(512, 3×3, ReLU)	Dense(4096, ReLU) Dense(4096, ReLU) Out(12, Softmax)
Alldritt [17]	(16,16, 1)	Conv(4, 3×3, ReLU) Conv(4, 3×3, ReLU) AvgPool(2×2) Conv(8, 3×3, ReLU) Conv(8, 3×3, ReLU)	Dense(32, ReLU) Out(1, Softmax)

Table 5.7: Literature model architectures used to obtain the metrics shown in Table 5.6 for Rashidi-Wolkow (RW) [8], Krull [9] and Alldritt [17] models. The architecture for the Gordon-Moriarty [4] model was not specified.

	Literature Model		Our Model	
	Krull [9]	Alldritt [17]	Krull	Alldritt
Accuracy	94%	95%	85%	95%
Precision	87%	90%	69%	95%

Table 5.8: Final results for training the models of the same architecture used in the ML results shown in Section 5.3.1.1 with open-source datasets from Krull [9] and Alldritt [17].

architecture shows significantly reduced values for both accuracy and precision, which is possibly due to the extremely small size of the input data compared to the features being observed. Individual images in this dataset consisted of a 64×64 pixels, which may be too few to adequately capture the variance in the shape of the cross-shaped molecules with different tips. This does not mean that the model is unsuitable, however the architecture used here in clearly not optimal for this dataset.

The model trained using the Alldritt dataset on the other hand resulted in comparable values for both accuracy and precision, with the latter being slightly improved. The input images in this dataset had an original pixel density of 21×21. The main advantage this dataset had was the simple structure being observed, with CO molecules appearing as "sombrero" like features, the variance of which can be captured well within the few pixels used.

It is worth noting the high degree of optimisation needed to obtain the best values for training a ML model. ML is often thought of as a straightforward process, with the only requirements being the labelled dataset and access to a computer. In reality, however, a lot of understanding is needed to be able to start training a ML model, and optimising the hyper-parameters for a model is not a trivial task, requiring repeated training with small variations between attempts with a large parameter space to make adjustments in. This is another significant downside to using ML as a whole, and hence an additional advantage to using deterministic classification techniques where possible.

5.5 Applicability and Limitations

The ability of the described deterministic method to make accurate classifications on multiple surfaces using only a single image of the surface being studied makes it a valuable tool, however there are some limitations which should be noted.

The main metric used is the CCR, which relies on a repeating structure being present on the surface being studied. Whilst this is the case for a very large number of surfaces studied using atomic resolution STM, including the systems presented here, there are situations where nothing on the surface could be used as a reference image. For example, at the imaging parameters used in Figure 5.5c, the atomic structure of the Cu(111) surface is not resolved, with only the electron standing wave patterns being visible. These standing wave patterns can vary a large amount, depending on localized scattering potentials from features such as atomic step edges and adsorbates. We do note in passing that it is also difficult for a human operator to assess the quality of the tip from an image of this type.

Conversely, where the appearance of the atomic resolution of the surface changes with bias (as is the case for positive and negative bias images of Si(111) for example), the deterministic model can be "retrained" to assess images at a different bias, using only one example image; to retrain an optimal ML model may require an entirely new training dataset and reclassification, depending on apparent topography difference between these biases. In addition to the need for a repeating structure, the chosen structure needs to contain sufficient detail such that a measurement of the CCR will differ between tip states. For example, the CCR measurement was not suitable for use on the Cu(111) surface with a low coverage of Cu adatoms and CO molecules (when using a tight crop of either adsorbates as a reference image), due to the fact that the ovular appearance of the adatoms/CO molecules (Figure 5.13c) caused by slightly misshapen tips would still give high CCR values when compared to a "good" tip reference image. This resulted in the addition of the circularity measurement metric, which combined with feature finding through CC was able to accurately classify the state of the probe tip. It seems likely therefore that a very broad

range of atomic and molecular scale structures are amenable to classification via a combination of deterministic methods.

We also considered the applicability of the deterministic methods to other surfaces which have previously been classified by ML methods in the literature, in each case commenting on whether the surfaces would suit CC based methods and, where possible, by applying our CC-based classifier to publicly available datasets. This work was shown in Section 5.2.2.2, where, when the CC method was able to be applied, as in the dataset from Alldritt *et al.* [17], comparable results were found to their ML counterparts. Additionally, for the other surfaces, mainly Krull *et al.* [9] and Rashidi *et al.* [8], it seems likely that deterministic methods would be able to achieve good classification schemes.

There are also cases in the literature in which surfaces have been imaged which would not be suitable to classification via deterministic methods, such as the work carried out by Gordon *et al.* [4]. In this work, the authors imaged the Au(111) surface without atomic resolution and in the absence of adsorbates, which shows only the characteristic herringbone structure. Similar to the standing wave pattern visible on the Cu(111) surface, the herringbone structure shows no specific regular features which could be used as a CC reference image and so ML based methods seem necessary.

5.6 Automated Tip Preparation Tool

Once a computationally based tip state classification scheme has been produced, it is possible to implement an automated STM tip preparation tool. This tool, similar to the one used for dataset generation, was implemented in LabVIEW, interfacing with Python. This is schematically shown in Figure 5.26, and works by repeatedly obtaining topographs of the surface and making classifications of each image. If the topograph is classified as "bad", the system moves a pre-defined distance away from the scan area and attempts to condition the tip *in situ*. Manual *in situ* tip preparation involves combinations of two processes: bias pulses applied to the tip which facilitate the ejection of matter from its surface, and indentations of the tip into the surface in an attempt to refine and sharpen the tip through the detachment

or attachment of matter on the tip apex. To emulate this manual procedure, the shaping events themselves were chosen beforehand to increase in magnitude over successive attempts, in order to reduce the chance of getting stuck in an "energetic minima" of a blunt but robust tip.

The distance the tip moves away from the imaging site varies, but is usually set to around 200 nm to ensure the imaging area is not affected by the shaping events as they can cause the scattering of contaminants in the area. In addition, the script counts the number of attempts taken so far in preparation, and when the counter exceeds a predetermined threshold, the tip will move macroscopically away from the current scan area by stepping away using the coarse motor. This is done because if the area is damaged, or otherwise unsuitable, the automated tip preparation tool will be unable to identify the tip as ever being "good".

This tool was implemented using the deterministic classifier and proof of principle experiments were performed on the Si(111) - 7×7 surface. In a trial of 20 runs, the automated tip preparation tool was able to prepare a tip from "bad" to "good" after an average of 12 shaping events, corresponding to approximately 10 minutes of imaging time at the scan size and speeds chosen here (scanning a 20×20 nm^2 area with a pixel density of 256×256 and a scan speed of 76.8 ms/line). This tool allows for the very time consuming and tedious process of tip preparation to be completely automated, leaving the user to use their time elsewhere.

An example of a tip preparation run using this tool is shown in Figure 5.27.

5.6.1 Rotation Correction

It is common for samples to be aligned rotationally different from one another, as shown in the molecules visible in Figure 5.12, which could cause a problem when using a CC based classification scheme as the reference image would be oriented differently from similar features on the surface. To overcome this problem, an additional method was used to correct for this rotation before the CCR is calculated for any given scan. This method works best for rotations of less than 10°, and so requires that at the start of the

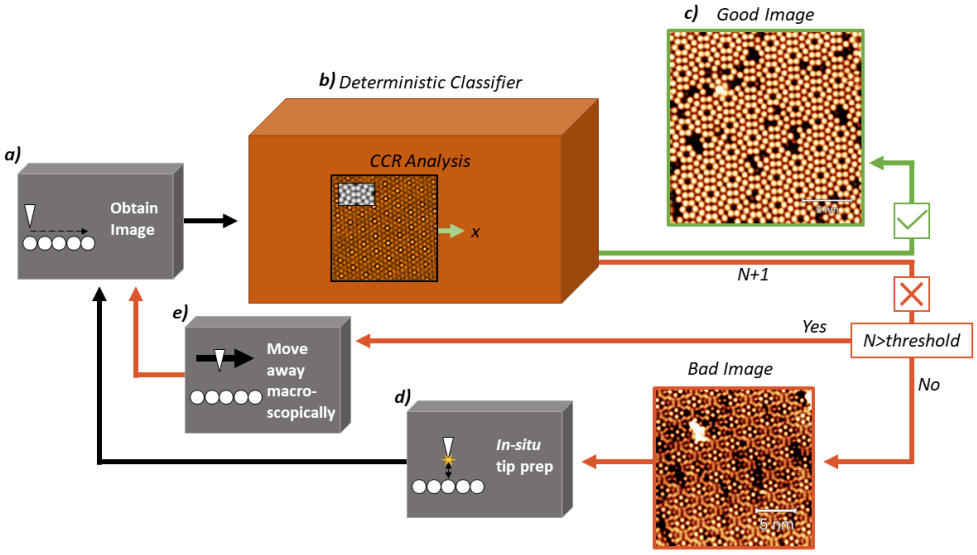


Figure 5.26: Schematic of the automated tip preparation tool. The process starts, a), by scanning an area to obtain an image of the surface. The image is processed to make the it appear flat and to remove the bottom 20 lines from the scan to avoid visible creep before it is then classified, b), using a cross-correlation based analysis script which compares the input image to a reference and outputs an estimated binary classification of "good" or "bad". If the tip is classified to be "good", c), the script then exits, however, if the tip is classified as "bad" the script will move away and attempt to re-prepare it, d). If the tip has already been through a set number of shaping events at this point, the script will instead reposition the scan area away macroscopically using the coarse motor, e), under the assumption that the area being scanned is not suitable to classify the tip.

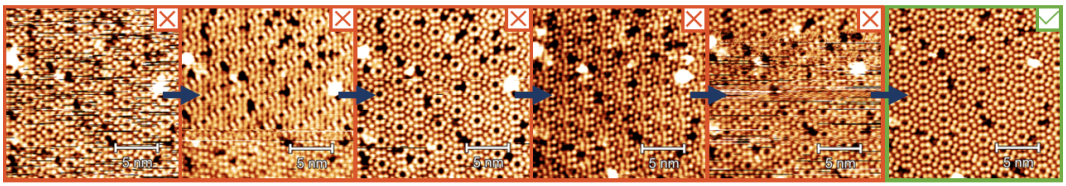


Figure 5.27: Representative sequence of constant current images showing the Si(111) - 7×7 surface at 2 V, 100 pA. Images showing the operation of the automated tip preparation tool which was able to prepare the tip from an initial "bad" state (far left) to a "good" tip (far right) in 5 shaping attempts.

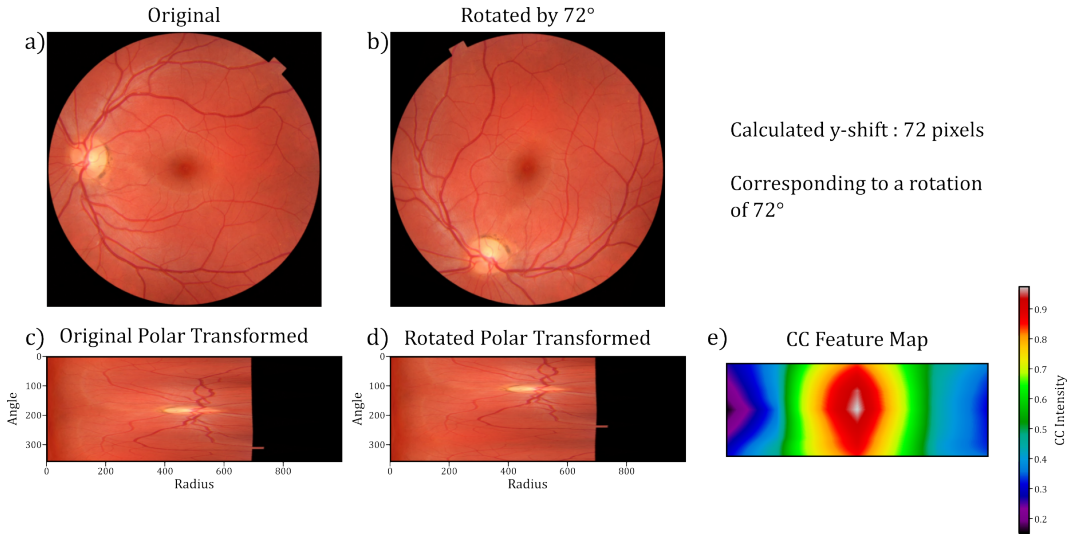


Figure 5.28: Example of automated rotation detection via polar CC. a) shows an original image, which for this example is rotated by 72°, b). Both the original and input image are then converted into the polar coordinate system (c-d), where the y-axis corresponds to the rotation angle, and the x-axis the radius. The CC can then be calculated between the two polar images, the feature map of which is shown in e), where the shift in y corresponds to the rotation different between a) and b).

experiment, the sample alignment is checked to be at least approximately aligned.

This was achieved by first using CC, with a full unit cell as the reference image, to find a clean unit cell on the initial input scan. Following this, it is possible to take advantage of the fact that rotation in Cartesian space is converted into translation along the angular coordinate θ axis in polar space. The overall method is shown in Figure 5.28. Therefore by converting both unit cell images (one used as a reference image and another found in the input image using CC) into polar space images, the translation in the θ axis can be found, again using CC. This method is improved upon for the next part of this project, and is described in more detail in Section 6.5.1.1. This obtained rotation can then be corrected in the Nanonis scanning software after the first scan, resulting in all future images being compared using similarly oriented samples.

5.6.2 Analysis

The tip preparation tool was tested for a total of 20 runs in order to evaluate the average time taken to fully prepare a probe tip from a "bad" to a "good" state. All of these runs were carried out on the Si(111) - 7×7 surface. The results from this evaluation are shown in Table 5.9.

Run Number	Time Taken (s)	Shaping Attempts
1	101.93	1
2	517.98	9
3	674.07	12
4	308.59	5
5	154.09	2
6	152.99	2
7	153.24	2
8	622.06	11
9	826.85	15
10	4893.21	93
11	958.12	12
12	113.11	1
13	153.23	2
14	7875.63	150
15	101.39	1
16	1657.62	31
17	2488.49	47
18	2029.63	38
19	587.69	18
20	675.87	12
Average	1140	23
Average Excluding 10,14	558	12

Table 5.9: Table of results from 20 tip preparation runs used to evaluate the tool. Runs 10 and 14 are seen as outliers as the script was unable make a positive classification due to the surface. Figures given in the main text used the average excluding these runs.

The average time taken to prepare the tip over the full twenty runs was 1140 s (19 minutes) which corresponds to around 23 tip shaping attempts.

Two of the runs, numbers 10 and 14, took significantly longer to achieve a "good" tip state when compared to other runs, and so an additional average was calculated excluding these runs. This resulted in a corrected average time taken of 558 s (9.3 minutes) which corresponds to roughly 12 shaping attempts. The two outlier runs were much slower at achieving a "good" tip due to the state of the sample itself; the distance between steps on this sample was around 50 nm, compared to the usual terrace width on the order of 100s of nm. This meant that tip changes could result in the scan area moving enough to include an area containing a step edge which would cause the image flattening procedure to change the image enough that CCR was not able to make an accurate classification.

5.7 Conclusion

A comparison between common machine learning techniques and more traditional image analysis techniques has been presented in the case of scanning tunnelling microscope tip state classification using atomic or molecular resolution topographical images. We find that using relatively simple image analysis techniques such as cross correlation produces comparable accuracy and true positive precision, and massively reduces the time needed to create a classification scheme, in addition to drastically reducing the amount of data needed to create a functioning classifier when compared to machine learning methods. Using this method we were able to classify datasets that were not possible to classify using machine learning, and also apply our methodologies to publicly available experimental datasets, while obtaining comparable classification precision. This suggests that the CC-based classifier is a robust, easily implemented and widely applicable methodology for atomic resolution studies.

In addition, an automated tip preparation tool was implemented using the CC classifier which is able to successfully obtain and maintain a stable probe tip, highlighting the proof of principle application of this technique in the automation of scanning tunnelling microscopy experiments.

We note that scripts such as this could be included in any automated experiment script which involves periodically scanning the surface, meaning

it is applicable for both obtaining a usable tip to start an experiment and maintaining it throughout random tip change events. An automated script such as this could be implemented alongside autonomous manipulation of individual atoms and molecules on a surface for device fabrication or setting up precise experiments [18, 19].

We have shown that deterministic techniques are able to achieve comparable accuracies and precisions in the case of tip state classification when compared to the frequently used machine learning based classifiers. As noted above there remain instances in which machine learning is a more suitable choice and thus it remains essential to consider the problem at hand when making a choice of computational tool.

References

(1) Barker, D. S.; Blowey, P. J.; Brown, T.; Sweetman, A. *ACS Nano* **2024**, *18*, 2384–2394.

(2) Gross, L.; Mohn, F.; Moll, N.; Liljeroth, P.; Meyer, G. *Science* **2009**, *325*, 1110–1114.

(3) Sweetman, A.; Jarvis, S.; Danza, R.; Moriarty, P. *Beilstein J. Nanotechnol* **2012**, *3*, 25–32.

(4) Gordon, O.; D'Hondt, P.; Knijff, L.; Freeney, S.; Junqueira, F.; Moriarty, P.; Swart, I. *Review of Scientific Instruments* **2019**, *90*, 103704.

(5) Møller, M.; Jarvis, S. P.; Guérinet, L.; Sharp, P.; Woolley, R.; Rahe, P.; Moriarty, P. *Nanotechnology* **2017**, *28*, 075302.

(6) Wang, S.; Zhu, J.; Blackwell, R.; Fischer, F. R. *Journal of Physical Chemistry A* **2021**, *125*, 1384–1390.

(7) Bonnet, N.; Dongmo, S.; Vautrot, P.; Troyon, M. *Microscopy Microanalysis Microstructures* **1994**, *5*, 477–487.

(8) Rashidi, M.; Wolkow, R. A. *ACS Nano* **2018**, *12*, 5185–5189.

(9) Krull, A.; Hirsch, P.; Rother, C.; Schiffrin, A.; Krull, C. *Communications Physics* **2020**, *3*, 1–8.

(10) Barnard, A. S.; Motevalli, B.; Parker, A. J.; Fischer, J. M.; Feigl, C. A.; Opletal, G. *Nanoscale* **2019**, *11*, 19190–19201.

(11) Yothers, M. P.; Browder, A. E.; Bumm, L. A. *Review of Scientific Instruments* **2017**, *88*, 013708.

(12) Bert Voigtländer, *Scanning probe microscopy*; NanoScience and Technology; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007, pp 77–81.

(13) Shi, C.; Velli, M.; Suenaga, R.; Suzuki, M.; Kakio, S.; Abe, M.; Sugimoto, Y.; Custance, O.; Morita, S. *Nanotechnology* **2005**, *16*, 3029–3034.

(14) Lewis, J. *Vis. Interface* **1994**, *95*, 120–123.

(15) Saxton, W. O.; Baumeister, W. *Journal of Microscopy* **1982**, *127*, 127–138.

(16) Koho, S.; Tortarolo, G.; Castello, M.; Deguchi, T.; Diaspro, A.; Vicidomini, G. *Nature Communications 2019 10:1* **2019**, *10*, 1–9.

(17) Alldritt, B.; Urtev, F.; Oinonen, N.; Aapro, M.; Kannala, J.; Liljeroth, P.; Foster, A. S. *Computer Physics Communications* **2022**, *273*, 108258.

(18) Celotta, R. J.; Balakirsky, S. B.; Fein, A. P.; Hess, F. M.; Rutter, G. M.; Stroscio, J. A. *Review of Scientific Instruments* **2014**, *85*.

(19) Kalff, F. E.; Rebergen, M. P.; Fahrenfort, E.; Girovsky, J.; Toskovic, R.; Lado, J. L.; Fernández-Rossier, J.; Otte, A. F. *Nature Nanotechnology* **2016**, *11*, 926–929.

6 Automated Tip State Classification for STS

Similar to the work carried out in Chapter 5, this chapter will focus on work aiming to automate the classification of the probe tip in scanning tunnelling microscopy (STM). However, in this chapter, rather than classifying the state of the probe based on apparent imaging topography, tips will also be classified based on the quality of scanning tunnelling spectroscopy (STS) measurements.

STS is able to extend the capability of STM, allowing for the direct measurement of electronic properties of surfaces and materials with atomic precision. This opens up the ability to map the local density of states (LDOS) of a sample with high spacial resolution [1–3], with different probe terminations resulting in vast range of results [4–7]. The detailed methodology for performing STS measurements is described in Section 2.1.3, where the final $\frac{dI}{dV}$ spectra is obtained. Peaks within a $\frac{dI}{dV}$ spectra correspond increases in conductance at specific bias values, revealing the position of energy levels within the material and allowing us to probe the LDOS.

As for imaging, the sharpness and overall tip shape is crucial in optimising the spacial resolution of STS measurements; sharp tips result in localised tunnelling through a single position, whereas blunt or misshaped tips cause averaging of contributions over larger areas, reducing the spatial resolution and potentially blending the electronic features between different sites.

STS measurements are the result of an integration over the available density of states (DOS) in both the tip and the sample, with the current measured therefore being proportional to the convolution of two. To isolate the DOS within the sample, it is crucial that the tip has a "flat" DOS, which is typically achieved by using a completely metallic tip.

If the tip is terminated with an a non-metallic species, it will have its own electronic structure, which will convolve with the LDOS of the sample, complicating the interpretation of the spectra. Therefore, the state of the probe is critical for obtaining reliable and reproducible STS data, with an ideal probe being one that is relatively sharp (enough to ensure a localised measurement, but not enough to introduce unwanted modulations into the measured DOS) and purely metallic, resulting in all features seen within STS measurements being related directly to the local area of the sample being measured.

Similar to tip preparation for imaging, methods of optimising the probe state, manually for ideal STS, is a slow and laborious problem, involving small indents and bias pulses applied to the tip, checking spectra and imaging after each probe shaping attempt. The automation of this would speed up the process drastically, and result in more consistent and objective spectroscopy measurements being made. One significant attempt to automate this classification using machine learning (ML) was carried out by Wang *et al.* [8], where the group aimed to classify the state of a STM tip based on STS measurements of the Au(111) surface. As will be discussed later, when using a purely metallic probe, a characteristic surface state is observed in the $\frac{dI}{dV}$ curve, appearing as a step at -0.48 V. Using a total of 1789 archived $\frac{dI}{dV}$ spectra, a ML model was trained which aimed to classify new spectra into one of five categories, based on how close to an ideal surface state the spectra appears to be.

The group achieved a final precision in classification of 84%, and a recall of 74%. As is the case for datasets used for image classification problems in scanning probe microscopy (SPM) (such as is described in detail in Chapter 5), the availability of such a large amount of data for training is usually very low, making ML based classifiers troublesome to train. In addition to the lack of data, ML models require careful labelling and a high level of knowledge to be able to train such a model, at which point it is still difficult to understand what the model is learning from the input data. Because of this, it would be preferable to have a model for making such classifications which did not rely on ML, circumventing these drawbacks whilst still making precise classifications for use in automation.

This chapter will detail the classification of STS data, without the need for

ML, therefore reducing the amount of data needed drastically. In conjunction with the feature finding capabilities described in Chapter 5, this would result in a fully automated STS experiment which will be discussed at the end of this chapter.

6.1 Substrate System

To classify the state of the probe for STS experiments, measurements are usually taken over bare areas of a metallic substrate, which show characteristic features within their $\frac{dI}{dV}$ spectra. A common example of this is the surface state seen on coinage metals, which appears as a step function around a specific bias value, shown in Figure 6.1 for the Au(111) surface. For this reason, when choosing a system for this problem one of the coinage metals whose bare surface STS measurements had been well documented in the literature was an obvious choice. With this in mind, the Au(111) surface was chosen, the surface state of which appears as a step function around -0.48 V [9, 10].

In addition to the classification of the probe quality based on the surface state, we will use the cross-correlation (CC) feature finding from Chapter 5 in order to automatically locate various molecules on the surface, and obtain conductance spectra over these. With this, it should be possible to conduct a fully automated experiment, where we are able to obtain a large number of STS measurements over various molecules automatically, this will be discussed in more detail in Section 6.5. For testing, a prototypical system of Tin Phthalocyanine (SnPc) on Au(111) was selected. This system has the advantage that the SnPc adsorbs on the surface in two distinct configurations, one with the tin atom facing up (SnUp), and the other with the tin facing down (SnDown) (Figure 6.2). In each of these two configurations, the STS measurements taken over the centre of the molecule should show differing features, and so distinguishing between the two should be possible in both STM imaging and STS.

The Au(111) substrate was prepared as described in Section 3.1.1, and all imaging for this section was carried out on the low-temperature STM system. A low coverage of SnPc was achieved on the freshly prepared

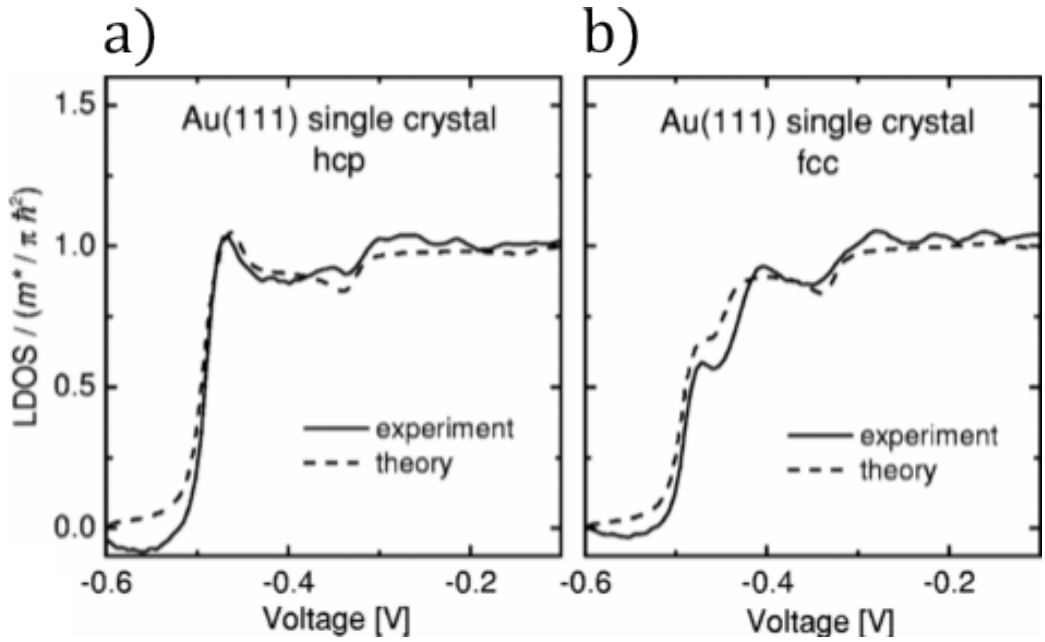


Figure 6.1: STS spectra of the surface state of a clean single Au(111) crystal in the a) hcp and b) fcc regions. Image from [9].

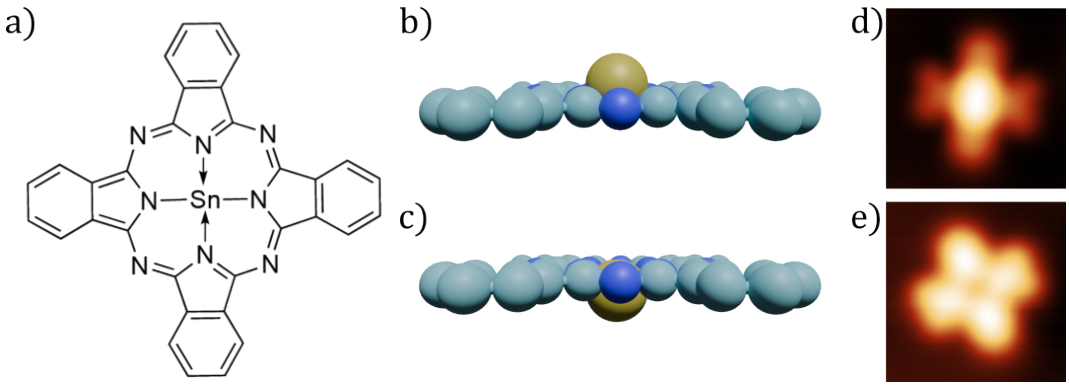


Figure 6.2: a) Structure of tin phthalocyanine (SnPc). Side-on view of SnPc, illustrating its non-planar nature in the b) SnUp and c) SnDown configurations. d) and e) show constant current STM images of the SnUp and SnDown configurations respectively, taken at 5 K, −100 mV, 50 pA.

Au(111) surface by direct thermal sublimation of the molecule onto the sample, with the sample held at 5 K in the scan head. A low coverage of the molecule was chosen specifically to allow for large areas of the bare surface to remain in order to collect STS measurements on the substrate for classification of the tip.

6.2 Dataset Generation

To create the classification models needed for this work, a large amount of data was needed in the form of STS spectra taken with a variety of different tip shapes and configurations. The process of the dataset generation was performed in a manner similar to that described in Section 5.1.1, with a few alterations and additions.

The overall process of the automated dataset generation script is shown in Figure 6.3. One addition to this script compared to the automated data gathering script from Chapter 5 is the addition of I(z) classifications, to ensure a tunnelling junction, the process of which will be discussed more in Section 6.2.1. This is done as a pre-filtering step before obtaining an image. The classification of the state of the probe based on imaging is still needed, as a key feature of a "good" tip for STS is the sharpness of the probe, which gives a high spatial resolution in the acquired data. With a blunt tip, it is not possible to be certain where a spectroscopy measurement is centred, and as such, the measurements are not suitable for high resolution work, even if the tip is purely metallic.

Once the imaging classification is complete, the I(z) is then classified again, ensuring that a tip change did not change throughout the scan. From this point, we can be confident that our tip is sharp, and that the topograph taken using that tip is accurate. The obtained topograph is analysed to find both the largest area of clean metal substrate (the method for this is described in Section 6.2.2) over which the I(V) spectra can be obtained, and to find the location of the molecules in the various configurations, over which, additional I(V) spectra are taken. After this full data gathering step, the tip moves away from the imaging area for a tip preparation step, in order to change the apex substantially before repeating all steps again. Throughout,

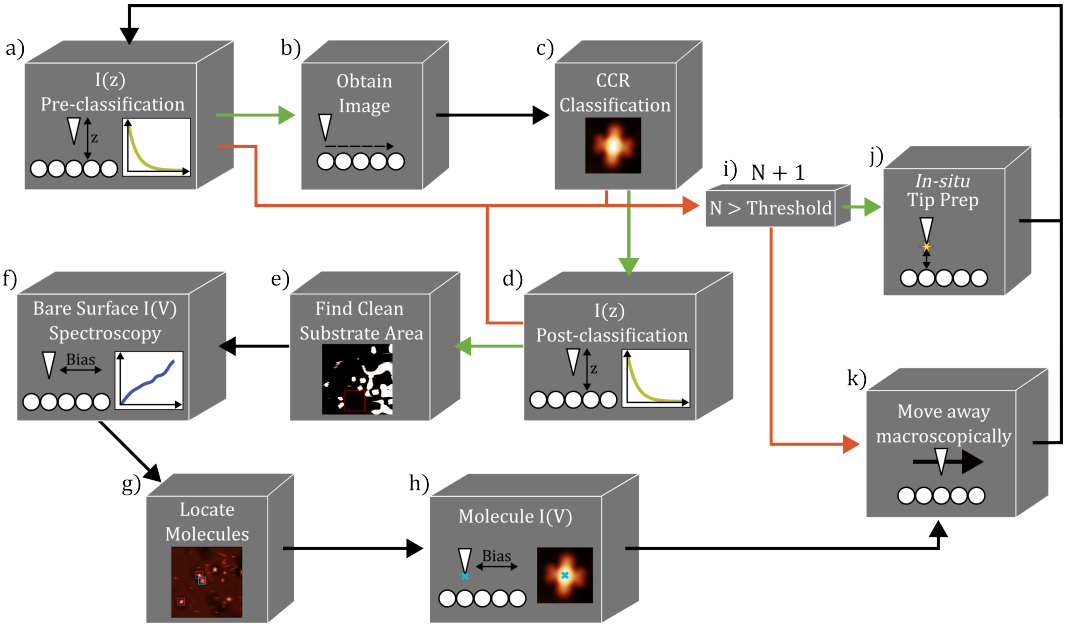


Figure 6.3: Schematic for the automated data gathering script. The script starts by taking an initial I(z) spectra, a), which is classified based on its exponential dependence. If the I(z) is classified as "good", the script then obtains a scan of a specific area, b), followed by a CC based classification, c). If the CC image classification determines the tip to be "good", the script moves onto another I(z) classification, d), followed by an analysis step to find a clean substrate area, e). Using the area found in e), the script obtains 15 I(V) spectra over different positions, f). The script then locates the different configuration of molecules present in the scan, g), before obtaining I(V) spectra over the centre positions of each molecule, h). In the classification steps a), c) and d), if the tip is classified to be "bad", the script will move on to either an *in situ* tip preparation step, j), or if the number of shaping attempts without a "good" tip has exceeded a pre-determined threshold, i), the tip is moved away macroscopically, k), assuming the area is not suitable for classification. Throughout the schematic, green arrows show positive classifications and red arrows indicate negative classifications.

if the tip is classified to be "bad" in either of the I(z) classifications or the CC based imaged classification, the script moves onto a tip preparation event. Similar to the previous data gathering script, if the tip fails in being classified as "good" more than a set number of times in a row, the tip is moved away macroscopically, assuming that the current area of the surface is not suitable for classification; this is usually either due to the area being damaged from tip preparation, or the absence of a SnUp molecule in the frame which is used in the CC classification of the image.

The CC classification is carried out as described in Section 5.2.1.1, with the reference image used being a small cropping of a SnUp molecule as is shown in Figure 6.2d. This was chosen for classification, as in this configuration, the Sn atom in the molecule shows a higher aspect ratio than in the SnDown configuration, and so is more sensitive to the sharpness of the tip. Using this method with a threshold of > 0.98, the model was able to reliably achieve sharp tips for gathering datasets.

6.2.1 I(z) Classification

In addition to the CC based image classification, I(z) spectroscopy measurements are taken and classified before and after the topography scan, in order to filter out the worst tips. Checking if an I(z) for a specific tip shows an exponential trend is a very efficient filtering step, removing bad tips without the need to obtain an entire scan, reducing the time needed for these classifications from ≈ 4 minutes to ≈ 10 seconds. We note that while a tunnelling junction does not mean a tip will image well, or is necessarily metallic, a non-tunnelling junction reliably indicates a contaminated tip.

This exponential dependence of the tunnel current on the tip-sample separation is described in Section 2.1.1.1. The classification of this exponential dependence was performed using the pearsonr function from Scipy in Python. This function evaluates the linear relationship of an input set of data, so by taking the logarithm of the current channel of the I(z) and inputting this into the pearsonr function, a metric for how close to a perfect linear relationship can be calculated. The pearsonr function uses the following equation:

$$r = \frac{\sum(x - m_x)(y - m_y)}{\sqrt{\sum(x - m_x)^2 \sum(y - m_y)^2}} \tag{6.1}$$

Where r is the output measure of the linearity of the input data, x is the distance or z vector, y is the logarithm of the current vector, $log(I)$, and m_x and m_y are the mean values of x and y respectively. The output of this function is a number between 0 and 1, where 1 corresponds to a perfect linear function. Using a threshold of > 0.9995 for a good linear (and therefore exponential) trend, spectra were classified with an overall accuracy of 94% and a 100% precision after being tested on a small set of data containing 36 total spectra.

Unstable or generally bad tips will show a non-exponential trend in their I(z) curves, with either jumps as the tip changes in structure throughout the measurement, or simply a different trend which usually indicated that the current is flowing through a non-vacuum junction, i.e., the tip is in constant mechanical contact with the surface. Since the CC classification needs a full scan for classification, it is much quicker for a single I(z) curve to be taken and analysed rather than a topograph. With this in mind, applying an I(z) classification as a filtering step should increase the efficiency in all automated STM tip classification methods, such as the tip preparation tool used in Section 5.6.

6.2.2 I(V) Generation

The classification of $\frac{dI}{dV}$ spectra on the Au(111) surface requires that spectroscopy measurements are taken over an area free of contaminants or other defects. Under these conditions, measurements should show a characteristic step at a bias of around -0.48 V. Before being able to test the automation of this classification by ML, a large amount of data is needed for model testing. Therefore, this script first analyses a given topograph to locate the largest area of bare substrate, over which the I(V) measurements can be taken, this process is described in Section 6.2.2.1. Once found, 15 individual

I(V) measurements were taken in this area, spread out in a grid to ensure measurements were taken over varied positions on the surface.

The I(V) curves were taken in the range -1.5 V to 1.5 V using the lock-in method, with 1000 points of measurement taken in this range. These parameters were chosen as it is conservative enough to keep the risk of a tip change due to the measurement low, whilst still broad enough to capture a typical bias window for STS of organic molecules on this surface.

6.2.2.1 Locating Clean Areas

Clean areas on the Au(111) surface were located by first binarising the input image around a threshold of $1.5\times$ the mean pixel value. From here, a Python function was written to find the largest square sub-matrix of 1's in the binarised image, representing areas without adsorbates. This sub-matrix is found by creating a size matrix, S, as shown in Figure 6.4b. The first row and column are copied from the input matrix, I, Figure 6.4a, and each cell from here is filled out sequentially (in order of rows then columns) as follows: if the current cell in the input matrix is 1, the corresponding cell in the size matrix is the minimum value of the left, top and top-left cells of the size matrix $+1$, if the current cell in the input matrix is 0, then the corresponding cell in the size matrix is 0. For example, the first value to be filled out is the first row and first column of the size matrix, $S[1,1]$; the corresponding cell in the input matrix, $I[1,1]$, is 1, so we can obtain the value for this cell in the size matrix by taking the minimum of the top ($S[0,1]=1$), left ($S[1,0]=0$) and top-left ($S[0,0]=0$) and adding 1, resulting in a value of 1. The next cell, $S[2,1]$ has the corresponding value $I[2,1]=0$ which results in a 0. By repeating this throughout the whole size matrix, the final matrix shown in Figure 6.4b can be created.

The maximum value in the size matrix shows the size of the largest square of ones in the input matrix, with its index corresponding to the bottom-right corner of the sub-matrix found. For the example shown in Figure 6.4, the largest square sub-matrix is a 4×4 matrix with the bottom right corner at $I[9,6]$. A real example of this is shown in Figure 6.5a, highlighting the largest square sub-matrix found in red.

a)

0	1	0	1	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	1	1	0	1	1	1	1	0	0
1	1	1	0	1	1	1	1	0	1
0	0	1	1	1	1	0	0	1	1
0	0	1	1	1	1	0	0	0	1
0	0	0	1	1	1	1	0	0	1
0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1

b)

0	1	0	1	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	1	1	0	1	1	1	1	0	0
1	1	2	0	1	2	2	2	0	1
0	0	1	1	1	2	0	0	1	1
0	0	1	2	2	2	0	0	0	1
0	0	0	1	2	3	1	0	0	1
0	0	0	1	2	3	2	1	1	1
0	0	0	1	2	3	3	2	2	2
0	0	0	1	2	3	4	3	3	3

Figure 6.4: a) Binarised input matrix of size 10×10. The largest sub-matrix of ones is highlighted in green. b) Corresponding size matrix obtained from a), with the largest sub-matrix size found shown in green.

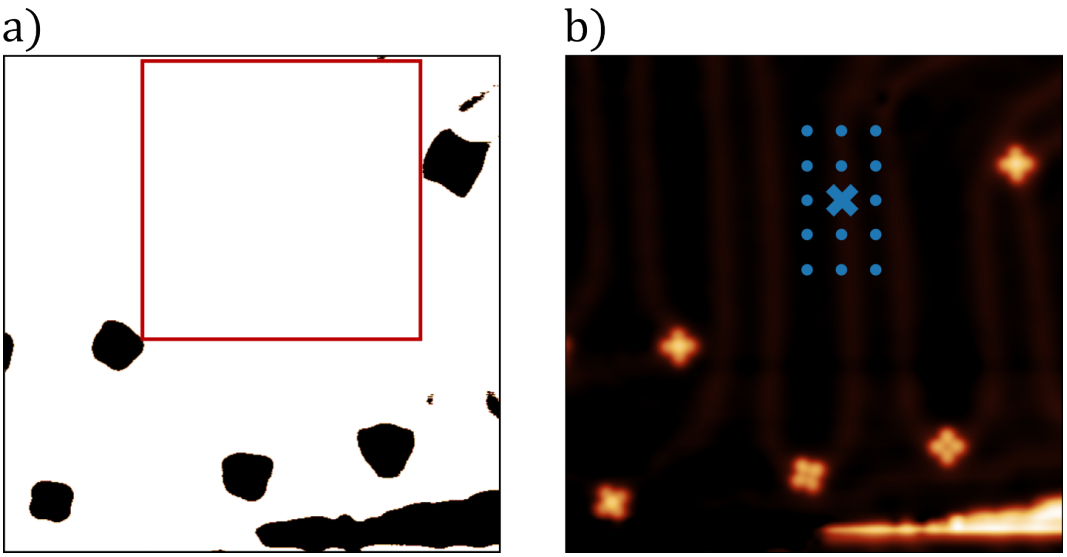


Figure 6.5: a) Binarised image obtained from b), showing a square in red, encircling the largest area of bare substrate. b) Constant height STM scan of SnPc on Au(111) taken at 0.1 V, 20 pA. The blue cross shows the centre of the square shown in a), and the blue circles show the other I(V) locations.

Once the largest square is found, 15 separate I(V) curves are obtained within this area in the positions shown in Figure 6.5b. This results in an automated method of gathering a large amount of STS data over a bare metal surface, at varying positions. This combined with the tip preparation after each round, leads to a varied set of data, obtained using different tip states.

6.2.3 Final Dataset

Using the data generation method described above, we obtained a total of 2604 individual spectra on the bare Au(111) surface, 86 of which were used for our classification models. In order to use this data for training and evaluation of models, the dataset required labelling. We note that labelling of the dataset is non-trivial, as for ML models the model can only attempt to 'learn' to evaluate spectra based on the 'ground truth' provided by the labelling.

Labelling was carried out similarly to in Chapter 5, where a custom Python script was written with a graphical interface. The script would show each spectra individually, with a choice of four labels depending on the visibility of the surface state (SS): SS "good", SS "step visible", SS "peak visible" and SS "not visible". When classifying the data, the specific region around the surface state step was focused on, with the "good" label being attributed to a spectra where the step was clearly visible at the correct position, showing few features before and after the step.

Whilst it was clear which data fell into each category, it is noted that even the data with the most visible surface state contained a background slope, as seen in Figure 6.6c. This slope most likely arises from a macroscopic property of the tip that was not removed by the moderate tip treatment steps used during the automated process, which did not include aggressive treatment such as high current field emission. Our data were therefore classified with this trend in mind, with the final "good" classifications often containing a trend in the < -0.5 V region. The SS "step visible" label was given to spectra whose curves show the step in the correct position, but where a general trend (slope) was also visible throughout the data. The SS "peak visible" label was for spectra where there was an apparent feature

Labels	Count
SS "good"	384
SS Step Visible	482
SS Peak Visible	1169
SS Not Visible	569
Binary "good"	866
Binary "bad"	1738

Table 6.1: Number of spectra in each category.

at the correct position for the step, but not necessarily a step, and the final SS "not visible" label was given to spectra where no feature resembling the surface state could be observed. A representative sampling of spectra from each labelling category are shown in Figures 6.6 and 6.7.

For the final classification, the SS "good" and SS "step visible" categories were combined into a single "good" category, and the SS "not visible" and SS "peak visible" were combined into a "bad" category. This was done to allow for a simple binary classification; further distinction between the classes is unlikely to improve the final model, and would greatly increase the complexity of the problem.

Table 6.1 shows the number of spectra in each category after the labelling step. For ML training, the data were split into training, validation and test sets at a ratio of 70:10:20. This left 1823 spectra for training, 260 for validation and 521 for final testing.

6.3 Classification Methods

6.3.1 Machine Learning Classifier

With the labelling completed, it was possible to train a series 1D convolutional neural networks (CNNs). In total, 72 models were trained, varying the number of convolutional layers between $1-3$, the number of dense training layers between $1-3$, the number of kernels in the first convolutional layer

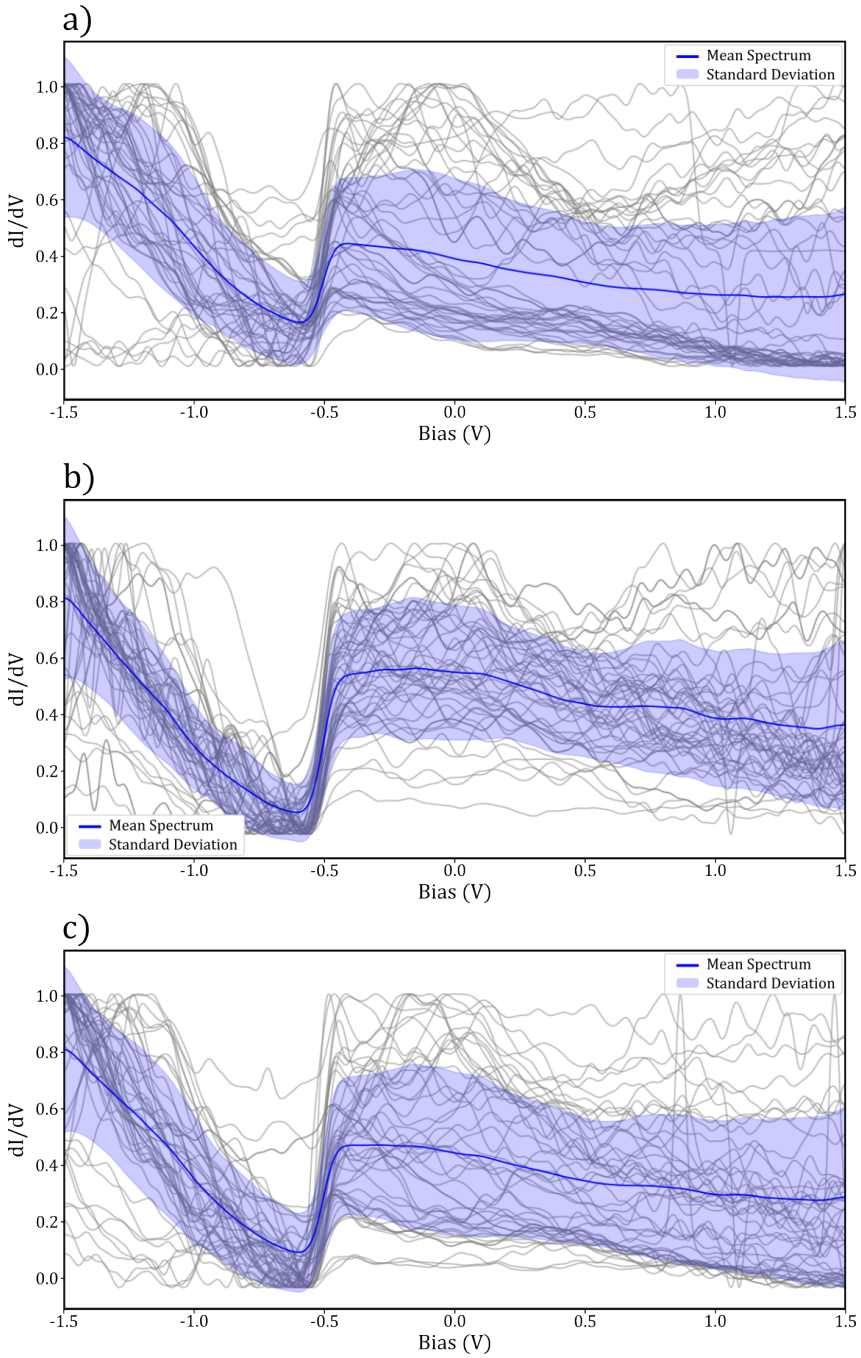


Figure 6.6: Samples of 50 normalised spectra taken over the clean Au(111) surface, mean (blue line) and standard deviation (shaded area) for a) Surface state step visible, b) surface state "good" and c) binary "good" labels.

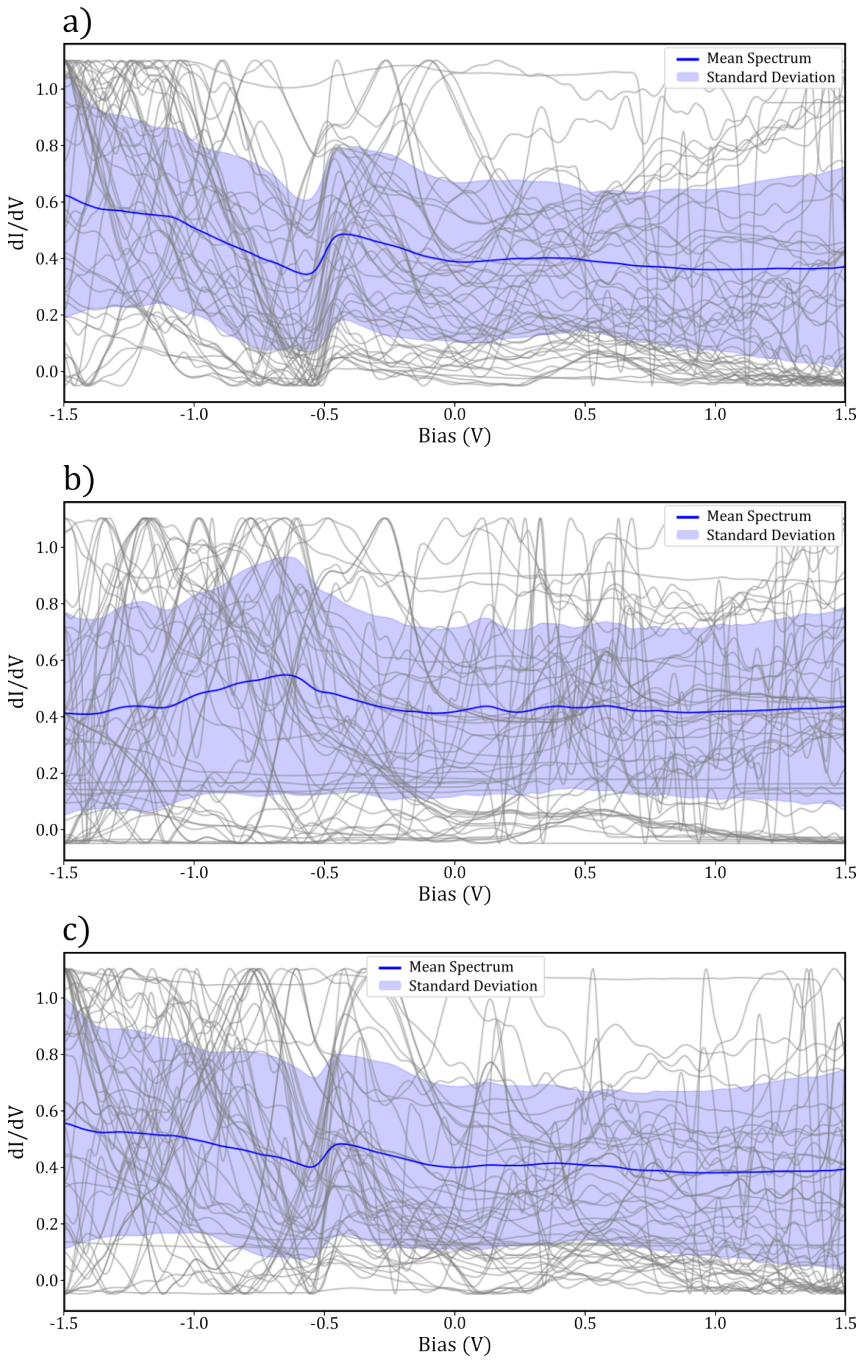


Figure 6.7: Samples of 50 normalised spectra taken over the clean Au(111) surface, mean (blue line) and standard deviation (shaded area) for a) Surface state peak visible, b) surface state not visible and c) binary "bad" labels.

(32 and 64 kernels were used, doubling in successive layers), kernels of sizes 3×3 or 5×5, and dropout layers with rates of either 0.3 or 0.5, including all combinations of these. The training was carried out on the training dataset containing 1823 spectra, validating the model after each epoch on the validation set of 260.

After training, each of these models were evaluated on a test set of 521 spectra, with their final accuracies, precisions and recalls compared. Previously, only the accuracy and precision metrics were considered, however, here it was decided that the recall should also be taken into account. The recall is defined as the percentage of all data labelled as "good", which is then also classified as "good". This metric therefore places more weight on false negatives than the precision metric and is also not as largely skewed by imbalanced datasets as the accuracy metric.

The model which achieved the best balance between the three metrics was one which contained 2 convolutional layers starting with 5×5 kernels, 32 in the first layer and 64 in the second, a single dense training layer, followed by a dropout rate of 0.3. The architecture of this model is shown schematically in Figure 6.8. This achieved an overall accuracy of 86%, a precision of 85% and a recall of 70%.

6.3.2 Deterministic Classifier

For the deterministic classifier, we required a method which would adapt to the entire dataset with a clear set of rules, outputting a metric describing how close any individual spectra is to the expected surface state spectra. To this end, we implemented a simple model to calculate the difference between the surface state step at -0.48 V with a perfect step function, both normalised between $0 - 1$.

In principle, for a "ideal" metallic tip, the spectra would appear completely flat on either side of the step function. However, as noted above, the majority of the data we acquired were not completely flat and showed a noticeable slope even when the SS was clearly visible. Therefore, in order to make a comparison between these tips and the ideal step function, additional processing is needed.

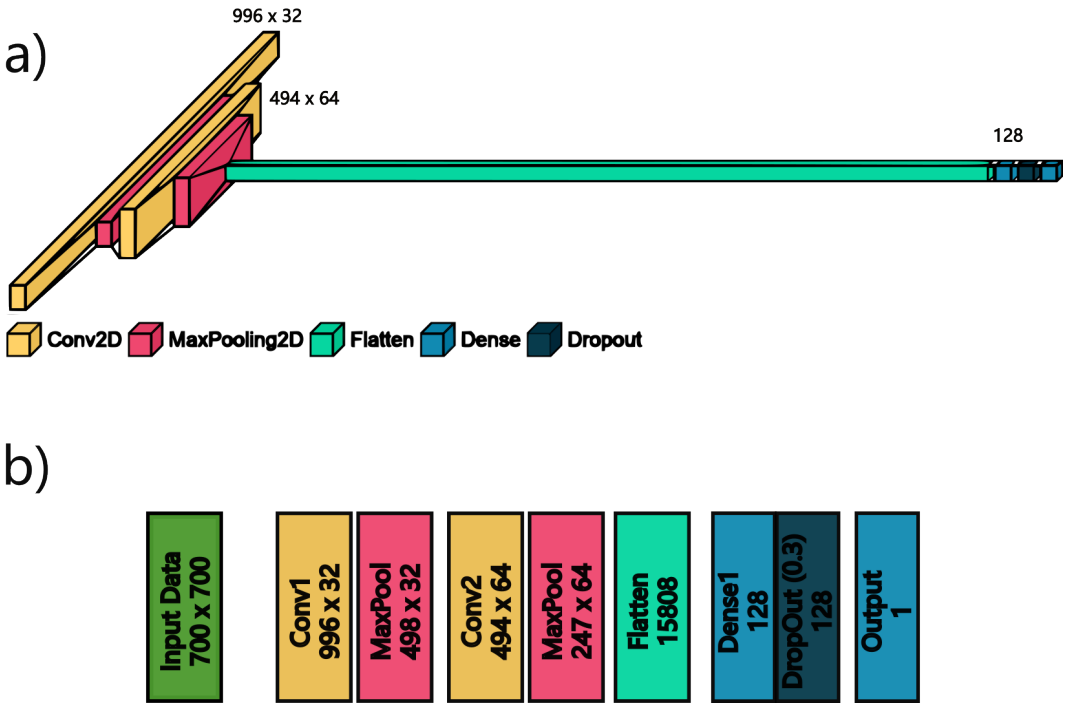


Figure 6.8: Architecture of the CNN model used to classify STS spectra. It consists of a total of 8 layers: 2 consecutive convolutional-pooling layers, a flattening layer, a single training-dropout layer, and ending with a single dense output layer.

Firstly, the spectra are cropped to remove anything outside of the categorisation window, which for this dataset was the bias range -0.55 V $\rightarrow 0.5$ V. From here, any general trend/slope visible in the data needs to be found and subtracted from the step. Commonly in our data, it seems that the trend is a linear offset in the $\frac{dI}{dV}$, and hence a linear function after the step can be fit to the data, and then subtracted from the original spectrum. The specific location of the turning point of the each spectrum is obtained by finding the minimum of the differential of the curve within a small range around -0.48 V, and the step is assumed to be contained within the categorisation range, following this determined turning point. From this, a linear function is fit to the window, an example of which is shown for both a "good" and "bad" classified tip in Figure 6.9a and c respectively.

Once the linear trend is found, it is subtracted from the original spectrum, the result of which is shown in Figure 6.9b and d. For a "good" spectrum, the resultant curve should appear roughly as a step function, and so by direct comparison to a perfect step function, starting at the turning point found earlier, a deterministic classification measure can be made. The specific metric output as the difference between these two curves is the root mean squared (RMS) error between the two, which is described by Equation 6.2

$$RMS = \sqrt{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{6.2}$$

Here, $\hat{y}$ are the perfect step function data points, and y are the spectra for classification. Similar to finding the optimal thresholds for classification in Section 5.2.1.1, a stacked histogram was plotted, showing the spread of the RMS in each category. This histogram is shown in Figure 6.10. From this, the final threshold chosen < 0.25, with all spectra resulting in a value within this range classified as "good".

Using this method, the deterministic model was able to achieve an overall accuracy of 82%, a precision of 86% and a recall of 53% when evaluated on the same test set used for the ML model.

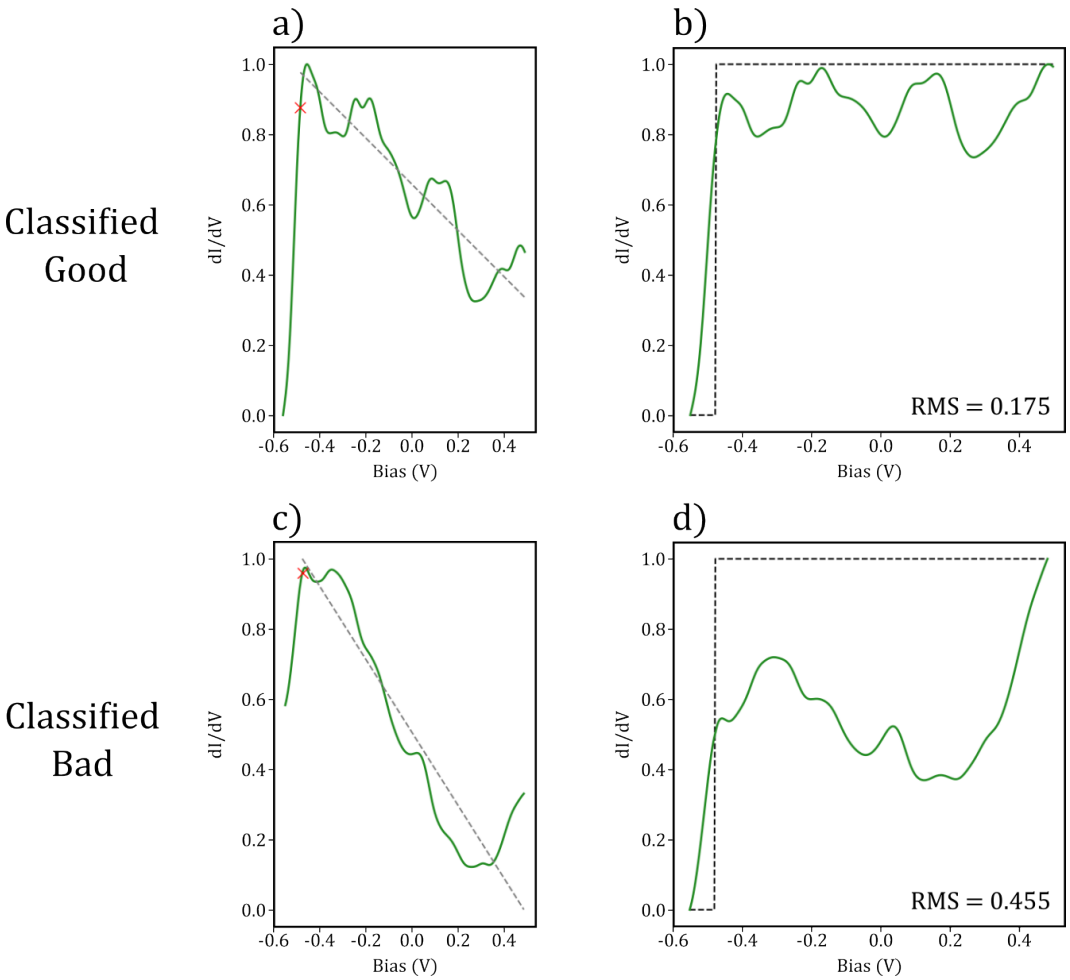


Figure 6.9: a) and b) show the categorisation window on a "good" and "bad" spectra respectively. The red crosses show the automatically found turning point of the step, and the dashed gray lines show the linear fit found past the step. b) and d) show the spectra in a) and b) with their respective linear fits subtracted. The black dashed curves show the ideal surface state step function.

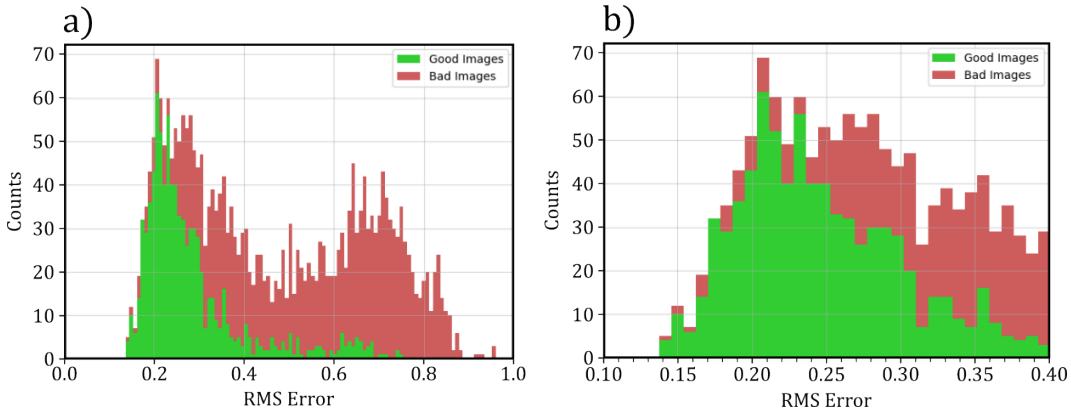


Figure 6.10: Stacked histogram made from labelled spectra, calculating the RMS error between each processed spectrum and an ideal step function. a) shows the full range of RMS, with b) showing the values between 0.1 and 0.4.

6.4 Results and discussion

Both the deterministic and ML based models were tested on the same isolated test set of 521 spectra, with the final results as given in Table 6.2. Both models achieve very similar accuracies and precisions, however the recall for the deterministic model is significantly lower than in the ML model. In practise, this lower recall would mean that more tips which a human may classify as "good" would be misclassified as "bad", slowing down the overall tip preparation process. However, since the precisions of both the ML and deterministic models are very similar, the probability of an automated tip preparation script exiting with a "bad" tip would be roughly the same using either model. As was noted in Section 5.4, as both models show comparable results in the precision of the final classification, the main advantage to using the deterministic model over ML is that the classifier requires much less labelled data for its creation, and hence is easier to apply to a new system.

A similar project to this was undertaken by Wang *et al.* [8], where various ML models were trained to classify the surface state of the Au(111) surface. Here, 9 different ML based models of widely different architectures were trained, as well as a simple correlation based model which compares spectra based on their MSE with respect to a spectrum which was decided to be the

	Deterministic	ML
Accuracy	82%	86%
Precision	86%	85%
Recall	53%	70%

Table 6.2: Table showing the accuracy, precision and recall obtained using deterministic and ML models to classify probe tips based on spectroscopy measurements.

highest quality. The correlation based method achieved very poor results, with a final precision of 41% and a recall of 53% (no accuracies were given for this work). The highest ML based classifier trained here was the CatBoost model, a ready made classifier from the sci-kit package in Python, which is commonly used for classification tasks. Using this model, the classifier was able to achieve a precision of 84% and a recall of 74%. These results are comparable to our ML based model, with only a slightly lower recall in our CNN based classifier differing between the two. It should be noted, however, that Wang *et al.* was attempting to make a classification between five different labels of spectra, whereas our dataset was split into a binary "good" or "bad" before training. In general, binary classifiers are expected to achieve higher accuracies as the differences between the two categories are less subtle.

6.5 Automated Experiment Discussion

In addition to automatically classifying the tip quality via STS on the Au(111) substrate, the script automatically located each SnPc molecule on the Au(111) surface, identified the different configurations of the molecule, and carried out lock-in $\frac{dI}{dV}$ measurements over the centre of each. In this section, we will discuss the STS data taken on the molecules, discuss the impact on the STS data quality due to the quality of the tip and highlight the advantages of automated assessment of tip quality and statistical categorisation of the data in STS.

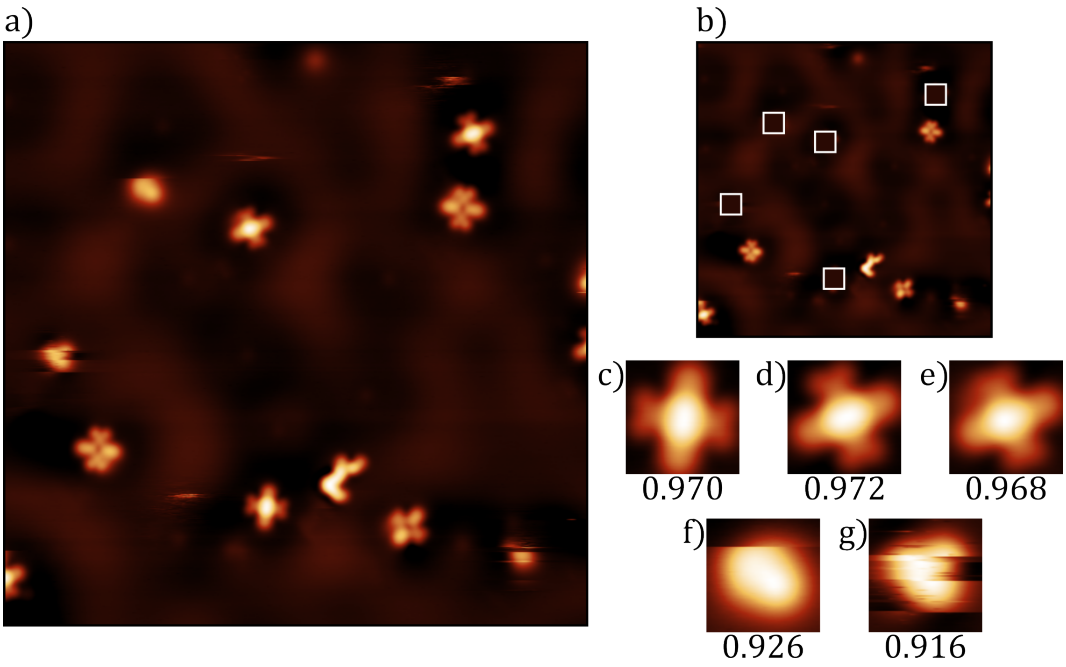


Figure 6.11: a) STM image of SnPc on Ag(111) taken at 0.1 V, 50 pA. b) scan from a) after the top 5 highest correlated positions (indicated by white boxes) are found and removed, shown in c-g) with their respective CCR values.

6.5.1 Molecule Locating

Once a series of Au(111) surface spectra had been obtained for use in the classifier training, the script continued on to obtain STS measurements over the centre of each located SnPc molecule, whilst also distinguishing between the two configurations (SnUp and SnDown) prior to measurement. The identification of each molecular configuration was determined using the CC method with two separate reference images as shown in Figure 6.2d-e. It was found that by using specific thresholds for each reference, the script could distinguish between two molecule configurations within the same image. Figure 6.11 shows an example of the top 5 cross-correlation ratio (CCR) values resulting from a specific input image. Note, the example here from an experiment on the Ag(111) surface, and so the CCR values are slightly different than those used in the final Au(111) experiment shown later in this chapter.

For the final distinctions on the Au(111) surface, the CCR thresholds used for the SnUp and SnDown molecules were 0.983 and 0.980 respectively. Using these thresholds on a small test set of 13 images, the script was able to locate the positions of SnUp molecules with a 100% accuracy and precision, whereas on the SnDown the accuracy achieved was 95% with a precision of 96%.

Once located, lock-in $\frac{dI}{dV}$ curves were obtained over the central atom of each molecule located, using the same -1.5 V to 1.5 V range, with 1000 data points, as were taken over the bare surface.

6.5.1.1 Upgraded Rotation Optimiser

Similar to when locating the C_{60} molecules in Section 5.2.1.1.2, the orientation of the molecule on the surface needs to be ascertained automatically. If the reference image is used to calculate the CCR without an applied rotation, it will not fully represent the similarity of the tip state with that used to obtain the reference image. It was found that the polar translation method described previously was not able to accurately find the rotation in all cases, therefore, the rotation optimiser needed to be improved.

This improved method calculates the CCR of the input image based on a series of rotated reference images, starting with a coarse approach of 6 references each rotated from each other by 60°. The highest correlation of these references is then rotated again by $\pm 30°$ in steps of 10°, followed by a finer step of rotating the maximal rotation by $\pm 5°$ in steps of 1°. This results in a rotation alignment with an angular resolution of 1°.

This method was carried out in addition to the original polar translation method, taking the angle which resulted in the highest CCR between the two used as the optimal rotation.

6.5.2 SnPc Switching Instability

The adsorption of SnPc on coinage metal substrates is well studied, and the molecule is known to undergo an irreversible switch from the SnUp to

the SnDown state on the Ag(111) surface [11] under hole injection. This is usually carried out intentionally by positioning the tip over the centre of an SnUp molecule, and applying a bias pulse via the tip of < -1.9 V. On injection, the Sn atom within the molecule is transiently oxidised to Sn^{3+}, which favours a new position closer to the surface, where the atom binds to the Ag(111), at which point charge transfer from the substrate to the molecule will return it to its neutral state [11].

Whilst carrying out bias spectroscopy over these SnPc molecules, it was found that a switch could occasionally be induced, even if the bias range used does not reach -1.9 V. If the integration time of the spectroscopy measurement was too high, with sufficiently negative bias (e.g., -1.5 V) the switch would commonly occur, and even with parameter adjustments to attempt to account for this, there is still a chance that the switch would be induced. The parameters for our spectroscopy measurements were chosen such that we could reduce the effect of this as much as possible (integration time of 2 ms and a settling time of 200 µs), however, as can be seen in Figure 6.12, switches from the SnUp state to SnDown were still not uncommon.

Due to the automated nature of this experiment, an image would be taken, the molecules located based on this image, and then the spectra would be obtained, saving each file with a name indicating which configuration of molecule the spectra was taken over. Unfortunately, due to this switching occurring over specifically the SnUp molecules, spectra labelled SnUp have the potential to be unreliably labelled. Additionally, it was observed that these switches could occur at the start of the measurement in the initial setting of the bias, or during the bias sweep itself. This results in some spectra showing an obvious switch during the measurement, which appears as a sharp change in the current, and other spectra, where the imaging indicates the molecules have switched configuration, but showing no obvious signs of switching within the spectra itself.

The initial intent of this was to be able to distinguish between the two molecular configurations through observations of the features within spectra over each molecule, in addition to the imaging, however, with the unreliability of the SnUp data, it is not possible to be certain if the SnUp spectra was taken over a molecule in the SnUp or SnDown state, without a significant amount of manual labelling by checking imaging before and after each spectra. For

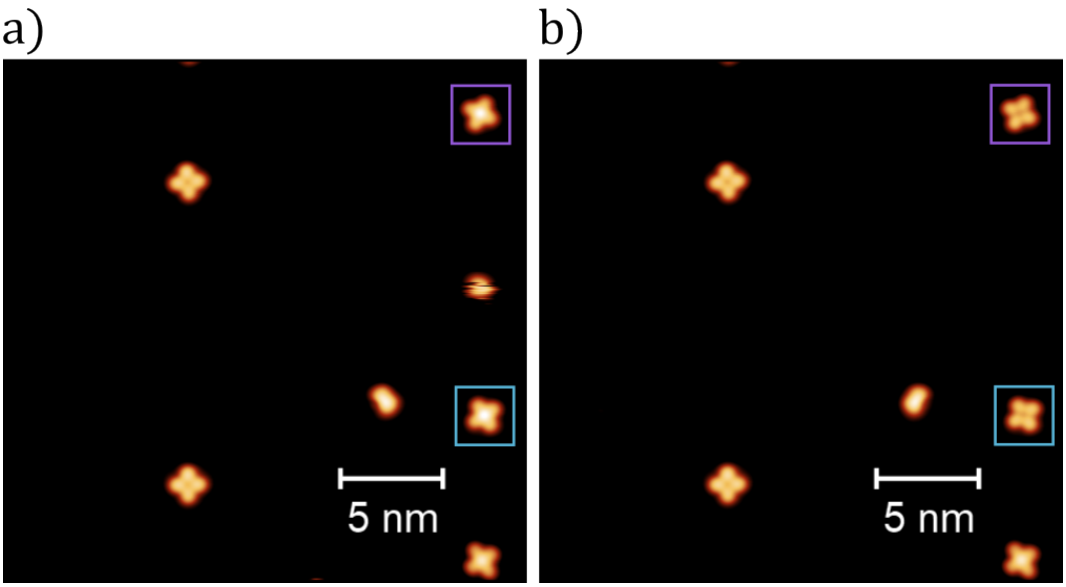


Figure 6.12: Low coverage of SnPc on Au(111) taken at 5 K, 100 mV, 20 pA. a) shows an initial image taken, after which STS measurements between −1.5 and +1.5 V are taken over the molecules. b) shows the same area after some of these measurements have been taken, blue and purple squares show the same molecules which have switched from SnUp to SnDown. Note the SnUp at the lower edge of the image has not switched, despite undergoing the same spectroscopy range as the other two, showing the spurious nature of this switching.

this reason, the results shown in this section will only consider spectra taken over the SnDown configuration.

6.5.3 Results

Using our STS spectra classification method as described in Section 6.3.2, we are able to process the entire dataset collected, and categorise spectra taken using a "good" tip, and with a "bad" tip.

During the data gathering process, before the spectroscopy measurements are taken over the SnPc molecules, the script would first obtain 15 spectra over the bare surface, repeating measurements forward and back through the bias sweep range. During these sweeps, it is possible that the tip would change, meaning that the 30 spectra may not all belong to the same category. For this reason, to classify the state of the tip which was used to carry out STS measurements over the molecules, we need to classify based only the backward curve of the final spectra taken on each iteration.

Throughout the data gathering, a total of 86 images passed the $I(z)$ and CCR pre-classification steps, making it through to obtain molecular STS measurements. Of these 86 tips, 30 had a final STS spectra on Au(111) which were classified as "good" and 56 as "bad". These 30 tips were used to obtain spectra over a total of 49 SnDown molecules. The mean of these curves is shown in Figure 6.13, where it can be clearly seen that there is a peak at roughly 0.8 V which is not present in the bare surface spectra seen in Figure 6.6c.

Previous STS data of SnDown molecules show a clear increase in conductance at both -0.85 V and 0.75 V when imaging at a setpoint of 50 pA [12]. These peaks in conductance correspond to the lowest unoccupied molecular orbital (LUMO) and highest occupied molecular orbital (HOMO) respectively. The cited work, however, was carried out on the Ag(111) surface, as opposed to Au(111) used here, which could explain the slight shift in the position of the HOMO from the literature value of 0.75 V to our consistent measurement of roughly 0.8 V. In addition, the work also suggests a current dependence on the position of the HOMO, which could be a contributing factor to the difference.

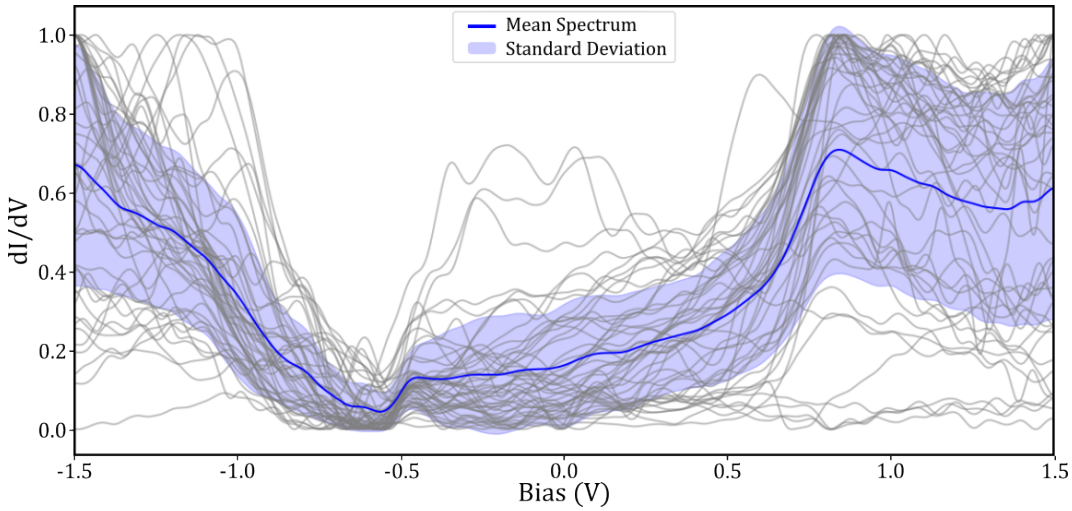


Figure 6.13: Gray curves show 49 normalised STS measurements taken over the centre of SnDown molecules taken with a "good" tip. Blue curve shows the mean and the standard deviation is shown in shaded purple.

When comparing the "good" $\frac{dI}{dV}$ spectra taken over the molecules (Figure 6.13) to the binary "good" bare surface spectra (Figure 6.6c), the surface state and general increase in conductance at biases below -0.5 V can be seen in both. However, unlike the HOMO, which is clearly visible in the molecular $\frac{dI}{dV}$, the LUMO is not visible at the expected bias value, -0.85 V. This is possibly due to the peak being obscured by the shoulder in the negative portion of the spectra. Comparing the region between -1.5 and -0.5 V in both spectra, it can be seen that the mean curve for the binary "good" bare surface spectra appears much more linear than the mean curve molecular spectra. The features contained within this could contain the LUMO, however, without completely deconvolving the tip and sample LDOS, it is difficult to say for certain.

Figure 6.14 shows the mean and standard deviation of the $\frac{dI}{dV}$ spectra taken over SnDown molecules taken with a "bad" tip. By comparing this to Figure 6.13 it is clear that the HOMO peak at 0.8 V is much less prominent. In addition to this, the features throughout the spectra have become less evident. This clear difference between the molecular STS taken with a "good" and "bad" tip, with the former showing expected features, reinforces that

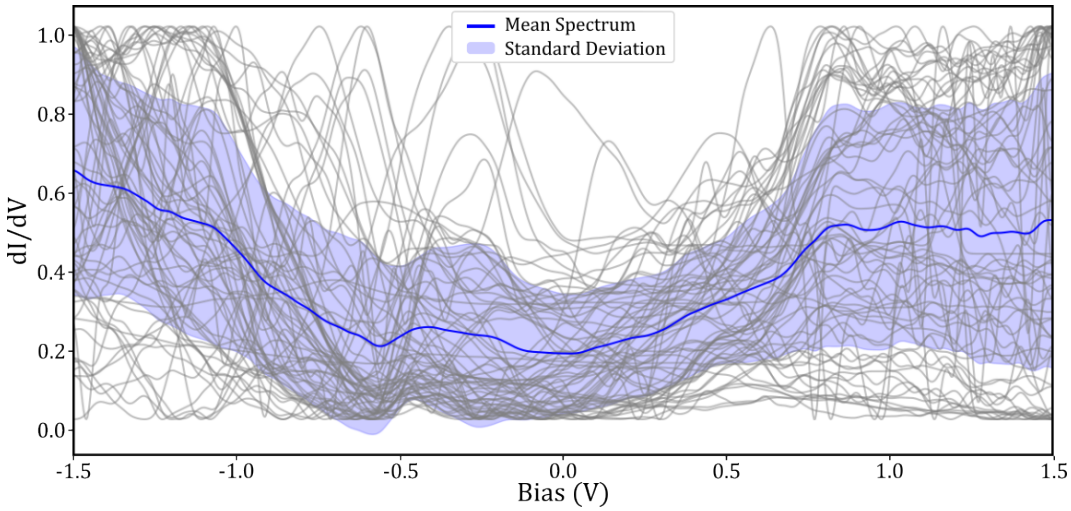


Figure 6.14: Gray curves show 71 normalised STS measurements taken over the centre of SnDown molecules taken with a "bad" tip. Blue curve shows the mean and the standard deviation is shown in shaded purple.

the tip state classification was successful in producing higher quality spectra, and highlights the importance of appropriately charactering the tip state before STS experiments.

Unfortunately, the majority of the spectra taken during this experiment contained a large shoulder between -1.5 and -0.5 V. This was most likely due to a persistent tip configuration, where the DOS in this region was not completely flat. We note that such features may not be changed under the light-to-moderate levels of tip indentation used during the automated data collection in this dataset. In spite of this, the spectra classified to be "good", based on the region around the step, demonstrated clearly that these tips produced superior measurement of the HOMO when compared to tips which were classified as "bad", where the surface state was not observed.

A clear advantage to performing automated experiments with a large number of different tips and over a large number of molecules in different surface positions, is that statistically, variations in the spectra due to the changes in the tip or small changes in the molecular adsorption, will be averaged out, and better approach those from ensemble techniques. As can be seen in both Figures 6.13 and 6.14, there is a substantial variation

in the individual spectra around the mean curve. This is most likely due to variations in the quality of the tip, or slight differences in the molecule itself. However, with a large enough aggregate of different tips, and taking STS measurements over different molecules, when averaged, these small variations should be dominated by the consistent features present in all the data. This can be seen particularly well in Figure 6.13, where some of the individual molecular spectra (grey curves), which here were all taken with tips classified as "good", show a featureless region around the HOMO, whilst others clearly show a strong peak. With individual spectra, it is possible that specific features in the $\frac{dI}{dV}$ could remain unobserved due to spurious problems with the tip.

For a human operator, taking a large number of spectra, with different, yet still "good" tips, on different instances of the same molecule is extremely time consuming. However, with the entire process being automated, this can be carried out very simply, and without any need for constant monitoring.

With only minor editing, the script used to generate STS data for this chapter could be used to carry out such an experiment. Currently, the script classifies the state of the probe first based on an $I(z)$ measurement, and then based on imaging, after which it generates a number of $\frac{dI}{dV}$ spectra over the bare area, before taking STS measurements over different configurations of molecules present on the surface. Using the proposed method of classifying the tip based on a $\frac{dI}{dV}$ spectrum taken over the bare surface, the script could take molecular spectra only using probes which have been classified as "good" based on the surface state spectra. An example flow diagram, with data taken from a generation run where the surface state spectrum was classified as "good" is shown in Figure 6.15.

6.5.4 Conclusion and Outlook

We have shown that it is possible to perform a fully automated experiment, carrying out STS measurements over targeted areas of specific organic molecules, including the ability to modify and characterise the state of the tip, by both analysis of its spectroscopic characteristics, and imaging quality without the use of machine learning. This enables us the ability to obtain

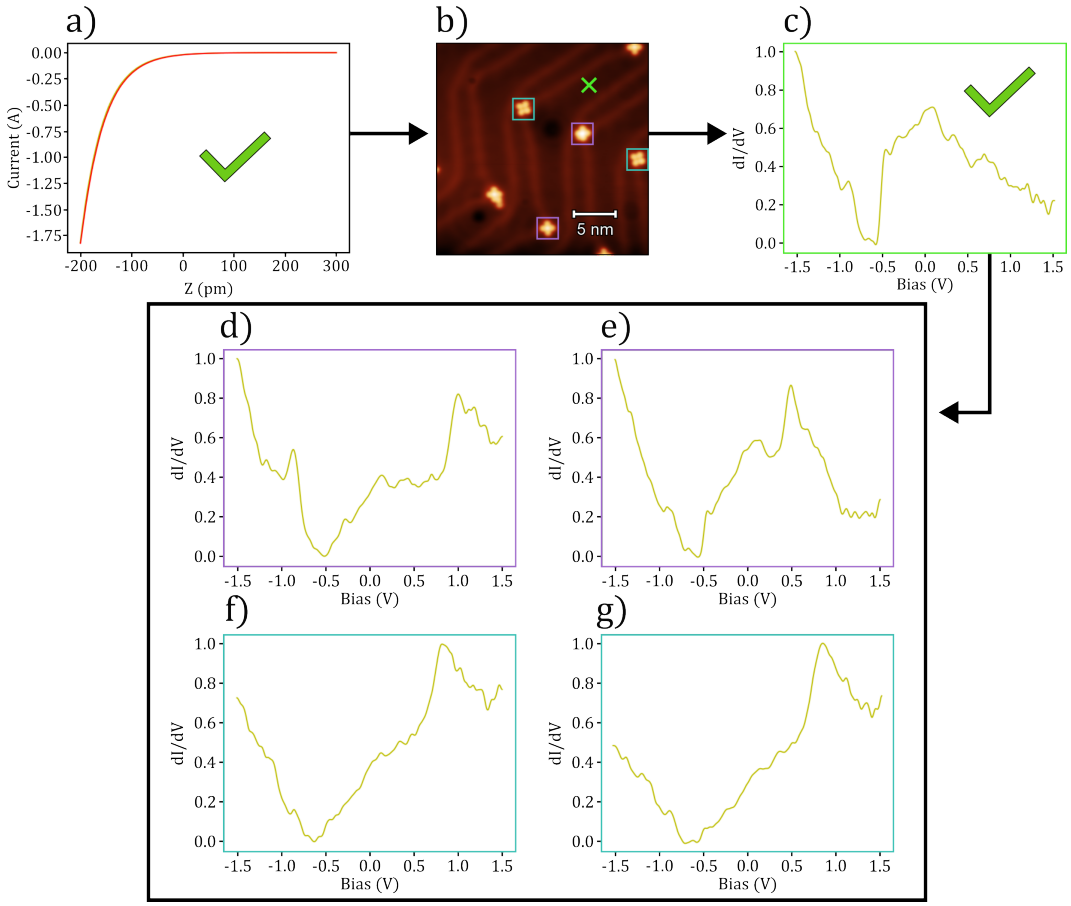


Figure 6.15: Example flow of an automated spectroscopy experiment taken over various SnPc molecules on the Au(111) surface. a) Initial $I(z)$ measurement is taken, where an exponential dependence is observed and so moves onto imaging, b). The tip is then classified to be "good" based on imaging, and so a clean area of the substrate is located (marked by a green cross), where a surface STS measurement is taken, c). This is then classified to be "good", at which point the various orientations of SnPc are located (SnUp in pink boxes and SnDown in blue boxes), where STS measurements are taken as shown in d)-g). d)-e) correspond to measurements taken over SnUp molecules, while f)-g) correspond to SnDown. The script would then change the tip and repeat the steps, over different areas, varying the tip after each set of STS measurements (formed through *in situ* tip preparation).

a large number of spectra over various features on a surface, without the need for an operator to be present, and to perform statistical analysis of the spectroscopic data, via the automated labelling of the state, and location of the spectrum. In addition, the ability to carry this out without machine learning means that this method could be easily applied to other systems with little effort.

Unfortunately, due to the manipulation of the SnUp molecules, the differentiation between different molecules via analysis of the STS spectra was not possible. The script was able to identify the different configurations accurately, and so clearly shows a viable method for automated experimentation. Further experiments using this script could be carried out using different molecule-substrate combinations. The script only requires that there is a characteristic state present in the system which can be observed in STS and modelled with some accuracy, such as the step function used here in order to assess the tip quality *in situ*.

Importantly, the ability to carry this out without machine learning means that this method can be easily adapted to different adsorbate/substrate systems without the need for extensive data collection to train ML models. This methodology can aid in the rapid characterisation of new materials via automated probing of different features in a system, taking numerous measurements over different areas, only requiring an operator once the experiment is complete, to process the resultant data for analysis.

References

(1) Rizzo, D. J.; Veber, G.; Cao, T.; Bronner, C.; Chen, T.; Zhao, F.; Rodriguez, H.; Louie, S. G.; Crommie, M. F.; Fischer, F. R. *Nature 2018 560:7717* **2018**, *560*, 204–208.

(2) Gross, L.; Moll, N.; Mohn, F.; Curioni, A.; Meyer, G.; Hanke, F.; Persson, M. *PRL* **2011**, *107*, 86101.

(3) Repp, J.; Meyer, G.; Stojković, S. M.; Gourdon, A.; Joachim, C. *Physical Review Letters* **2005**, *94*, 026803.

(4) Repp, J.; Meyer, G.; Paavilainen, S.; Olsson, F. E.; Persson, M. *Science* **2006**, *312*, 1196–1199.

(5) Bartels, L.; Meyer, G.; Rieder, K. H. *Applied Physics Letters* **1997**, *71*, 213–215.

(6) Lagoute, J.; Kanisawa, K.; Fölsch, S. *Physical Review B* **2004**, *70*, 245415.

(7) Liljeroth, P.; Repp, J.; Meyer, G. *Science (New York, N.Y.)* **2007**, *317*, 1203–1206.

(8) Wang, S.; Zhu, J.; Blackwell, R.; Fischer, F. R. *Journal of Physical Chemistry A* **2021**, *125*, 1384–1390.

(9) Andreev, T.; Barke, I.; Hövel, H. *Physical Review B - Condensed Matter and Materials Physics* **2004**, *70*.

(10) Thomas, J. C. et al. *npj Computational Materials 2022 8:1* **2022**, *8*, 1–7.

(11) Yongfeng, W.; Kröger, J.; Berndt, R.; Hofer, W. A. *Journal of the American Chemical Society* **2009**, *131*, 3639–3643.

(12) Toader, M.; Hietschold, M. *Journal of Physical Chemistry C* **2011**, *115*, 3099–3105.

7 Conclusion

Throughout this thesis, various methods of automating techniques used in scanning probe microscopy (SPM) have been explored, with a specific focus on comparing the use of machine learning (ML) with deterministic methods for classification and substrate analysis. Deterministic methods, being much less data-intensive, provide significant advantages over ML, especially when large labelled datasets are unavailable.

We began by introducing the core SPM and ML concepts in Chapter 2, followed by a description of the experimental and computational techniques used to obtain and analyse the data used throughout this thesis in Chapter 3. In Chapter 4, a scale-invariant method for mapping out the atomic positions, including those obscured by defects/adsorbates, was presented. This method utilises the fast Fourier transform (FFT), applied to input topographical scanning tunneling microscopy (STM) images, predicting the state of the surface shown in the input, without the defects or adsorbates - essentially recreating the same imaging area in a perfectly clean configuration. This is carried out by isolating high intensity peaks in Fourier space, which correspond to frequencies which make up the repeating structure of the surface, filtering out any non-reciprocal features in the input. A peak finding function then identifies the location of adatoms in both the input, and filtered images, comparing the two to identify both the clean adatoms, and those which are obscured.

Chapter 5 introduced alternative image classification methods for determination of the state of an STM probe, diverging from ML-based methods, which at present are predominantly used. The deterministic methods discussed depend on the nature of the system in question, utilising a comparison of an input image with the reference image, showing the expected appearance of a specific feature (when scanned using an ideal tip), using a method known

as cross-correlation (CC). This approach was trialed on three substrate systems: Si(111) - 7×7, B:Si and Cu(111) with a low coverage of C_{60} molecules. For the silicon based systems, a ML based convolutional neural network (CNN) classifier was also trained for a direct comparison. The CC-based classifier achieved results comparable to the CNN, demonstrating that ML is not always necessary for image classification. However, on an additional Cu(111) dataset, with a low coverage of Cu adatoms and CO molecules, the method was not able to clearly distinguish between a "good" and a "bad" tip due to the simplicity of the reference feature. This resulted in an additional circularity measure being implemented to improve classification. The use of these deterministic classification methods allow for the problem of maintaining a probe tip in STM to be solved, without the need for large, labelled datasets, and their effectiveness was demonstrated in an automated tip preparation tool, significantly optimising the process by eliminating the need for an operator to be present.

In Chapter 6, we explored a similar tip state classification technique applied to scanning tunnelling spectroscopy (STS) measurements. Typically, the quality of a probe for STS is characterised by having a flat density of states (DOS), indicative of a purely metallic structure on the tip. This is often assessed by taking STS measurements over a position with a characteristic appearance in the resulting $\frac{dI}{dV}$ curve. We focus here on the surface state observed on Au(111), which appears as a step function at a bias of -0.48 V, and is featureless otherwise. A large amount of spectra were obtained over the bare surface, and the surface state visibility was manually labelled for use in training a ML classifier for comparison. A deterministic method was used to accurately classify the state of the probe by comparing each spectrum to an ideal step function around the expected bias. Once again, it was found that the deterministic classifier can determine the quality of the probe with comparable results to the ML method, affirming that simpler deterministic approaches can often be preferable.

Whilst ML is undeniably a powerful tool for classification problems, it is commonly used without careful consideration of whether it is the optimal solution, particularly when taking into account the required data collection and labelling of data. There are, of course, instances where the deterministic methods described in this thesis are not suitable, but it is crucial to consider

simpler alternatives before defaulting to a technique with known drawbacks. As with many scientific advancements, plus ça change, plus c'est la même chose (the more things change, the more they stay the same). Simpler, time-tested methods continue to carry worth against more complex techniques, especially in situations such as in labor intensive experimental physics, where minimising the requirement for large datasets is paramount.