MAX PLANCK INSTITUTE
FOR POLYMER RESEARCH

# Automated Correlative Light and Electron Microscopy (CLEM) Using Deep Learning

## Dissertation

Zur Erlangung des Grades

“ Doktor der Naturwissenschaften (Dr. rer. nat.)“

im Promotionsfach Chemie

am Fachbereich Chemie, Pharmazie, Geographie und Geowissenschaften (FB09)

der Johannes Gutenberg-Universität Mainz

Vorgelegt von

**Daksh**

Geboren in Sambhal, Indien

Mainz, 18.11.2025

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

SFB 1066

Dekanin:

Betreuerin:
ggf. Zweitbetreuer:

Tag der mündlichen Prüfung:

This thesis was carried out from October 2021 until August 2025 in the Department of Physical Chemistry of Polymers, led by Prof. Dr. Katharina Landfester, in the Electron Microscopy group of Dr. Ingo Lieberwirth at the Max-Planck-Institute for Polymer Research, Mainz.

## Declaration of independence / Eigenständigkeitserklärung

I hereby declare, that I have written the thesis entitled: **Automated Correlative Light and Electron Microscopy Using Deep Learning** independently and have not used any sources or aids (including AI-based applications or tools*) other than those specified. All verbatim or analogous citations and quotations are labelled and referenced (this also applies to texts or words searched by using available AI tools and search engines, i.e. Google or ChatGPT). I confirm that I have not used any aids whose use has been explicitly excluded by the examiner.

*I have documented the AI tools used in the Appendix III "Use of AI tools & software".

By submitting this work, I take over responsibility for the entire document submitted. I am therefore also responsible for any AI-generated content that I have included in my work. I have checked the accuracy of the (AI-generated) statements and content to the best of my knowledge and belief. In carrying out this research, I complied with the rules of standard scientific practice as formulated in the statutes of the Johannes Gutenberg University Mainz to ensure standard scientific practice.

_____

Mainz, 18.11.2025

# Acknowledgements

*“I have no special talents. I am only passionately curious.” - Albert Einstein*

Daksh

# List of Abbreviations

| | |
|---|---|
| AI | Artificial intelligence |
| IR | Image registration |
| CLEM | Correlative light & electron microscopy |
| IP | Image pair |
| MSE | Mean squared error |
| EM | Electron microscopy |
| SEM | Scanning electron microscopy |
| TEM | Transmission electron microscopy |
| LM | Light Microscopy |
| FM | Fluorescence microscopy |
| LR | Low resolution |
| HR | High resolution |
| FoV | Field of view |
| DNN | Deep neural network |
| ANN | Artificial neural network |
| CNN | Convolutional neural network |
| MI | Mutual information |
| MINE | Mutual information neural estimator |
| TM | Template matching |
| MS-SSIM | Multiscale structural similarity index measure |
| GPU | Graphics processing unit |
| CPU | Central processing unit |
| RAM | Random access memory |
| GAN | Generative adversarial network |
| SRGAN | Super-resolution generative adversarial network |
| GPT | Generative pre-trained transformers |
| VAE | Variational autoencoder |
| TFUDL | Training free unsupervised deep learning |
| ML | Machine learning |
| DL | Deep learning |

| | |
|---|---|
| FE | Feature extraction |
| TLC | Transform landmark correspondence |
| UI | User interface |

# List of Tables

# List of Figures

# Table of Contents

# Abstract

Correlative light and electron microscopy (CLEM) has emerged as a pivotal technique in the detailed structural analysis and precise identification of biological specimens by integrating fluorescence labelling with electron microscopy. This combined modality harnesses the advantages of both imaging techniques, allowing for enhanced visualization e.g. biological samples at multiple scales and resolutions. The advent of deep learning (DL) approaches has fundamentally transformed microscopy imaging within the biological sciences. By leveraging DL algorithms, it is now possible to extract intricate and meaningful features from complex imaging data, thereby enabling sophisticated analytical tasks that were previously unattainable. These tasks encompass a broad spectrum of challenges, including, but not limited to, image segmentation, classification, object detection, and resolution enhancement. The application of DL methodologies to these problems has demonstrated substantial improvements in accuracy and efficiency compared to conventional image processing techniques[1,2].

Within the scope of CLEM, one of the foremost challenges is the accurate registration of images acquired from disparate modalities. Image registration (IR) defined as the process of geometrically aligning two or more images of the same object or source, is essential for correlating and integrating complementary spatial information derived from different imaging systems. The inherent differences in image characteristics such as contrast, scale, and spatial resolution between light microscopy and electron microscopy necessitate sophisticated multi-dimensional and multi-modal registration techniques. Effective registration not only facilitates meaningful data fusion but also enables comprehensive characterization of biological and any other visual entities, including cells, tissues, and subcellular structures such as proteins. A variety of image registration methods have been developed to address this challenge[3,4,5]. Most existing techniques rely heavily on supervised learning frameworks, which, despite delivering high registration accuracy, require extensive manual intervention for the annotation or identification of key reference points or landmarks. This dependency on human input inherently limits the scalability and automation potential of these methods, presenting a bottleneck for high-throughput CLEM applications.

In this context, I propose a novel image registration strategy, specifically tailored for cross-modality datasets characterized by heterogeneous spatial resolutions. The precise localization of landmarks across modalities remains a particularly demanding problem due to the variations in imaging physics and sample representation. While prior efforts have explored model-based landmark prediction and manual annotation combined with supervised learning paradigms, these approaches are constrained by their reliance on annotated datasets and labour-intensive pre-processing[6,7]. My approach advances the field by introducing an automated, training-free, unsupervised neural network algorithm capable of performing robust image registration for CLEM data. This algorithm autonomously identifies and localizes landmark positions with high precision, eliminating the need for manual landmark selection or extensive training data. Additionally, the method is scalable, accommodating very large image dimensions up to 11,000 x 11,000 pixels and dependent only on the available GPU memory capacity. This capacity to handle ultra-high-resolution images ensures its applicability to cutting-edge microscopy datasets and facilitates the automation of the multimodal image registration pipeline, thereby enhancing throughput and reproducibility.

In summary, the proposed approach represents a significant advancement in the automated processing of correlative microscopy data, addressing critical challenges in multi-modal image registration through an innovative unsupervised learning approach. This work not only contributes to the methodological toolkit available for biological image analysis but also holds promise for accelerating discoveries by enabling more seamless integration of multimodal imaging data.

To utilize these AI-based methods effectively, it is essential to perform an adequate pre-processing of the image files to ensure reliable results. In this work, I present a workflow designed to address the challenges associated with image pre-processing for the registration of multimodal image data. The primary focus of my study is the precise overlay of images obtained from different microscopy modalities. This involves correlating electron microscopy images with light microscopy images, which differ not only in visual representation but also in pixel resolution. The objective of the proposed pre-processing is to generate image pairs that exhibit exactly the same image details. To achieve this, I have incorporated a template matching method based on the normalized correlation coefficient. Additionally, to harmonize image dimensions and pixel sizes, I leverage the fact that both images depict the same underlying structures, providing an internal standard for calibration.

This calibration of pixel size is a crucial step for successful template matching and the subsequent registration of image pairs; however, it still requires some degree of manual supervision.

Currently, there are no computational tools or software solutions available that can automatically align pixel sizes between images acquired from different sources. Once pre-processed, the image pairs are ready for use as inputs in image analysis, registration, and segmentation workflows, with potential applications extending beyond these areas. My weakly supervised image pre-processing pipeline holds promise for integration into automated microscopy platforms, enabling unattended operation and facilitating high-throughput imaging analysis of multimodal datasets.

# Zusammenfassung

Die korrelative Licht- und Elektronenmikroskopie (CLEM) hat sich durch die Integration von Fluoreszenzmarkierung und Elektronenmikroskopie zu einer zentralen Technik für die detaillierte Strukturanalyse und präzise Identifizierung insbesondere biologischer Proben entwickelt. Diese kombinierte Methode nutzt die Vorteile beider Bildgebungsverfahren und ermöglicht eine verbesserte Visualisierung biologischer Proben unter verschiedenen Vergrößerungen. Die Entwicklung von Deep-Learning-Ansätzen (DL) hat die Bildanalyse grundlegend verändert. Durch den Einsatz von DL-Algorithmen ist es nun möglich, komplexe und aussagekräftige Merkmale aus komplexen Bilddaten zu extrahieren und damit anspruchsvolle Analyseaufgaben zu lösen, die zuvor nicht möglich waren. Diese Aufgaben umfassen ein breites Spektrum von Herausforderungen, darunter Bildsegmentierung, Klassifizierung, Objekterkennung und Auflösungsverbesserung. Die Anwendung von DL-Methoden auf diese Probleme hat im Vergleich zu herkömmlichen Bildverarbeitungstechniken zu erheblichen Verbesserungen in Bezug auf Genauigkeit und Effizienz geführt[1,2].

Eine der größten Herausforderungen bei CLEM ist die genaue Registrierung von Bildern, die mit unterschiedlichen Modalitäten aufgenommen wurden. Die Bildregistrierung (IR), definiert als der Prozess der geometrischen Ausrichtung von zwei oder mehr Bildern desselben Objekts oder derselben Quelle, ist für die Korrelation und Integration komplementärer Informationen aus verschiedenen Bildgebungssystemen von entscheidender Bedeutung. Die inhärenten Unterschiede in den Bildmerkmalen wie Kontrast, Maßstab und räumliche Auflösung zwischen Lichtmikroskopie und Elektronenmikroskopie erfordern ausgefeilte, mehrdimensionale und multimodale Registrierungstechniken. Eine effektive Registrierung erleichtert nicht nur eine sinnvolle Datenfusion, sondern ermöglicht auch eine strukturelle Identifikation von z.B. Zellen, Gewebe oder subzelluläre Strukturen wie Mitochondrien oder andere Zellstrukturen. Um dieser Herausforderung zu begegnen, wurde eine Vielzahl von Bildregistrierungsmethoden entwickelt[3,4,5]. Die meisten bestehenden Techniken stützen sich stark auf überwachte Lernframeworks, die zwar eine hohe Registrierungsgenauigkeit liefern, jedoch umfangreiche manuelle Eingriffe für die Annotation oder Identifizierung wichtiger Referenzpunkte oder Landmarken erfordern. Diese Abhängigkeit von menschlichen

Eingaben schränkt die Skalierbarkeit und das Automatisierungspotenzial dieser Methoden von Natur aus ein und stellt einen Engpass für CLEM-Anwendungen mit hohem Durchsatz dar.

In diesem Zusammenhang schlage ich eine neuartige Bildregistrierungsstrategie vor, die speziell auf multimodale Datensätze mit heterogenen räumlichen Auflösungen zugeschnitten ist. Die präzise Lokalisierung von Landmarken über verschiedene Modalitäten hinweg bleibt aufgrund der Unterschiede in der Bildgebungsphysik und der Probenrepräsentation ein besonders anspruchsvolles Problem. Während frühere Bemühungen sich mit modellbasierter Landmarkenvorhersage und manueller Annotation in Kombination mit überwachten Lernparadigmen befassten, sind diese Ansätze durch ihre Abhängigkeit von annotierten Datensätzen und arbeitsintensiver Vorverarbeitung eingeschränkt[6,7]. Mein Ansatz bringt das Feld voran, indem er einen automatisierten, trainingsfreien, unüberwachten neuronalen Netzwerkalgorithmus einführt, der in der Lage ist, eine robuste Bildregistrierung für CLEM-Daten durchzuführen. Dieser Algorithmus identifiziert und lokalisiert Landmarkenpositionen autonom und mit hoher Präzision, wodurch die manuelle Auswahl von Landmarken oder umfangreiche Trainingsdaten überflüssig werden. Darüber hinaus ist die Methode skalierbar, sodass sie sehr große Bildabmessungen von bis zu 11.000 x 11.000 Pixeln verarbeiten kann und nur von der verfügbaren GPU-Speicherkapazität abhängig ist. Diese Fähigkeit, Bilder mit ultrahoher Auflösung zu verarbeiten, gewährleistet die Anwendbarkeit auf modernste Mikroskopiedatensätze und erleichtert die Automatisierung der multimodalen Bildregistrierungs-Pipeline, wodurch der Datendurchsatz und die Reproduzierbarkeit verbessert wird.

Zusammenfassend stellt der in dieser Arbeit vorgestellte Ansatz einen bedeutenden Fortschritt in der automatisierten Verarbeitung korrelativer Mikroskopiedaten dar und bewältigt kritische Herausforderungen bei der multimodalen Bildregistrierung durch einen innovativen Ansatz basierend auf unüberwachten Lernen. Diese Arbeit trägt nicht nur zum methodischen Werkzeugkasten für die biologische Bildanalyse bei, sondern verspricht auch eine Beschleunigung von Entdeckungen, indem sie eine nahtlosere Integration multimodaler Bilddaten ermöglicht.

Die Entwicklung KI-basierter Methoden zur Bildanalyse hat in letzter Zeit erhebliche Fortschritte gemacht und ermöglicht eine effiziente und zeitsparende Extraktion von

Informationen aus großen Bilddatensätzen. Um diese KI-basierten Methoden effektiv nutzen zu können, ist es jedoch unerlässlich, eine angemessene Vorbereitung der Bilddateien durchzuführen, um zuverlässige Ergebnisse zu gewährleisten. In dieser Arbeit stelle ich zusätzlich einen Arbeitsablauf vor, der entwickelt wurde, um die Herausforderungen im Zusammenhang mit der Bildvorverarbeitung für die Registrierung multimodaler Bilddaten zu bewältigen. Der Schwerpunkt meiner Studie liegt auf der präzisen Überlagerung von Bildern, die mit verschiedenen Mikroskopiemodalitäten aufgenommen wurden. Dazu müssen Elektronenmikroskopbilder mit Lichtmikroskopbildern korreliert werden, die sich nicht nur in ihrer visuellen Darstellung, sondern auch in ihrer Pixelauflösung unterscheiden. Das Ziel der Vorverarbeitung ist es, Bildpaare zu generieren, die genau die gleichen Bilddetails aufweisen. Um dies zu erreichen, habe ich eine Template-Matching-Methode auf Basis des normalisierten Korrelationskoeffizienten integriert. Um Bildabmessungen und Pixelgrößen zu harmonisieren, nutze ich zusätzlich die Tatsache, dass beide Bilder dieselben zugrunde liegenden Strukturen darstellen und somit einen internen Standard für die Kalibrierung bieten. Derzeit gibt es keine Rechenwerkzeuge oder Softwarelösungen, die die Pixelgrößen von Bildern aus verschiedenen Quellen automatisch aneinander ausrichten können. Diese Kalibrierung der Pixelgröße ist allerdings ein entscheidender Schritt für ein erfolgreiches Template-Matching und die anschließende Registrierung von Bildpaaren; sie erfordert allerdings noch ein gewisses Maß an manueller Überwachung.

Nach der hier vorgestellten Vorverarbeitung können die Bildpaare als Eingabe für Bildanalyse-, Registrierungs- und Segmentierungs-Workflows verwendet werden, wobei sich potenzielle Anwendungen auch über diese Bereiche hinaus erstrecken. Meine schwach überwachte Bildvorverarbeitungs-Pipeline ist vielversprechend für die Integration in automatisierte Mikroskopieplattformen, da sie einen unbeaufsichtigten Betrieb ermöglicht und die Bildanalyse multimodaler Datensätze mit hohem Durchsatz erleichtert.

# Chapter 1: Introduction

Within just a few years, artificial intelligence (AI) has evolved from an unnoticed niche technology into a technology that has found its way into many areas of digital life. Especially its use in commodity applications like text or image generation is particularly fascinating and was unthinkable just some years ago. Current trends and modern research forecast, that AI will be able to provide solutions to complex problems without requiring any help, supervision or guidance in the future. Because the AI neuronal network is mimicking the neuronal system of a brain, it has the ability to learn by using machine-learning algorithms. Therefore, it can generate logical structures from unsorted or clustered complex data and extract the data pattern in order to learn, how to apply this pattern. Let's take a look at the direct contest between computer and human, using board games as an example. Chess is a very complex board game. In 1997, the IBM computer Deep Blue defeated the reigning chess grandmaster and world champion Garry Kasparow. However, this victory is solely attributable to Deep Blue's extremely high computing power; no neural network was implemented to Deep Blue´s algorithm. Go is another, even more complex board game. Due to the complexity and larger board with 19 x 19 fields, traditional brute-force algorithms like Deep Blue fails. To account for the complexity of this board game, AlphaGo is developed by Google's DeepMind in 2015[8]. It became the first AI that defeated a professional Go player. Besides of human-computer competition on board games and commodity applications, AI also has an impact on the scientific research. Predicting the 3D structure of a given protein sequence has been a great challenge and only experimental data could solve this prediction. With the AI program AlphaFold (also developed by Google DeepMind) it became possible, to predict the 3D protein structure for a given protein sequence of amino acids[9]. This was a scientific breakthrough. Therefore, the Google DeepMind's researchers received the 2024 Nobel Prize in Chemistry for protein structure prediction and computational protein design[10].

However, AI is not as successful in every area. Working with images, for example, still presents many challenges for the implementation of AI. Very basic pattern recognition systems are already installed in many cars, where the front camera registers traffic signs and forwards them to the driver via the display. However, when it comes to more complex, image based challenges, simple pattern recognition is not sufficient. For example, the registration (alignment) of two images coming from different sources is still a challenging

task for AI. The problem is, although the two images display the very same object, the appearance, the contrast or even the features in the display may be totally different due to the different sources. This is the case for correlative light and electron microscopy (CLEM), where, because of the different physical interactions of these two methods, the displayed features can have a very different appearance. Light microscopy (LM), especially fluorescence microscopy, is able to localize very specific molecules via a labelling with fluorescent dyes. On the other hand, electron microscopy (EM) is characterized by extremely high resolution and label-free imaging of all structures in the field of view. This combination is of special interest in biological applications, where the location of a particular molecule, e.g. a protein or a nanoparticle, within a cell needs to be determined. Here, the EM supplies the high resolution of the cellular structure and the LM localizes the fluorescently labelled molecule. The combination of those two information finally yields the localization of the labelled object (protein, nanoparticle) within the cell, displaying it´s environment with EM´s high resolution. However, since the information is coming from different microscopes, the final step is to align both images. This process is commonly known as registration.

In my thesis, I am going to apply AI for the registration of CLEM image pairs of different resolutions from different modalities (microscopes), presenting a workflow for the generation of automated CLEM micrographs. The main focus of my thesis is to provide in-depth information on the implementation of deep learning techniques across different or multi-modal imaging modalities with an emphasis on multi-modal microscopy samples.

The first chapter covers an overview of the two microscopy methods and provides definitions of certain image characteristic parameters as field of view (FoV), pixel size, and other key parameters that are important for multi-modal image registration. Subsequently, an overview of different generative AI models that can be used for image processing is given. Finally, this chapter concludes with describing the objective and motivation of this thesis.

# Microscopy methods

## Light microscopy

Light microscopy (LM) is an imaging technique that uses visible light and optical lenses to magnify and visualize the specimen, typically up to 1000-1500x magnification. It is widely

used for observing cells, tissues, and microorganisms, offering real-time imaging with relatively simple sample preparation[11,12]. As for nearly every microscopy method, its maximum resolution is diffraction limited to approximately half the wavelength of the used probe[13]. In case of light this would be roughly 250 nm. Classical wide field light microscopes use a parallel illumination of the specimen with white light and the magnified image is then generated by the objective lens. In this work, however, I use images from a confocal laser-scanning microscope, which has a very different working principle. Fig. 1 represents the schematic image of a confocal laser-scanning microscope. Here, a laser beam is scanned over the specimen. The objective lens focusses the beam to a narrow point and the emitted light from that particular point is then detected by a photomultiplier (or similar detector). In such a microscope, the excitation wavelength can be adjusted to the used fluorophore / dye and the emission band is selected by using special filters. Using this mode, dedicated fluorophores can be detected. However, the confocal laser-scanning microscope can also be operated in the reflected mode, where the light that is reflected from the specimen is detected. This kind of images more or less resemble a wide-field micrograph, as shown in Fig. 1. However, although using a focused laser beam, the resolution is still dictated by the diffraction limit.

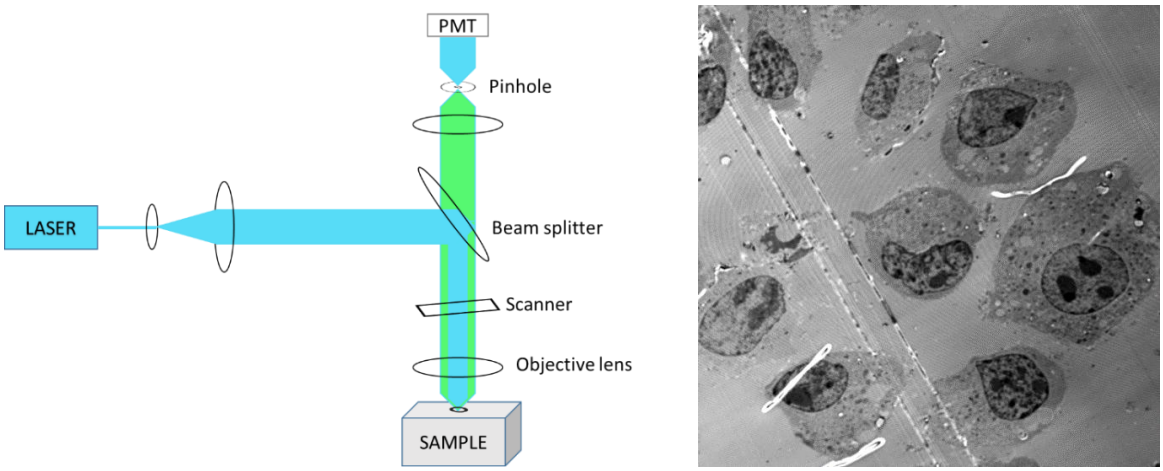


*Fig. 1: Schematic sketch for working principle of a confocal laser-scanning microscope with an example of a captured reflected channel LM image.*

## Electron microscopy

An electron microscope uses an electron beam as a source of illumination and the transmitted or scattered electrons are detected by a camera or a sensor to generate a micrograph. As like the light microscope, it is subjected to the diffraction limited resolution. However, because the wavelength of electrons is considerably smaller, a 200

kV electron has a wavelength of 2.5 pm, the resolution of an electron microscope is much lower. An electron microscope can reach a sub-nanometer resolution. Depending on the physical setup, we distinguish between a scanning and a transmission electron microscope. The scanning electron microscope basically scans a focused electron beam over the sample and detects the electrons emitted from the surface of the specimen[12,14,15]. The schematic design of both microscopes is presented in Fig. 2.

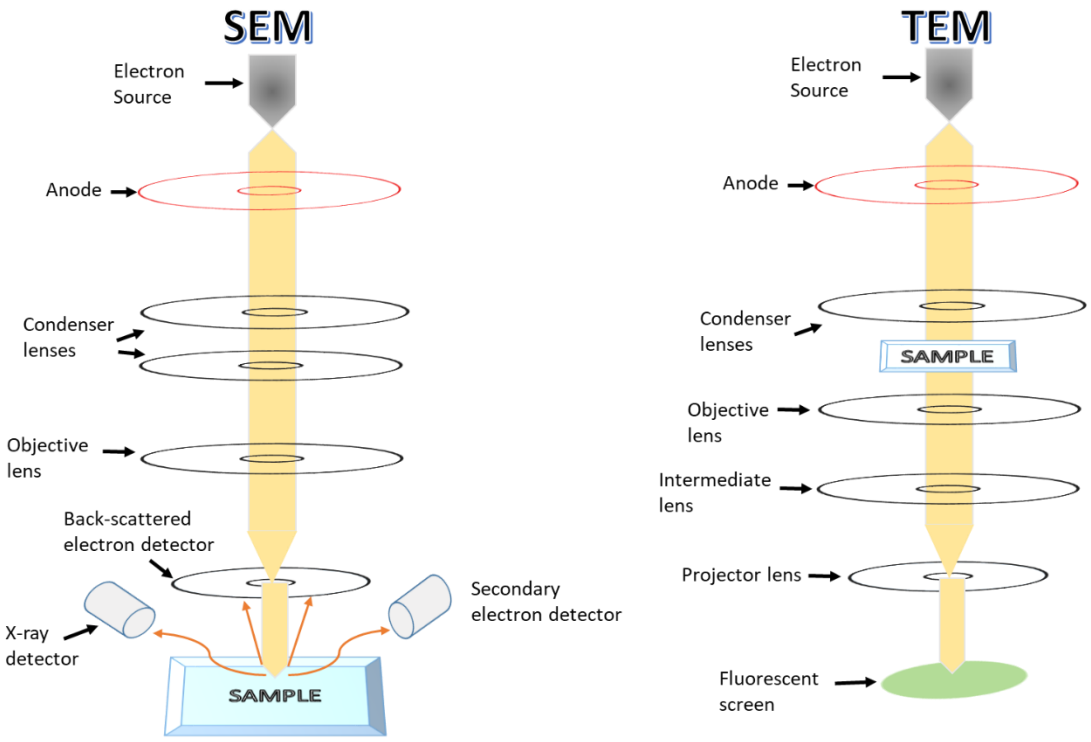


*Fig. 2: Schematic sketch illustrating the working principle of a scanning electron microscope (left) and a transmission electron microscope (right).*

In a scanning electron microscope, the image recording is performed by scanning a focused electron beam across the surface of a sample in a raster pattern[14]. As the beam interacts with the surface, it generates secondary and backscattered electrons, which are detected point by point. The signal intensity from each point is directly converted into a pixel brightness, creating an image of the local current of secondary or backscattered electrons from the surface of the specimen. The resolution of a scanning electron microscope typically ranges down to approximately one nanometre, depending on the instrument and sample, and is generally limited by the interaction volume and electron beam diameter.

On the other hand, conventional transmission electron microscope is a typical wide field microscopy method, where an ultrathin sample is illuminated by a parallel, high-energy electron beam. The transmitted electrons finally form a 2D projection of the transmitted

sample on a 2D camera detector. A conventional transmission electron microscope can achieve a resolution better than 0.1 nm, sufficient to visualize atomic structures[12,14,15]. As for any digital image, both scanning electron microscope and transmission electron microscope yield a 2D pixelated image, which is a 2D matrix of pixels with each pixel having a certain pixel value, which can be interpreted as an intensity or displayed as a brightness.

## Image characteristics of LM and EM

When imaging the same sample through LM and EM, the resulting micrographs significantly differ in their field of view, pixel size, resolution, and magnification. Common LM provides a large field of view (FoV), typically ranging from hundreds of micrometers to several millimeters, and is ideal for observing overall morphology or large structures. EM yields a smaller field of view but delivers high-resolution details.

One must bear in mind that every digital image is just a pixel matrix. However, when it comes to microscope images, so called micrographs, we have to consider the magnification of the microscope in order to measure the size of any displayed feature. Here, the pixel size, which is given as a length unit, is a very important parameter. The pixel size scales reciprocal to the microscope magnification; the higher the magnification, the smaller the pixel size of the respective micrograph. Fig. 3 demonstrates the relationship between FoV, pixel size and image dimension. Fig. 3 a shows an image of coins, acquired using an image dimension of 1500 x 1200 pixels. The FoV, i.e. the visible image section, is 264 x 211 $mm^2$ and the pixel size is 0.176 mm/pixel. When we acquire the same FoV but with a less image dimension, i.e. less pixels (Fig. 3 b and c), we can see, that the images become “pixelated” and finally unrecognizable (Fig. 3 c). In a microscopist’s language, decreasing the FoV is done by increasing the magnification. Because the camera system is the same, the image dimension stays constant (Fig. 3 d). One can consider this as “zooming in”. Here as well, the reduction of the image dimension results in pixelation and an increase in the pixel size (Fig. 3 e and f). Therefore, the relationship between FoV, image dimension and pixel size are rather simple, given by: FoV = pixel size * image dimension.

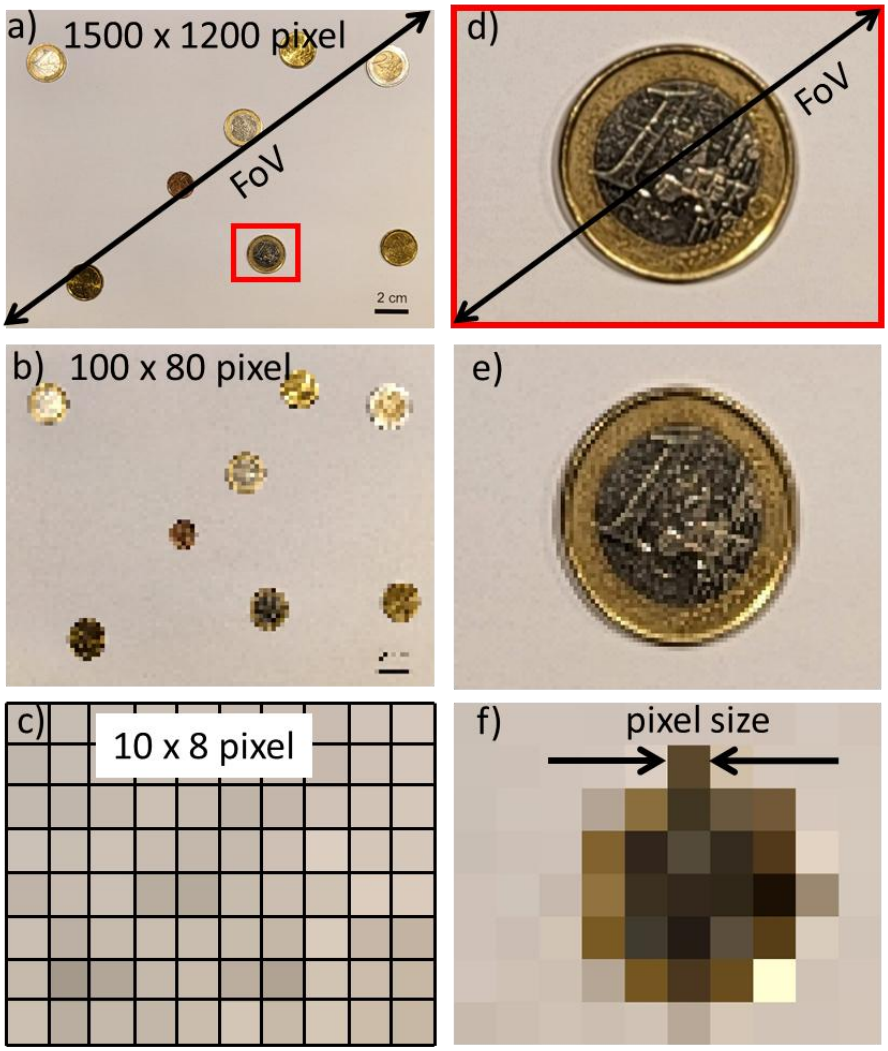


*Fig. 3: Example of Euro coins illustrating the three characteristic parameters of a 2D microscopic image. a) shows an image with image dimension of 1500 x 1200 pixels. The pixel size can be calibrated by the scale bar in the lower right corner. At this image dimension, the pixel size is calibrated to 0.176 mm / pixel, yielding a FoV of 264 x 211 mm. b) reducing the image dimension to 100 x 80 pixels does not change the FoV, but only the pixel size to 2.6 mm / pixel and c) further decrease of the image dimension yields a pixel size of 26 mm / pixel. d) Decreasing the FoV (red box in a) is like zooming in with a microscope. Because the camera system is the same, the image dimensions do not change. However, the pixel size has decreased to 0.0285 mm / pixel and the FoV becomes 43 x 34 mm$^2$. e) and f) with decreasing image dimensions the pixel size increases to 0.43 and 4.3 mm / pixel, respectively.*

I emphasize this correlation between pixel size and image dimensions so explicitly at this point because it is of essential importance for image processing, and I will revisit this topic in detail in Chapter 2 and 4. However, as simple as it may seem, accurately determining the pixel size in microscopic images is by no means trivial. There is no scale bar on the specimen like in Fig. 3 a) but instead, the microscope needs to be calibrated using a known standard. Calibration in microscopy, encompassing LM, SEM and TEM, is essential for accurate measurements and is often challenged by several factors. Magnification drift in the EM, caused by variations in lens current, temperature, or vacuum conditions can lead to discrepancies between actual and displayed magnification. SEM is susceptible to scan

distortions from nonlinear beam scanning, while stage or sample tilt errors impact scale accuracy. Detector inconsistencies and aging also affect measurement precision. Inaccurate calibration occurs due to incorrect magnification settings, camera length, detector geometry, or outdated reference standards and can lead to systematic measurement errors. Additionally, electron optics drift and changing imaging conditions may affect pixel size over time. To address pixel size calibration issues in EM, a regular calibration of the system using reliable reference standards like lattice spacing's or certified grids is necessary. Although LM generally operates at lower magnifications and resolutions (~200-300 nm) than EM (10 to 0.1 nm), inaccurate calibration of the optical system, such as improper scale bar settings or misalignment of calibration standards, can lead to measurement errors as well. Changes in lens focus, objective switching, or imaging conditions can also affect spatial accuracy. Consistent imaging conditions and routine checks help maintain measurement accuracy[11,12,14,15,16]. This consideration of pixel size calibration is important because we want to register multimodal images from different microscope sources with different calibration standards. If differences in calibration arise here, identical features in both images will have different sizes, which ultimately leads to misalignment during registration using deep learning algorithms.

## Generative artificial intelligence (AI) in microscopy

Generative AI is a subset of artificial intelligence that creates new content such as text, images, music, or different forms of data by learning patterns from existing data[17]. Unlike traditional AI that mainly analyzes or classifies, generative AI produces original outputs that mimic human creativity. It powers tools like ChatGPT for writing, DALL·E for image creation, and various models for generating audio or video[17,18,19,20]. Generative AI uses generative and discriminative models that learn the underlying patterns and structures of their training data as inputs and use them to produce new data. Generative models are a fundamental class of machine learning models focused on understanding and replicating the underlying structure of data, while discriminative models learn the boundary between different classes or categories. Generative models learn the full probability distribution of the input data and can sample new data points from this distribution. This makes them especially powerful for creating realistic synthetic data[21].

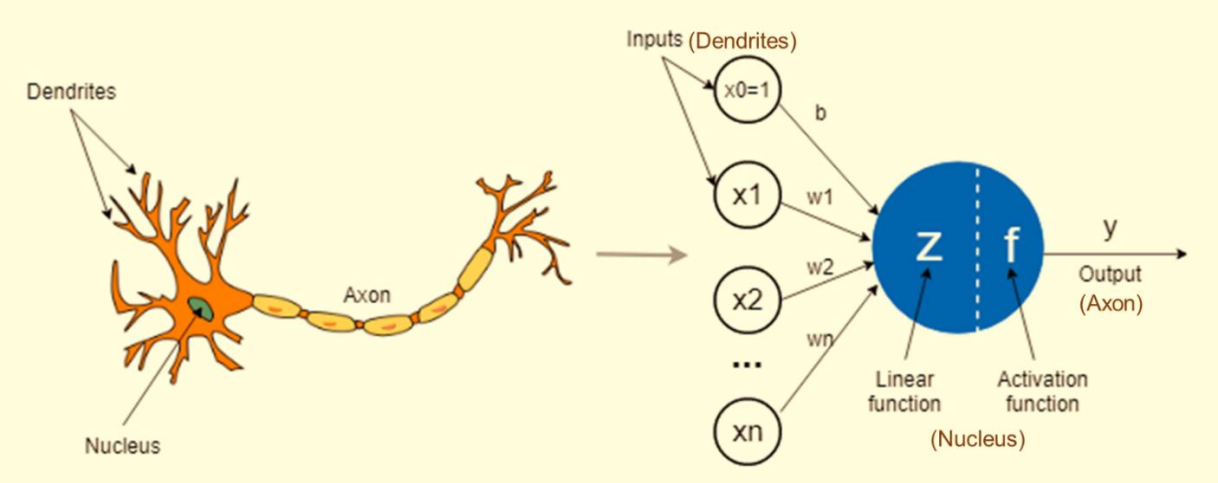


*Fig. 4: Typical similar representation of a brain neuron structure with an artificial neuron. (Digital free licensed image)*

The generative AI models, and all neural network models are made up of artificial neurons. The artificial neurons are designed to be similar to human brain neurons in structure and in working principle. A single artificial neuron can be compared to a biological neuron in the brain (Fig. 4). Here, the dendrites receive incoming signals (inputs) and the cell body (or soma) processes these inputs. Finally, the axon transmits the output to other neutron, connected by their dendrites. Artificial neurons are designed in a very similar way. The inputs are received using a weight function *w* and passed to an activation function – the cell body. Here, the processed signal is generated and then sent to the next layer as output, similar to the function of the axon. At the learning stage, the weights are adjusted in order to achieve the desired output, very similar to the learning of the brain, where the individual connections of the dendron are strengthened or broken down during learning.

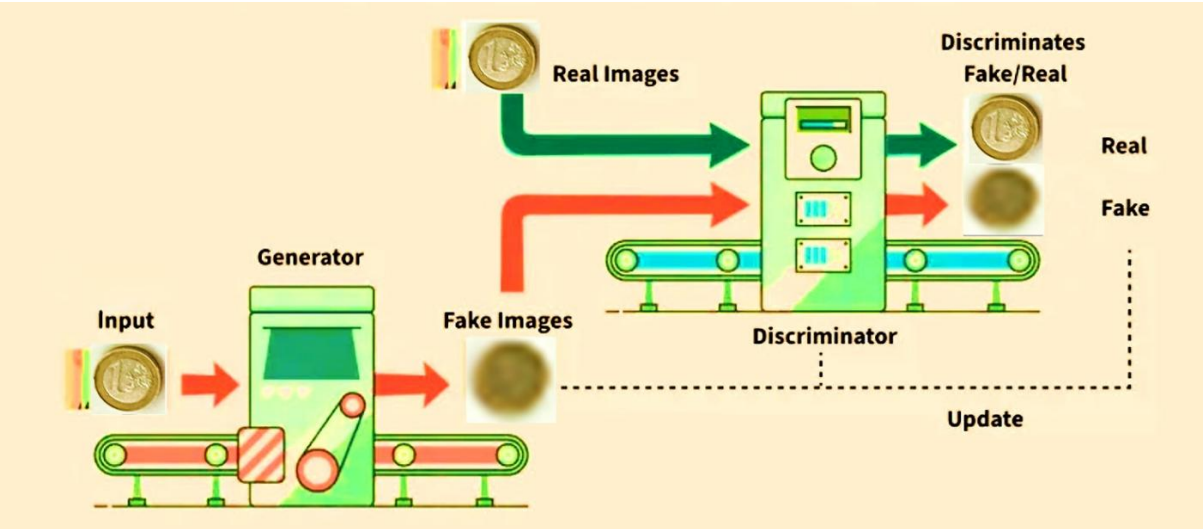


*Fig. 5: A pictorial representation of a GAN model.*

The categories of generative AI models include generative pre-trained transformers (GPTs), generative adversarial networks (GANs), variational autoencoders (VAEs) and diffusion models[17,18,20,21,22]. A pictorial representation of working GAN model is shown in Fig. 5 and presents the difference between generated real and fake images. A GAN model is a combination of generator and discriminator models that generates multiple real/fake images using the input images. Generative AI systems are described as multimodal as they are capable of interpreting and producing more than one type of data format, such as text, images, or audio. For example, GPT-4o is a multimodal model with the ability to both understand and generate content across textual, visual, and auditory domains. Generative AI models have transformed image processing by enabling the creation, enhancement, and manipulation of images with a high degree of realism and automation. These models use deep learning techniques e.g. GANs, VAEs, and diffusion models to perform tasks that were previously difficult or impossible using traditional image processing methods[1,2,16,17,18,20,23]. Some of the key applications of generative AI models are presented in Table 1.

Table 1: List of generative AI models used in different applications.

| **Applications** | **Challenge/Task** | **Generative Models** |
|---|---|---|
| Image generation | New visual content creation | GANs, Diffusion Models[17,22] |
| Super-resolution | Enhance image clarity | SRGAN, ESRGAN[24,25] |
| Inpainting | Restore or edit damaged/missing parts | Contextual GANs[26] |
| Style transfer | Apply artistic styles | CycleGAN, NST-GAN[27,28] |
| Image-to-image translation | Convert image formats or styles | Pix2Pix, CycleGAN[27,29] |
| Text-to-image | Generate images from descriptions | DALL·E, Stable Diffusion[19,22] |
| Facial editing | Realistic face manipulation | StyleGAN, FaceGAN[30,31] |

| | | |
|---|---|---|
| Medical image synthesis | Augment data for healthcare | MedGAN, VAE-GAN[21,32] |
| 3D reconstruction | Create 3D views from 2D inputs | NeRF, 3DGAN[33,34] |
| Data augmentation | Improve training data diversity | Any generative model[17] |

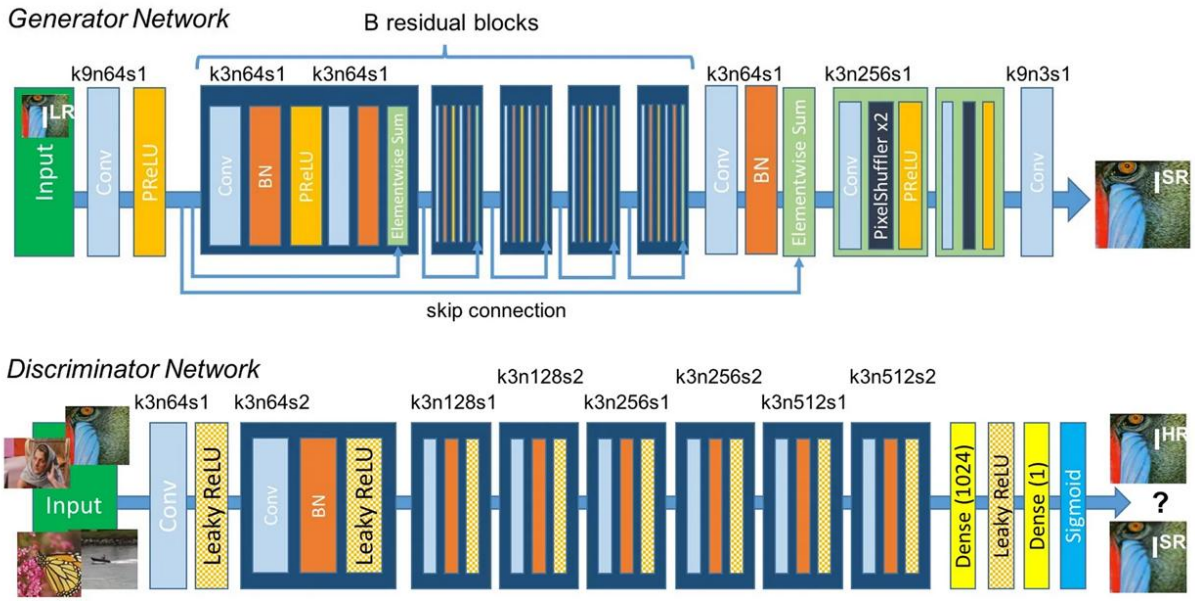


*Fig. 6: An architecture of a SRGAN model. © [2017] IEEE, Reprinted, with permission [15].*

Generative AI is increasingly applied to microscopy image analysis, and can be powerful tools for enhancing, interpreting, and generating microscopy data across various domains including image generation to segmentation, denoising, object detection, super-resolution and 3D image reconstruction[16,24,26,35]. Enhancing the quality of low resolution images, or to reduce noise within an image is the key challenge in the field of microscopy. However, super-resolution generative adversarial network (SRGAN) can be used to generate a high-resolution (HR) image from a low-resolution (LR) image that seems realistic and rich in fine details. The architecture of a SRGAN model is shown in Fig. 6 and represents the used models with their connected layers[24]. By transforming a LR micrograph into its HR counterpart, it improves the visualization of fine cellular and subcellular structures in tissues and organs[24,25]. In Appendix I, I present a CLEM micrograph which is registered using HR LM image-pair from LR images. These HR LM image-pair is generated using the SRGAN model.

## Objective of my thesis

The main objective of my thesis is to do automated CLEM, i.e. registering or aligning two images of the same object acquired using different sources and having different resolution, FoVs, pixel sizes and image dimensions without providing landmarks or any information of the image-pair. For image registration (IR), we use an image-pair consisting of an EM and a LM image, displaying the very same sample and FoV. Besides the automated registration, I would like to also highlight the need of a proper image pre-processing in order to achieve a suitable, multi-modal image-pair from both raw images which then can be used for the automated IR process.

For my automated IR workflow, any microscopic data, such as scanning tunneling microscopy, atomic force microscopy, electron microscopy, and confocal laser scanning microscopy can be used[36], provided that we have structurally similar images available. Here, I will limit myself to the challenges of IR in CLEM datasets[36]. EM images are characterized by high resolution and non-specific representation of the sample. LM images provide a limited extent to localize specific molecules or proteins using a fluorescent label[37,38]. This strength of LM can be used to visualize specific molecules or dedicated cell organelles. These two microscopy methods complement each other in an ideal way. Superimposing both images with pixel fidelity gains synergetic information. However, the images of both methods always originate from different devices and therefore generally have a) different pixel sizes, b) different pixel dimensions and c) show a different field of view, which must be superimposed by shifting, rotating, and stretching. The determination of the three transformation matrices containing combined information of translation, rotation, and stretch/shear transforms have the key to a precise registration (superposition) of the two input images.

The following chapters will address the key challenges and concepts, and provide a detailed overview of the research. A concrete, solution-driven approach and the resulting findings will be presented in the subsequent chapters.

**Chapter 2** focuses on the methodology, core concepts, challenges, and a brief overview of image registration challenges with multi-modal images. This chapter provides a direction that can be used for addressing these problems and highlights the machine/deep learning pipelines that is further discussed in detail in later chapters.

**Chapter 3** describes my unsupervised deep learning approach used for automated CLEM. It provides the neural network framework and explains the workings of my algorithm in detail. It showcases the flexibility and strength of the algorithm across different multi-modal image pairs and presents the various generated CLEM micrographs.

**Chapter 4** addresses the challenges of image pre-processing and the testing of irregularities within image pairs, including versatile, shift-rotated, and information-loss-affected image pairs. It showcases my solution-driven approach and presents the registered CLEM micrographs for these diverse and challenging image pairs.

**Chapter 5** presents the weakly supervised image pre-processing pipeline used to generate image pairs with accurate pixel sizes. It focuses on the challenges of image pre-processing and explains why it is necessary before performing any mathematical or computational operations on the images.

Finally, **Chapter 6** presents the overall conclusion of the preceding chapters and outlines the solution for achieving the main goal of registering two multi-modal images without requiring any prior information or manually selected landmarks i.e. training-free unsupervised deep learning CLEM (TFUDL-CLEM or automated CLEM). It also highlights the open challenges that can be addressed in the future.

# Chapter 2: Methodology

This chapter highlights the techniques, problems, and challenges involved in multi-modal image registration for CLEM micrographs. It provides an overview of currently available digital image processing solutions (software and tools) and briefly explains how automation can be achieved in CLEM micrograph IR. Finally, I briefly sketch my algorithmic approach, which is used to address these challenges. The working principles and concepts are described in detail in the following chapters 3, 4 and 5.

## Multi-modal image registration for correlated light-electron microscopy (CLEM)

IR is defined as the process of aligning two or more images of the same source or sample that differ in resolution, information content, display origin, field of view, or acquisition time[39,40,41]. The process of IR involves several steps, including feature detection (e.g., edges and corners), feature matching (e.g., correspondences or landmarks), transformation estimation (e.g., rotation, scaling, and translation), and image resampling and warping[3,4]. The primary goal of image registration is to establish a common coordinate system among two or more images to facilitate their comparison and integration. The core IR techniques, summarized in Table 2, employ different geometric transformation models as rigid, affine, projective, and deformable models for IR[5,42,43,44]. Various methodologies have been developed to achieve successful IR using either manual or semi-automated approaches, such as landmark-based methods[37,45]. The complex vector computations between images and the requirement for automatic differentiability are key factors that pose significant challenges to automated IR. Despite advances in mathematical modeling, computer vision, and machine learning, multi-modal image registration remains one of the most challenging tasks[5,46,47,48,49,50].

Table 2. Classification of image registration techniques.

| **Classification Criteria** | **Category** | **Description** |
|---|---|---|
| Transformation models[51,52,53,54] | Rigid | Translation and rotation only (preserves distances and angles) |

| | | |
|---|---|---|
| | Affine | Includes scaling, shearing, rotation, and translation |
| | Projective | Allows perspective transformations |
| | Non-rigid | Deformable transformations for local misalignments |
| | • Elastic | Physically inspired deformations (e.g., tissue movement) |
| | • Free-form | Uses splines or grid-based transformations (e.g., B-splines) |
| Matching method[32,55,56,57,58] | Feature-Based | Uses distinctive features extracted from images |
| | • Points | Matching key points like corners or blobs |
| | • Lines | Matching edge-based features like contours or ridges |
| | • Contours | Matching object boundaries or segmented shapes |
| | Intensity-based | Directly compares pixel values between images |
| | a) Correlation | Measures similarity using statistical correlation |
| | b) Mutual information | Measures shared information content between different modalities |

Several soft computing methods, including landmark-based and feature extraction techniques, as well as tools for image registration, are available for manual use or by training convolutional neural network (CNN) architectures with annotated or ground truth data[37,59]. Table 3 lists recent machine learning (ML) approaches used for IR across different datasets. However, no standardized algorithm or method has yet been developed for fully automated image registration due to numerous factors such as high dependency on the

images, illumination variations, similarity metrics, and complex non-linear distortions. These challenges often result in unsuccessful mappings or transformations within multi-modal images[4,57].

Table 3. Image registration based on different machine learning (ML) techniques.

| Category | ML Architecture | Description |
|---|---|---|
| Supervised learning | CNN-based registration[60,61] | Uses paired images and known transformations; learns feature correspondence and transformation estimation. |
| | Siamese networks[4] | Learns image similarity through contrastive loss; used to match features or patches. |
| | Regression networks[4] | Predicts transformation parameters directly (e.g., translation, rotation). |
| Unsupervised learning | Spatial transformer networks (STN)[54] | Learns to spatially transform input data with no ground-truth alignment needed (segmentation + registration). |
| | VoxelMorph[42,62] (deformable registration) | Learns deformation fields without supervised training; widely used in medical imaging. |
| | Cycle-onsistent GANs[4,27] | Enforces consistency in mapping between image domains without requiring aligned training data; used for multi-modal image registrations. |
| Reinforcement learning | Agent-based alignment[63]c | An agent iteratively chooses actions to align two images based on a reward signal. |
| Self-supervised learning | Contrastive learning[4,64] | Uses proxy tasks (e.g. patch matching) to learn useful representations for registration. |
| Hybrid deep learning models | Deep feature-based similarity + MI + traditional[4,65] | Combines deep feature extraction with mutual information and classic optimization techniques. |

In CLEM datasets, the registration of EM images with LM images is the key task, requiring cross-modal image registration (IR). In recent years, researchers have employed various approaches for the registration of biological and medical images[37,43,53,66,67]. For microscopy images, commonly used tools include ICY software[45], ImageJ[68] and eC-CLEM[69,70]. Table 4 lists scientific toolkits generally used for manual or supervised image registration. Both ImageJ and eC-CLEM perform IR by manually selecting landmarks in both images, followed by affine transformations. These tools are time-consuming and demand careful selection of landmarks.

Additionally, various approaches have been developed for computed tomography (CT) and magnetic resonance imaging (MRI) registrations, such as Voxelmorph[42], Hypermorph[71], and DRMIME[72]. Modern and hybrid deep learning (DL) methods use the concept of a fixed image and a moving image, the HR image is fixed, and the LR image is transformed until alignment is achieved using affine, spatial, or geometric transformations. The segmentation method for medical images, based on machine learning with U-Net architecture[35], is fast but suffers from reduced localization accuracy. However, methods based on this architecture are limited to small image sizes, typically 256 × 256 pixels. To process larger images, the common strategy is to divide them into smaller, and suitable sub-image patches[73,74,75]. This limit in image dimension is one of the main challenges for most of the approaches listed in Table 4 and therefore a new concept for IR is necessary.

Table 4. List of available tools/software for obtaining CLEM micrographs.

| **Software/Tools** | **Type** | **Key Features** |
|---|---|---|
| DeepCLEM[52] | Software | Automated fluorescence prediction and registration |
| CLEM-Reg[56] | Software | Point cloud-based registration, multi-modal support |
| MirrorCLEM[76] | Hardware-integrated | Real-time image overlay, automated stage movement |
| eC-CLEM[69] | Methodology | Multidimensional data integration, 3D imaging support |

| | | |
|---|---|---|
| MoBIE[77,78] | Software platform | Large-scale data visualization, multi-modal integration |

The first step toward successful IR involves computing complex-valued matrices, which can be performed using neural networks, as demonstrated in previous studies[51,79,80]. The role of mutual information (MI), a histogram-based technique in homography estimation, has been described and mathematically implemented within deep learning architectures[81,82]. Techniques such as segmentation, feature extraction, and the calculation of differentiable complex values using matrix exponentials have been applied to similar datasets using methods like DRMIME[72], MINE[83], CNN-MINE[84], Global-Net[60], U-Net[35], CNN[37,59] and GAN[53,85,90].

Several supervised approaches for CLEM registration have already been proposed[60,64,63,86]. However, these methods become infeasible for large images and remain largely unexplored in such contexts[61]. Additionally, supervised and training-based solutions are memory-intensive and require large amounts of annotated training data[87,88], which are usually not available. Manual annotation to prepare such a dataset is extremely time consuming and therefore not feasible.

In this dissertation, I address the key challenges of multi-modal image registration and propose a solution for automated registration using an unsupervised deep learning technique. Unlike artificially generated data, this work focuses on multi-modal images acquired from EM and LM micrographs of the same sample. However, these microscopy modalities differ in field of view, resolution, magnification, and contrast due to being captured by different sensors. Combining EM and LM yield a synergistic effect, EM provides highly resolved morphological information, while LM combined with fluorescence labelling localizes individual molecules within the specimen[36,87]. The resulting CLEM micrograph offers high-resolution imaging of cellular ultrastructure with additional molecular localization information.

Here, I present an image registration method based on training-free, unsupervised deep learning with hidden deep neural network layers. This approach integrates untargeted segmentation tasks with feature extraction and optimizes registration using a moving image framework. It handles non-linear deformations such as stretching and rotation, and is

flexible with respect to input data. The method is time-efficient and capable of processing micrographs up to 5120 × 3820 pixels on the GPU memory, supporting multi-channel image registration through the generated transformation matrix. Detailed explanations and results are provided in Chapters 3 and 4. With this, I also propose a breakthrough strategy to overcome GPU memory limitations, enabling successful registration of large CLEM micrographs (e.g. 11k x 11k pixels) by leveraging full CPU memory capacity. This approach can be extended to even larger image pairs depending solely on the available CPU memory.

Beyond CLEM, which has applications in medical and biological imaging, my registration approach can be easily extended to other fields where dual-modality images provide complementary information for example, satellite image alignment (remote sensing), astronomy, augmented reality, and autonomous driving systems[64,87,89,90].

## Automatic image pre-processing and generating image-pairs between multi-dimensional and multi-modal images

Image pre-processing[6] is a fundamental and crucial step in computer vision[46] and image analysis[1,23]. It encompasses a series of techniques used to prepare raw image data for analysis and model training. Specifically, image pre-processing involves preparing raw images to be fed into algorithms or machine learning models[49] for further analysis, learning, or decision-making. Raw images often contain challenges such as noise, inconsistencies, and irrelevant information. The aim of pre-processing is to remove such hindrances and enhance the relevant features of an image.

Effective pre-processing directly impacts the performance of machine learning algorithms by enabling them to learn meaningful patterns more efficiently and accurately. The goal is to improve raw image quality by eliminating inadequate information or extracting useful features, ultimately enhancing computational algorithm performance[91]. Typical pre-processing operations include resizing, normalization, noise reduction, image augmentation, and various transformations[6,7]. In this context, effective pre-processing is essential for achieving robust and reliable results in tasks such as object detection, image classification, image segmentation, and recognition across various biological and medical image datasets[23,48,91].

The aim of image pre-processing is to make data consistent across all images, ensuring that machine learning models can process it efficiently while reducing computational costs. It is crucial to balance improving model efficiency with preserving the necessary information for accurate analysis[46]. For precise analysis and better accuracy in various microscopy images[11], image pre-processing plays a vital role and remains a fundamental area of research and application, facilitating advancements across industries ranging from healthcare to autonomous systems[16]. Table 5 lists common image processing tools generally used to perform various operations on images. However, the image processing (either pre or post processing) tools listed in Table 5 are not autonomous but require human intervention.

Table 5. List of available image processing tools/software.

| **Software** | **Key Features** | **Use Cases** |
|---|---|---|
| ImageJ/Fiji[92,93,94] | Rescaling with various interpolation methods<br>Advanced image registration (TurboReg, StackReg)<br>Multimodal registration (MultiStackReg) | Align and rescale images to a common resolution and pixel size<br>Multimodal image alignment |
| Adobe Photoshop[95] | Resizing to match FoV and pixel size<br>Smart objects for quality preservation<br>Manual image alignment | Align images for design or analysis<br>Rescale images of different resolutions |
| 3D Slicer[96] | Rigid and non-rigid image registration<br>Resampling to uniform voxel size and resolution | Align and resample images for multimodal analysis |
| MATLAB[97] | Image resizing & resampling using interpolation methods, registration with imregister, batch processing via scripting | Automate image alignment and Pre-processing<br>Align images from different modalities |

| | | |
|---|---|---|
| SimpleITK[98,99] | Resampling and registration for images with different resolutions<br>Supports rigid and non-rigid registration | Align images across modalities<br>Resize and resample for multimodal analysis |
| Elastix[55] | Multimodal image registration (rigid/non-rigid)<br>Resampling to match resolution during registration | Register and resample images to a uniform resolution and pixel size |
| Bio-Formats[60,100,101] | Integrates with ImageJ/Fiji for handling various microscopy formats<br>Supports multimodal image registration | Align and process images from different microscopy techniques |
| ITK[102,103] | Advanced registration algorithms<br>Supports resampling to match pixel size and FoV | Align and resample images from different modalities |

In Chapter 5, I highlight the image pre-processing pipeline used to generate image pairs for subsequent creation of CLEM micrographs[39]. My approach is based on template matching (TM), a computer vision technique that compares portions of an image (templates) with a source image to find regions that best match the template[104,105,106]. Although TM is a basic method compared to more advanced image recognition techniques, it is useful in cases where simple alignment between template EM and source LM images is required. Typically, the FoV of an LM image is much larger than that of an EM image. The goal of TM is to locate the exact image area of the EM image within the LM image in order to then crop it out. In this way, we create an image pair with the same FoV, which is then ideally prepared for the subsequent IR.

I implemented the TM approach within my supervised image pre-processing pipeline and experimented with various aspects. TM locates the initial position of the template image within the source image. In this pipeline, a LM image with a large field of view (FoV) serves as the source image, while a binned-down EM image is used as the template.

Detailed explanations and results are provided in Chapter 5. The resulting image pair serves as an optimal input for the subsequent machine learning registration process. Moreover, this pre-processing approach can be easily adapted to other multimodal image systems[6].

Finally, combining the IR with the TM pre-processing, I present a combined toolkit for a weakly supervised registration of multimodal images. In the first step, a multi-modal image pair is generated from raw images using a weakly supervised image pre-processing algorithm, followed by a registration method based on a training-free, unsupervised deep learning technique. Because the EM and LM image pairs are registered automatically, my workflow is termed Automated CLEM, and the corresponding CLEM IR method is named TFUDL-CLEM (Training-Free Unsupervised Deep Learning for CLEM).

# Chapter 3: TFUDL-CLEM Algorithm

The following Chapter 3 is based on the manuscript "TFUDL-CLEM: A Training-Free Unsupervised Deep Learning Registration Method for Correlative Light and Electron Microscopy", submitted to Nature Communications Biology. At the time of writing, the revised version is under review. For the thesis, this chapter was extended with additional details and provides detailed information of my automated CLEM algorithm, as well as a demonstration of its performance. The verbatim use of the manuscript text is indicated by the use of the **Garamond** font.

Before the images from the different sources can be superimposed at all, one has to make sure that they match in both pixel size and pixel dimension. Usually, LM images have a larger pixel size than EM images due to the physically lower resolution. Likewise, the pixel dimension of the images must match. Usually, in addition to the fluorescence information, a confocal laser scanning microscope (cLSM)[107] also provides a reflected image, which roughly corresponds to a bright field image. Accordingly, the LM data in my work has at least two channels. I found, that using the reflected channel for image registration gives better results. This is most likely due to the fact that the reflected image contains much more information and features than the fluorescence image. However, with the calculated transformation matrices, the fluorescence channel can then be overlaid on the EM image quite easily with pixel accuracy for the CLEM output.

The first step in our work is image pre-processing, where the LM images are upscaled to achieve an image format identical to the EM image. This pre-processing leads to a blurring of the LM image channel, but no loss of information occurs. Subsequently, a Gaussian pyramid from the EM and LM image pair is formed with the original resolution at the bottom followed by the blurred images on the higher levels. For each level, the images are binned by a factor of 2, reducing the pixel number by a factor of 4. Here, we have implemented 3 levels in total. This downsizing is necessary to simplify the optimization and to speed up the process.

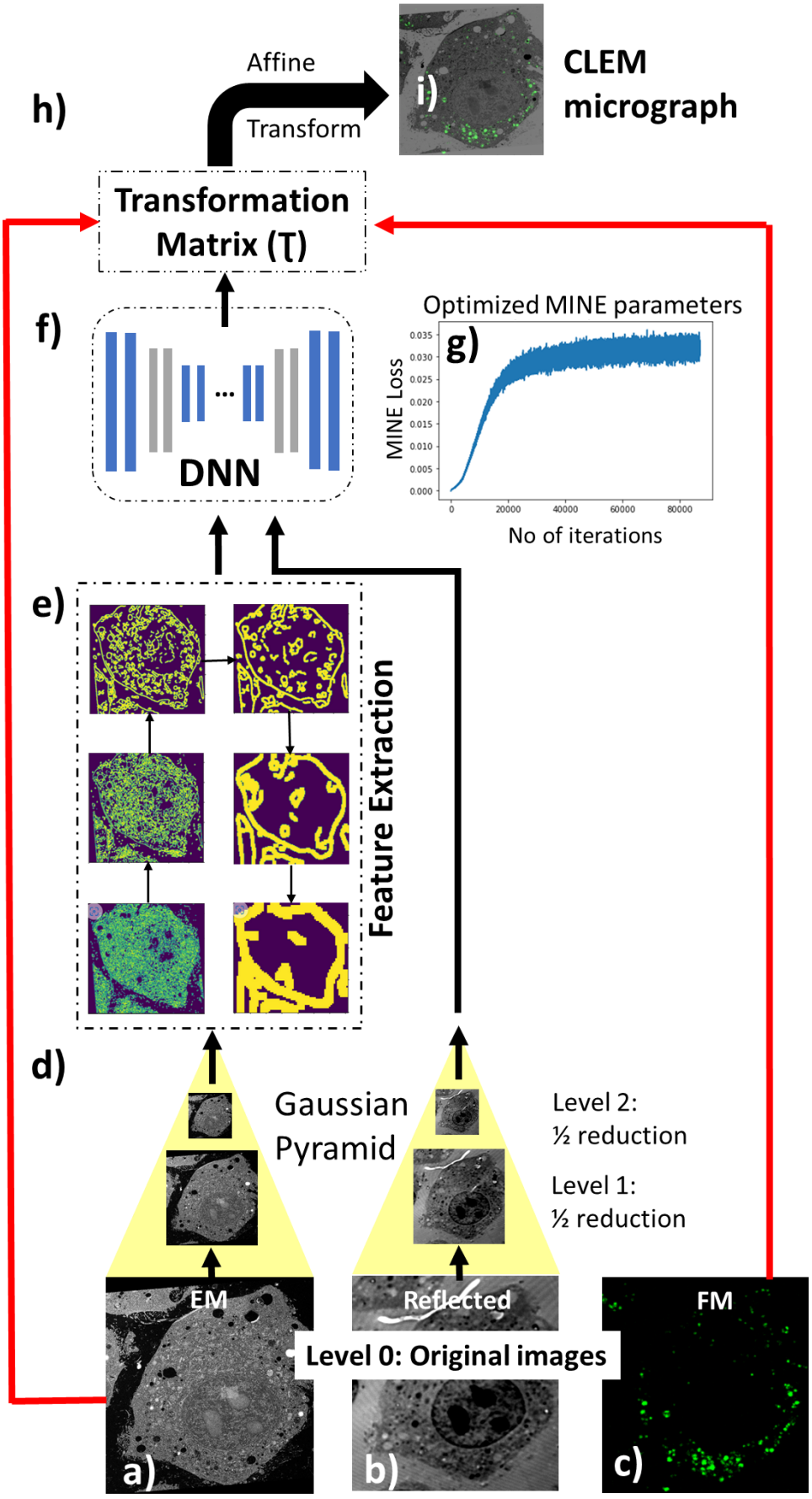


*Fig. 7: Schematic representation of unsupervised learning for the first image pair, starting with the input images consisting of the a) electron microscopic image, b) the reflective- and c) the fluorescence channel of the light microscopic image. d) After pre-processing the original images to the same image*

*dimension and pixel size, a 2-level Gaussian pyramid is generated from both the EM image and the reflective channel of the LM image, reducing the image dimension by a factor of 4. e) Subsequently, the EM image is subjected to another 6-stage FE before it is mapped with the LM image using a DNN f). Here, the concept of a moving image is applied to the LM image and the DNN optimizes the MINE loss. g) After 87000 iterations, the DNN is forced for an early stop. In addition to the parameters for the quality of the mapping, the DNN provides a transformation matrix that describes the affine transformations for the superposition of the EM and LM images. h) This matrix is then applied to the original unreduced fluorescence image and superimposed on the EM image. This completes the unsupervised learning step yielding the CLEM image i) and the DNN is ready to process additional image pairs based on this learning process.*

In the next step, the downscaled EM image from the top of the Gaussian pyramid is fed to the feature extraction (FE) process. FE results in a segmentation of the input image by applying canny edge detection. The aim of the FE is to reduce the complexity of the image and to emphasize smart structural features as shown in Fig. 19. The pixel size of the image is maintained, but the depth of information is reduced from a greyscale to a binary image. This process is repeated up to level 6, yielding a binary image which contains the most dominant structural features of only the EM image. The final sub-pixel image at level 6 is then used for obtaining the coordinates within the homogeneous plane and provides the values to flatten the information with the corresponding Gaussian-blur EM image of adjacent (nearest neighborhood) sub-pixels in the reflected channel of the LM image. This feature mapping is achieved through a deep neural network and we optimized the homography hyperparameters using different learning rates at each hidden layer of neural network and regularly optimize for the minimized neural estimation loss (MINE). We assume that the DNN accesses the last three FE levels in order to optimize the mapping. Here, we use the concept of moving one image (the LM image) over a fixed image (here, the FE EM image). Fig. 7g and Fig. 14 show the development of the MINE loss optimization with the number of iterations. The optimum number of iterations was determined empirically and set just before the abrupt increase in the MINE loss (Fig. 14). We found that exceedingly approx. 87,000 iterations will lead to overfitting e.g. the quality of registration will reduce and can even lead to mirroring effects (Fig. 13). Therefore, in our IR architecture workflow, the concept of early stopping is applied to optimize the DNN algorithm after 87,000 iterations.

Once the mapping of the two input images is completed, the vector values of the transformation matrix are calculated by the neural network using the differentiable matrix exponential[72,80]. Using these, the fluorescence image is transformed to match the EM image, and both images are superimposed yielding the final, registered CLEM image. On a common computer with one NVIDIA GPU, the total optimization takes approx. 3 h on a 5120 x 3840 pixels image pair, the registration result is shown in Fig. 11. The maximum image size depends solely on the memory capacity of the GPU. It is therefore also possible to process larger

images. Fig. 17 shows the registration of an 11k x 11k pixel image on an NVIDIA H100 GPU. The computing time was approx. 3 h.

For testing, we used datasets comprised of biological CLEM images. Cells were incubated with fluorescently labeled nanoparticles and subsequently fixed using high-pressure freezing (HPF) followed by freeze substitution using heavy metal stains and epoxy resin. Finally, thin sections of the resin-embedded cells were inspected by confocal laser scanning microscopy and electron microscopy, using the same sample section. Fig. 8 displays the results obtained from such a dataset in comparison to the manually registered image pairs using the ImageJ transform landmark correspondence plugin (TLC). EM images were acquired in a scanning electron microscope (SEM) using the backscattered electron signal. The LM data contains two channels, the fluorescence channel of the nanoparticles and the reflected channel (Fig. 13). All image data details for the presented dataset images of EM-LM are given in Table 6.

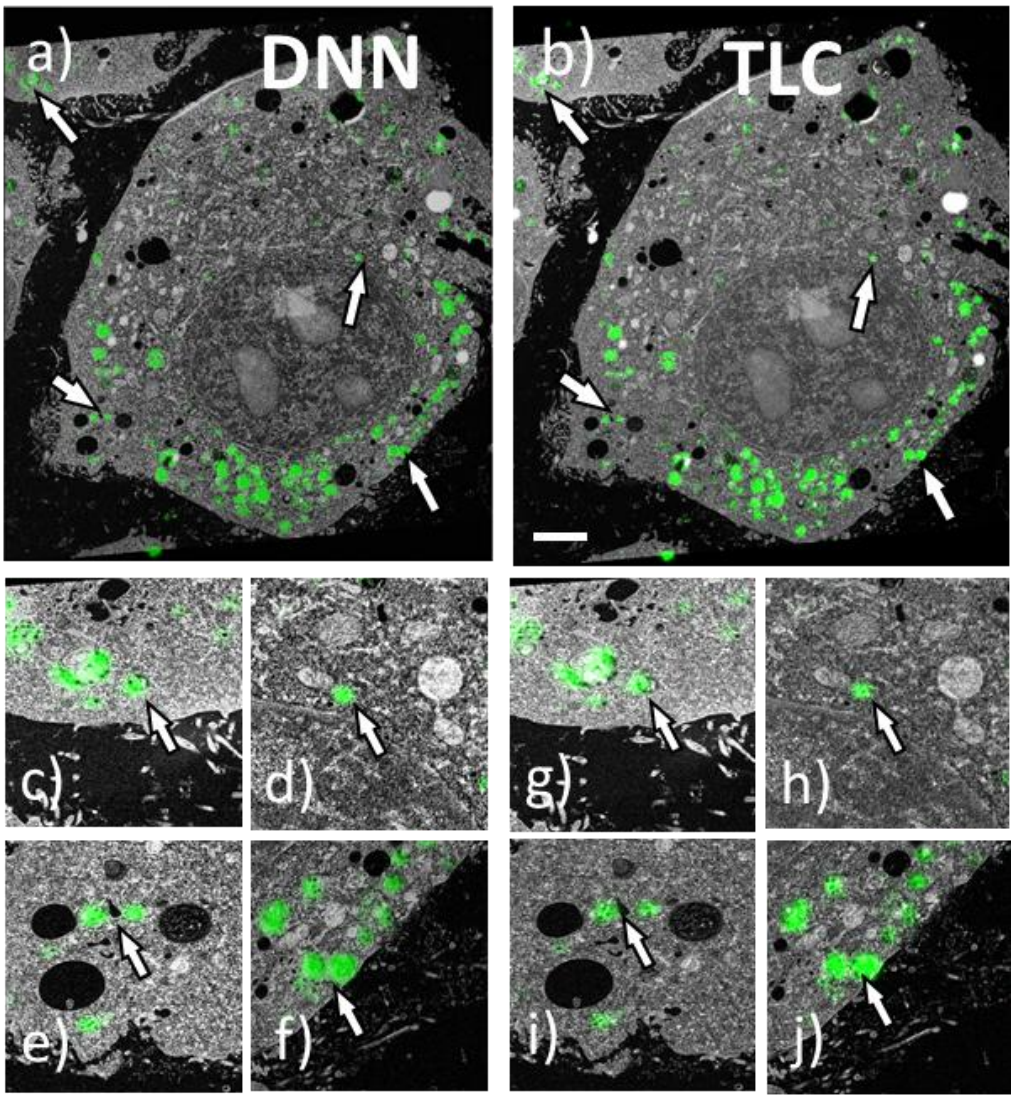


*Fig. 8: Comparison of the registration results between our DNN approach (left) and landmark-based registration using the TLC plugin for ImageJ (right). It can be seen that our DNN gives almost identical results as the human-assisted landmark-based registration using the TLC plugin. The arrows in a) and b) mark the positions, which are shown in higher magnification in c) to j). The registration results of our DNN and the manual method are almost indistinguishable. Scale bars in b) 5 µm.*

The initial optimization result of the first image pair in comparison to the manual registration with the TLC plugin is displayed in Fig. 8. The information content of both CLEM images is identical. The areas marked with arrows in a) and b) are shown with higher magnification in c) to j) for better recognition. The comparison of both methods shows that both overlay the target structures well with the fluorescence signal. The target structure, i.e. the PS NP, can be seen well in the EM image at higher magnification. Hence, our DNN approach identifies the features as precisely as the manual, landmark based TLC approach.

A prerequisite for an accurate registration is, however, a thorough image pre-processing (Fig. 15 in chapter 4). In general, there are several aspects that need to be addressed: i) Most crucial is the adjustment of the pixel size between both images (Fig. 16). Usually, this is done by

adjusting the image size of the LM image. This needs to be done manually via calibration measurements based on common landmarks in both, EM and LM images. ii) When necessary, a rotation between the images should be compensated. iii) The image dimensions (i.e. the height and width in pixels) of the EM and the LM image have to be equal. This is done by cropping the images yielding the same FoV.

## Maximum image dimensions and pixel size calibration

The maximum processable image dimension depends on the size of the GPU memory. For a common PC with a NVIDIA RTX 3080 GPU, we were able to process image pairs up to dimensions of 5120 x 3560 pixels. When using a NVIDIA H100 GPU, image pairs up to dimension of 11k x 11k pixels can be processed. However, pixel size calibration plays an increasingly important role with increasing image size, as shown in Fig. 16 and Fig. 17. At a certain point, the pixel size needs to be more precise than possible by manually comparing common landmarks in the image. As a consequence, the registration shows an increasing offset from the upper left corner of the image to the lower right corner (Fig. 17 i). The left part of Fig. 9 displays the registration of an 11k x 11k pixels image pair, processed on a NVIDIA H100 GPU. For clarity, we only present the overlay of the EM and the reflected LM channel, because the offsets are more visible than in the FM overlay. When inspecting the markers that can be assigned unequivocally, an increasing offset is observed with increasing distance from the upper left corner of the image (Fig. 9 c to f).

To overcome this problem, we used an alternative approach for registering this large image pair. Instead of processing the full-sized image, we shrink it to a 1k x 1k pixel image pair (10x binning). The calculated transformation matrix, however, was then applied to the full-size image pair (Fig. 9 right). Inspecting the same landmarks reveals a successful registration (Fig. 9 c' to f').

The physical resolution of the LM image determines the maximum permissible binning factor. As the physical resolution of the LM image is at least two orders of magnitude larger (in the sense of worse) than that of the EM image, we do not lose any information with this procedure. Instead, we achieve the best possible registration without the inaccuracy in the pixel size calibration leading to an offset. In principle, we reduce the information density of the EM image to that of the LM image. This approach also reduces the processing time (see Table 6). This reduction of the image information can be taken a little further, but only up to a reduction to a 500 x 500 pixel image pair (Fig. 18). A further reduction leads to a failure and an error in

our software, presumably because no information is contained in the level 6 feature extraction, as can be seen from Fig. 19 (bottom left).

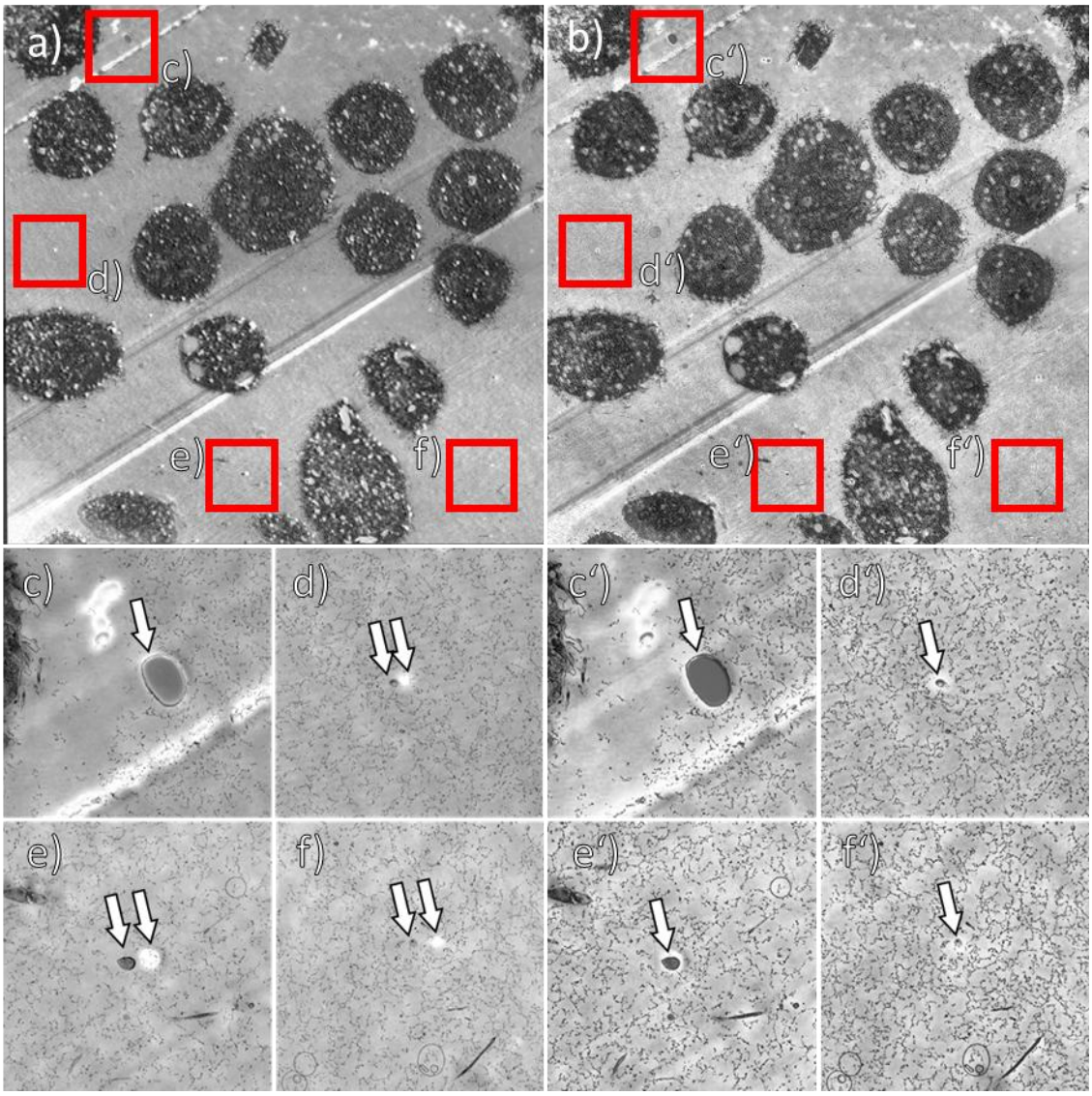


*Fig. 9: Correlation between image dimension and accuracy of pixel sizes calibration: a) left side of the panel shows the registration of a 10116 x 10080 pixels image pair, cropped from a region of the dataset shown in Fig. 17. Image processing was done using a NVIDIA H100 GPU, processing time: 2 h and 20 m. Displayed is the overlay of the EM image with the reflected LM channel, as the offset between the two image channels is much easier to perceive in this display than in the overlay with the fluorescence channel. Closer inspection of common, distinctive landmarks (c to f) reveals an increasing shift with increasing distance to the 0, 0 origin of the image at the top left edge. b) right side of the panel shows the same initial image pair, but binned down by a factor of 10 prior to processing. Processing was done on a NVIDIA RTX 3080 GPU, processing time 45 m. Afterwards, the calculated transformation matrix was applied to the original image pair, to retain the full resolution of the EM image. Closer inspection shows no noticeable offset between EM and LM channel (c' to f').*

## Tolerance against image shift and rotation

In the example in Fig. 8, the two input images were already manually processed during image Pre-processing so that both, the image shift and the rotation between the two input channels were as small as possible. Under realistic conditions, however, it is hardly possible to achieve

perfect compensation of image rotation and shift, especially since the examined specimen usually has to be transferred between two different microscopes. Therefore, we evaluated the tolerance limits for shift and rotation between the image pairs. First, the images were aligned as precisely as possible in order to then shift or rotate the FoV of the LM channel accordingly. Details on image Pre-processing can be found in the text below. The effect on the superimposed image is best recognized by the superimposition of the EM image with the reflected channel of the LM image (Fig. 10). The corresponding overlays with the fluorescence channel are shown in Fig. 21.

For the evaluation shown in Fig. 10, we used an image pair with pixel dimensions of 5120 x 3560 pixels. Fig. 10 e) displays the superposition of an almost perfectly aligned image pair. As expected, the superposition is successful. We then shifted the FoV of the LM channel (panels a) to i)) and the algorithm was able to compensate this shift. The FoV shift is visible by an offset between the LM and EM image, as indicated by the small white arrows. Surprisingly, the maximum tolerable image shift depends on the direction. We find maximum values of 80 and -250 pixels (corresponding to 1.6% and 4.9%) for a shift in x-direction and 100 and -120 pixels (corresponding to 2.8% and 3.4%) in y-direction. The same limit values apply for the diagonal elements as for the orthogonal shifts, i.e. we can compensate for a shift of -250 and -120 pixels in the -x and -y directions, for example (Fig. 10 c).

For rotation, we find tolerance values of 10° and -15° (Fig. 10 j and k). Above these values the superposition fails. And as already observed for the diagonal shift, a combination of rotation and shift can be compensated for each transformation as shown in Fig. 10 i. The reason for the observed directional anisotropy (e.g. the shift tolerance in -x direction is larger than in +x direction) is unclear. We can only speculate that it could probably be due to the tolerance shift that is dependent on pixel-wise differences in the image-pair, and shows altogether effect of affine transformation in the projective space and does not show similar corresponding transformations.

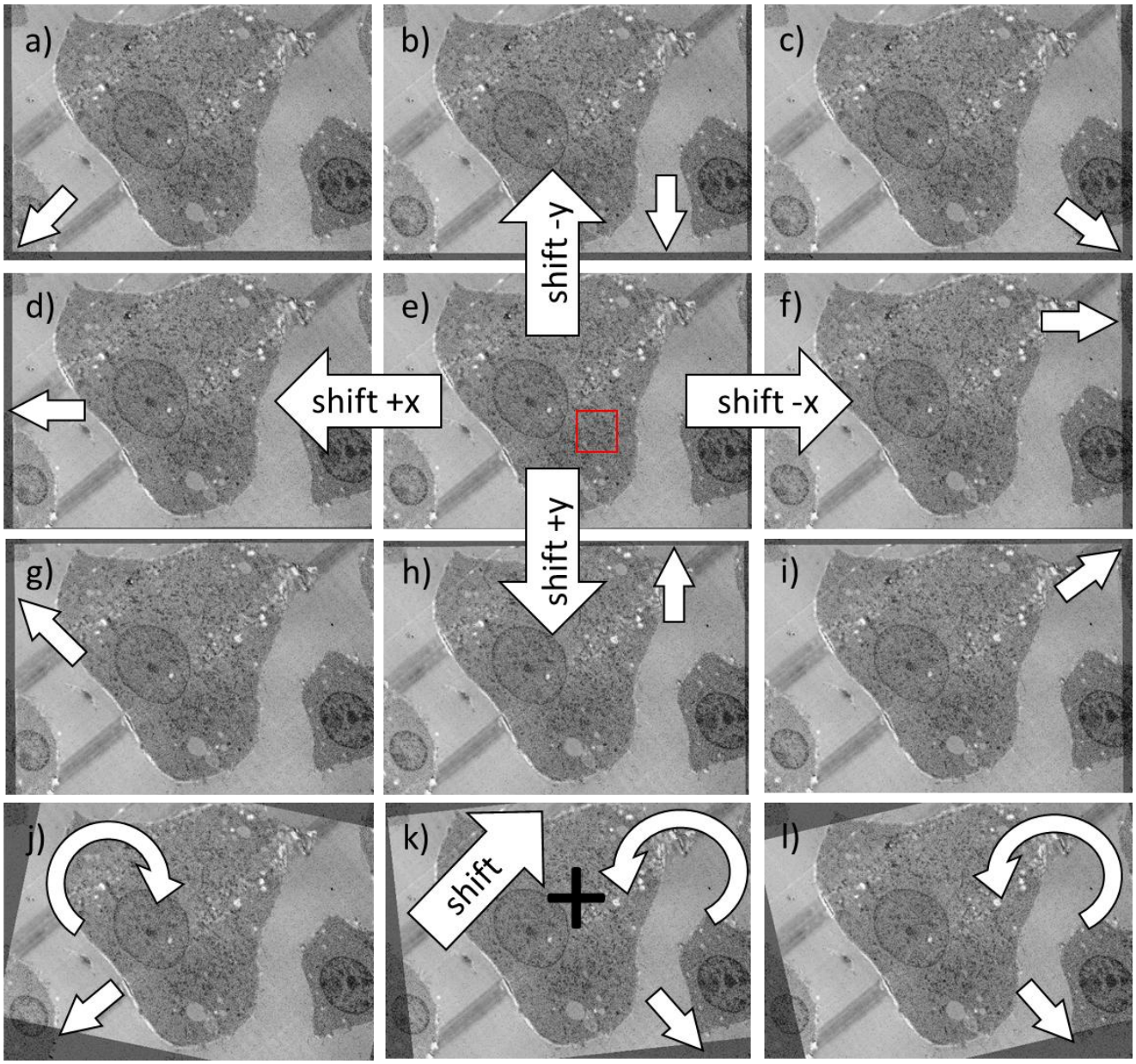


*Fig. 10: Tolerance against image shift and rotation. Here we show the overlay of the EM image with the reflected LM image, because the compensation of the algorithm (small white arrows) can be seen much better here (small white arrows) than in the CLEM images with the fluorescence channel (Fig. 21 left). e) displays the overlay result of the initial image pair, where the shift and rotation in the FoV have been compensated manually in the image Pre-processing The red box indicates the zoomed area shown in Fig. 14 a) to i) show the results of an additional image shift of the LM FoV. The maximum tolerable shift in x-direction is +80 and -250 pixels, this corresponds for a 5120 x 3560 pixels dimension to 1.6% and 4.9%, respectively. In y-direction the maximum shift is +100 and -120 pixels, corresponding to 2.8% and 3.4%, respectively. For the diagonal elements a), c), g) and i), the respective maximum shifts in x- and y-direction can be applied simultaneously. A rotation between the input images can be compensated for up to angles of j) 10 ° and k) -15 °. And even in combination, we can combine the maximum values for rotation and image shift without a noticeable deterioration in the overlay result: l) -15 °rotation and -250 pixels x-direction, -120 pixels y-direction.*

## Versatile datasets

We have seen that our software has only a few requirements for the image pairs, primarily the pixel size and the FoV of both images must match. In principle, we can therefore register any image sources. Therefore, to verify the versatility of our approach, we tested image pairs of

different samples and with different EM sources. Fig. 11 shows the CLEM results for different samples on the one hand and for different EM image sources, e.g. from SEM and TEM, on the other hand.

Fig. 11 a displays the registration of the same sample of intracellular PS nanoparticle as shown in Fig. 8 but at a higher magnification of the EM image, e.g. a smaller pixel size. In Fig. 11 b the input EM images source is a TEM image of a cell section with an inverted contrast compared to SEM. This is one of the advantages of our approach. It is not necessary for the input images to always come from the same sources; in principle, any image source is suitable for our method. Moreover, we are not necessarily limited to cell structures, as shown in Fig. 11 c), where we show the registration of fluorescent nanodiamonds dispersed on a glass surface. Finally, Fig. 11 d displays the registration of a multicolor fluorescence LM image, which demonstrates the advantage of using the calculated transformation matrix to register more than one FM channel. More examples on various specimens are shown in Fig. 22.

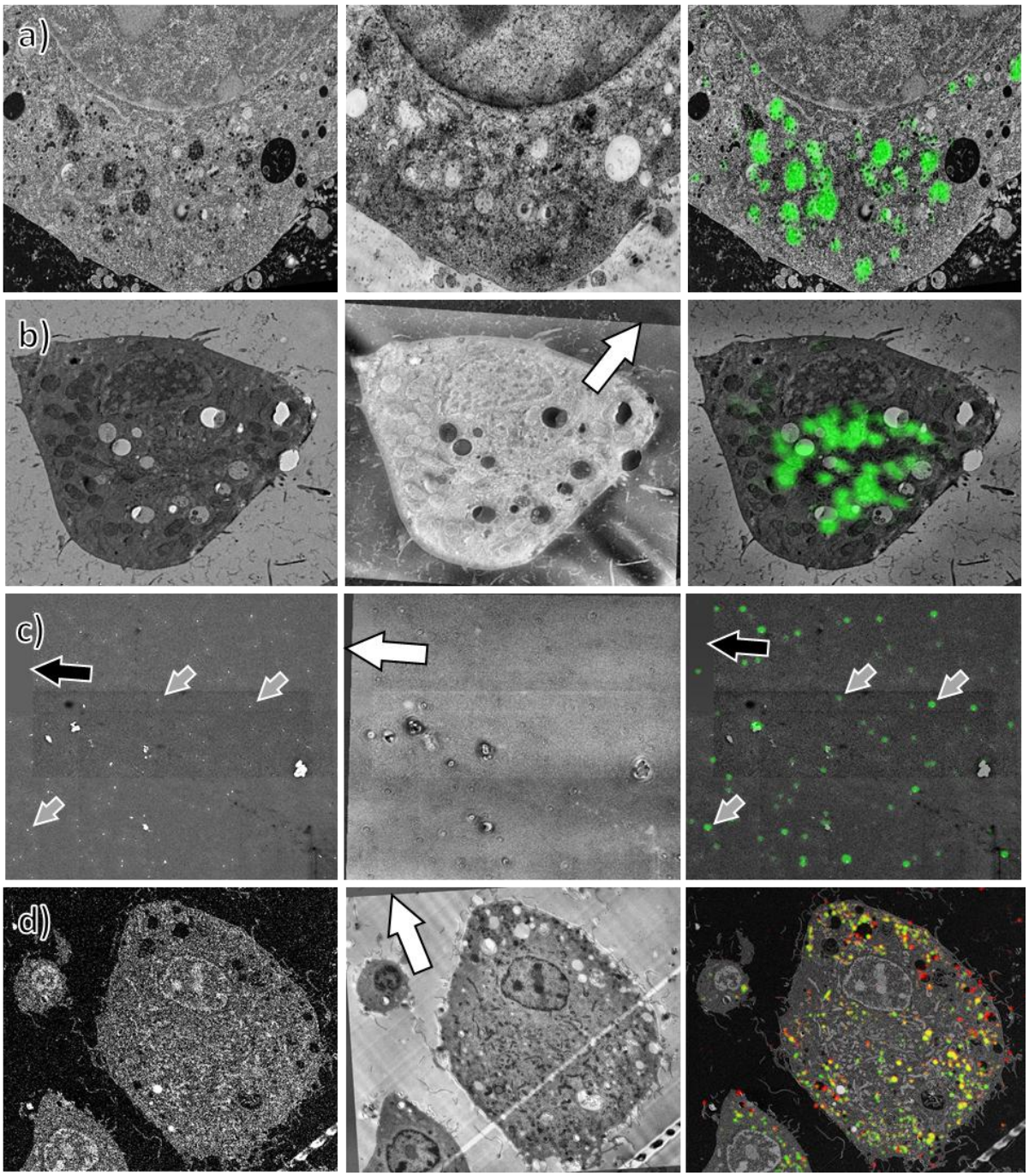


*Fig. 11: CLEM results for various experimental data sets. The CLEM overlays shown are from different samples as well as from different EM devices. The left column shows the EM input image, the right column shows the CLEM registration generated and the middle column displays the overlay of the EM with the reflected channel. This illustration is the best way to evaluate the accuracy of the registration. In particular, the rotation between the EM and LM channels can be clearly seen in this representation (white arrows). a) Same sample as in Fig. 8 but at a higher magnification of the EM image. b) Here, the input source for the EM channel is a TEM image. In comparison to a SEM image, the contrast of a TEM image is inverted. Here, fluorescently labelled intracellular nanofibres were localized using CLEM. But our software is not just limited to biological data sets. In c) we demonstrate the CLEM Registration of fluorescent nanodiamonds dispersed on an indium-tin-oxide (ITO) coated glass cover slip. In the SEM image the nanodiamonds are well perceived (grey arrows) and their fluorescence can be correlated to their morphology. Here, the SEM image is composed of several individual acquisitions using common stitching algorithms. On the upper left there is an area of missing information in the EM channel (black arrow) which has been filled with a gray value. The registration by our software was yet successful. d) As our approach is based on the calculation of a transformation matrix, it is not restricted to one channel only, but it can also handle multicolor fluorescence by registration of several fluorescence channels.*

## Pending challenges

It is possible that a data set may contain image pairs that are not ideal. In particular, we want to discuss a case here where part of the information is missing in one of the images. For example, if we have an image pair in which the image area of the EM image extends beyond the FoV of the LM image (the area marked with "padding area" in Fig. 23 a, the information of the LM channel is missing in this area. However, since our software only accepts image pairs with the same pixel dimensions, this missing area must be added. This can be easily achieved by simply adding the missing area to the image, as in Fig. 23 d, the so-called padding. The pixel value of the padding can of course be freely selected. For this missing information problem, we have observed, that zero-padding i.e. padding with pixel values of zero (black)- causes a failure in the registration (Fig. 23 e), whereas grey padding i.e. padding with any other pixel value works well for our software (Fig. 23 f and g). This zero pixel value problem has been observed for other application in computer vision as well[108]. Therefore, our software can handle missing information quite well when applying a correct padding.

Another problem that could arise is overheating of the GPU. We ran most of the tests on a standard PC with an NVIDIA RTX 3080 graphics card. We observed that the results deteriorated after three consecutive runs at the latest, as can be seen in Fig. 24. Here we have registered an identical image pair 4 times in succession by our software and already the 3rd run fails, which we attribute to the GPU becoming overheated.

## Neural network pipeline

Mutual information (MI) assumes that one feature of one image says something about the feature of the other corresponding image[57]. Typically, the modalities are defined as images acquired with different microscopes and registering these images ideally called as multi-modal image registration[50,109]. Some materials lead to a brightness for a particular or common feature in one image and darker in the other image collected via different microscope. Here, MI is a useful criterion for aligning the images through the mapping of feature extraction and is the basis for our proposed image registration software[58]. It uses one channel as moving and the other as fixed image. The moving image is deformed until the values of MI between the fixed and moving image is maximized. In the mathematical quantification, the MI between two variables is defined by determining the distance between a joint distribution and a complete independence by the Kullback-Leibler divergence:

$$MI(x,y) = \int PXY(x,y) log \frac{PXY(x,y)}{PX(x)PY(y)} dxdy,$$

where PXY is the joint density for random variables X and Y, PX and PY are marginal densities. Theses joint densities are calculated from the histograms of the two images. To make the computation optimized, we use the discretization of MI with neural networks and this is achieved by combining the duality of the Donsker-Varadhan with MI. This is particularly defined by Mutual Information Neural Estimation (MINE) and the optimized discrete values can be defined as[83,84]:

$$\max_{\theta} \sum_{l=1}^{L} MINE\left(F_l, Warp\left(M_l, M_{exp} \sum_{i=1}^{k} v_i H_i\right)\right) + MINE\left(F_1, Warp\left(M_1, M_{exp} \sum_{i=1}^{k} (v_i + v_i')H_i\right)\right),$$

where l is the starting length of Gaussian pyramids till the maximum levels (L), F and M represents fixed and moving image respectively, v denotes the vector values calculated via matrix exponential together with Transformation matrix (H), k is the number of sub-pixel feature extraction levels and θ represents the parameters for the neural network using AMSGrad (Adaptive Moment Estimation with a Strongly Non-Convex Decaying Learning Rate) with Adam for optimization[110].

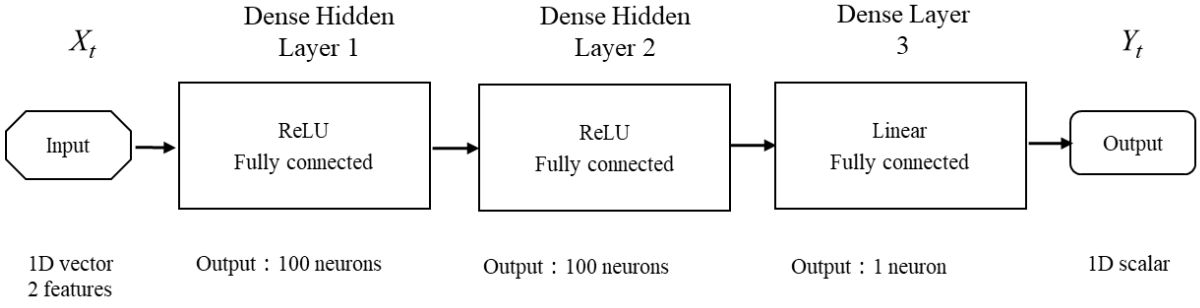


*Fig. 12: Architecture of our DNN model. Our DNN is 3-layer Multi-Layer Perceptron (MLP) model, the Input $X_t$ is a 1D vector of 2 features and passes input features directly to the 1st hidden layer. The first hidden layer is fully connected, containing 100 neurons and uses rectified linear unit (ReLU) activation function for nonlinearity. The output of first hidden layer is vector of 100 neurons and serves as the input to the 2nd hidden layer. This hidden layer is fully connected, and transforms the features learned by the 1st hidden layer and uses ReLU activation function with 100 neurons. The output of this layer is vector of 100 neurons and serves as the input to the output dense layer. The output layer has linear activation function and contains 1 neuron for predicting a single value or similarity score i.e. combined MI loss score for our model. The output $Y_t$ is a scalar value and reflects the MI loss score generated between the pixel locations by processing the sub-pixel pairs (from feature extraction) and finally with optimized processing, the network learns to estimate the accurate MI loss score between the entire image-pair.*

Our neural network is fully connected and has 2 hidden layers with rectified linear unit (ReLU) for non-linearity having 100 neurons and together work with homography parameters. The architecture and detailed view of model is represented in Fig. 12. The learning rate (lr) for

hyper-parameters is defined as $\alpha$ = 1e-5, $\beta$ = 5e-4, and $\gamma$ = 1e-5 and optimized number of iterations are 87,000 which take around 45 m (for a 1k x 1k Image-pair) to validate the hyper-parameters and provide the first correlated image from the image-pair. After that, there is no need for further validating the network and it takes around 5-10 s for registering the image pair together. All the steps are performed on Desktop-PC equipped with Nvidia GeForce RTX 3080 graphic card and an Intel(R) Core i9-11900K @3.50 GHz CPU. We observe that the total iteration time scales with the square root of the image pixel number.

This chapter 3 and next chapter 4 are adapted from a submitted manuscript and its supplementary information.

*"TFUDL-CLEM: A Training-Free Unsupervised Deep Learning Registration Method for Correlative Light and Electron Microscopy"*
D. Daksh, A. Kaltbeitzel, G. Glaßer, K. Landfester, I. Lieberwirth

Paper Contributions:

Daksh (first author) - conceptualization, methodology, coding and designing the neural network pipeline, setup different test cases, writing, preparing and editing of manuscript, data analysis and interpretation of all corresponding results, Anke Kaltbeitzel & Gunnar Glaßer - acquire the different sample images of LM and EM, Katharina Landfester & Ingo Lieberwirth - acquiring funding for the project, design and discussion of the concept, data analysis and interpretation of experimental results, and editing of the manuscript.

# Chapter 4: Testing & Challenges of Image Pre-processing

The following Chapter 4 is based on the submitted manuscript "TFUDL-CLEM: A Training-Free Unsupervised Deep Learning Registration Method for Correlative Light and Electron Microscopy". For the thesis, this chapter was extended with additional experiments and details. The verbatim use of the manuscript text and the corresponding supporting information is indicated by the use of the **Garamond** font.

## Assessing the effectiveness of unsupervised optimization of the DNN

Our approach to register LM and EM images is composed of two stages. In the first stage, the DNN is optimized with only one image pair. This is done without supervision, but solely on the basis of the MINE loss and the approach of the moving image. Fig. 13 D) - I) shows the development of the DNN learning. Already after 1000 iterations the registration is so good that visually one can hardly see an improvement with an increasing number of iterations. But on the other hand, one can also see that the MINE loss parameter continues to increase with increasing iterations (Fig. 14). Comparing the CLEM images, e.g. Fig. 13 b) - d), there is hardly any difference in the accuracy of the registration. However, the number of iterations cannot be increased indefinitely. Exceeding a certain threshold of approximately 87,000 iterations, the registration becomes drastically worse due to overfitting, which can be seen clearly in the image details (e.g. Fig. 13 e - f). This effect of overfitting is directly accompanied by a sharp increase in the MINE loss value, as can be seen impressively in Figure 8. For this reason we introduced an early stop at 87,000 iterations. After this step, the DNN is optimized to register CLEM image pairs. It is worth mentioning here that the choice of the first image pair is critical for the quality of the DNN's performance. The main objective for this first learning process is, to our experience, a good match of both image crops, i.e. the EM and the reflected LM image should display the same image area. This first training / optimization of the DNN is the most computational expensive process. It takes approximately 5 to 6 h on a standard PC equipped with an NVIDIA GPU.

After the DNN is trained with the initial image pair, further image pairs can be fed into the DNN for registration. These post-training image pairs just need several seconds to be processed, making our approach ideal for batch processing.

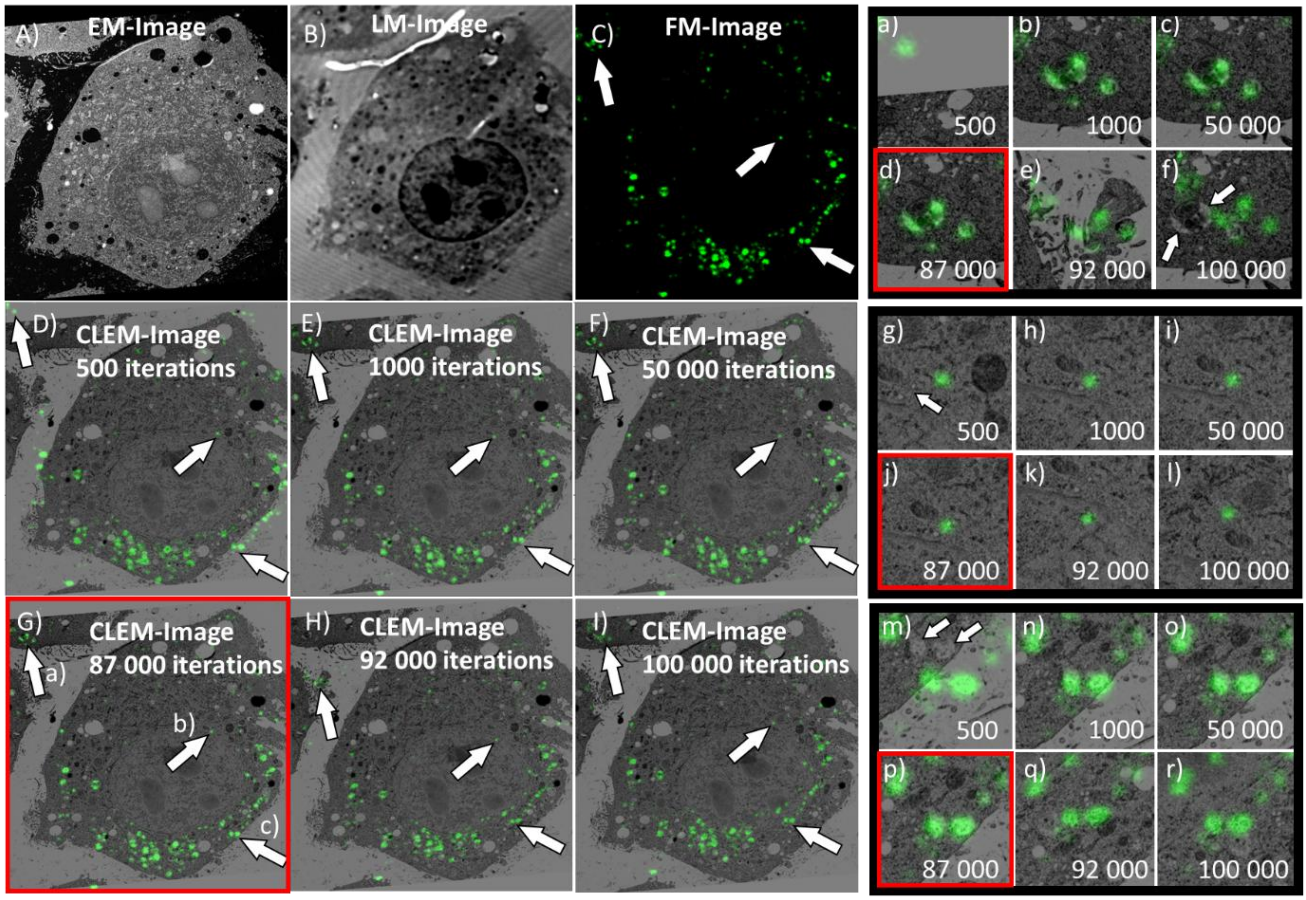


*Fig. 13: Mapping accuracy as a function of the number of iterations: A) to C) are the input images, the DNN was trained using the EM and the reflective channel of the LM image. After mapping, the EM image is overlayed with the transformed fluorescence microscopy image. D) to I) display the mapping results after training iterations as indicated in the images. The optimal number of iterations was empirical determined to be 87,000 (indicated by the red boxes). We have selected three recognizable features in the FM image as examples, marked by the white arrows and indexed them a) to c) in Figure G). a) to r) on the right side of the figure show the magnified sections of these areas. It is observed that already after 1000 iterations a first match with the target structures is achieved. Although no improvement in accuracy can be seen visually from the images, both the MSE and MS-SSIM values improve with the number of iterations. At 87,000 iterations, the system has optimized the superposition problem; further iterations then lead to a deterioration of the match again, as can be seen clearly in the enlarged sections.*

The number of iterations for the optimization is a crucial parameter. By simply checking the MINE loss parameter over the iteration number we could observe a sudden increase when exceeding 87000 iterations (Fig. 14). This is a clear indication of overfitting and therefore we introduce an early stop just before overfitting occurs.

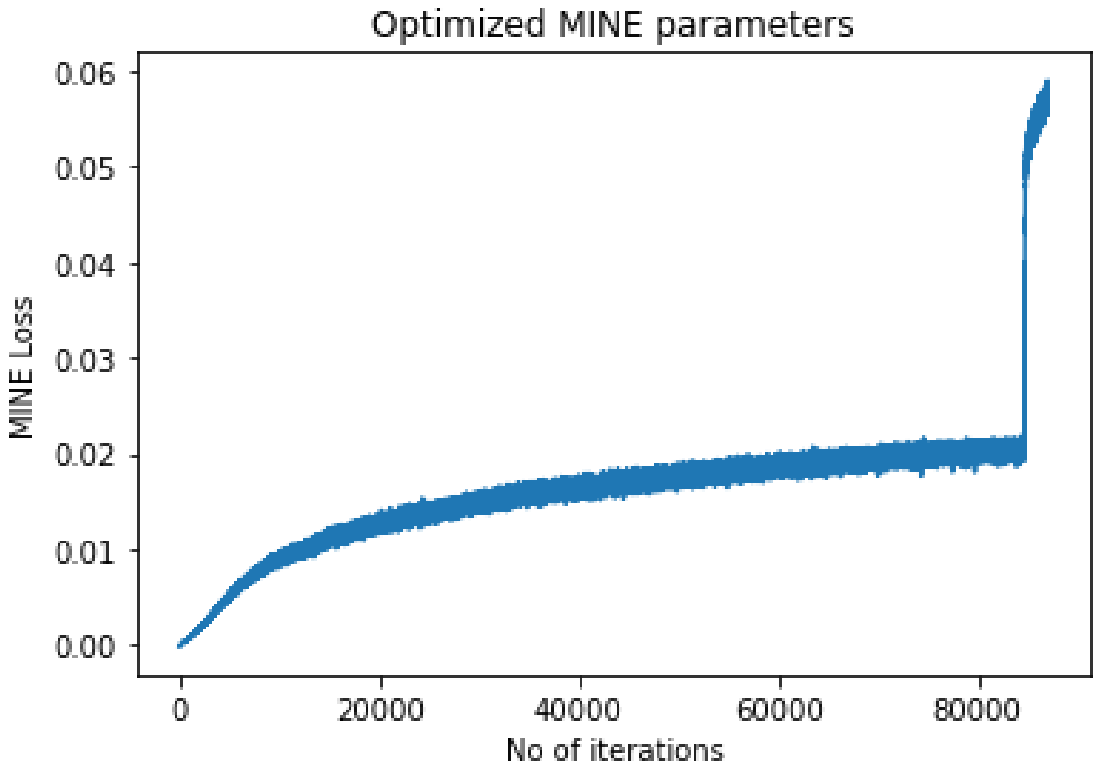


*Fig. 14: Diagram of the MINE loss as a function of the number of iterations. The onset of overfitting can be seen very clearly in the sudden increase of the value above 87,000 iterations. Therefore, we introduced an early stop at this number of iterations, since the matching becomes worse from here on, as can be seen well in Fig. 16.*

## Image pre-processing

For successful registration, the input images must fulfill a number of requirements. First of all, the images must have the same dimensions and pixel size. The latter is a very critical parameter that has to be thoroughly adjusted. Usually, microscopy images have a pixel calibration, which is provided by the microscope, based on a magnification calibration. However, we found during our tests, that these pixel calibrations are not necessarily consistent between two different microscopes. A mismatch in pixel size between the two input images will cause the registration process to fail (Fig. 16).

The image pre-processing routine is shown in Fig. 15, using the example of the data set from Fig. 9. Because the considerably lower resolution of the LM compared to the EM, the LM images usually have a larger FoV and pixel size. Both images already come with a calibrated pixel size from the respective microscopes. However, we found that these pixel size calibrations are not reliable and preferred to re-calibrate the image pairs. For this, we only adjust the LM image; the EM image was not altered at all. The first step in the processing routine is to manually identify common landmarks in both images and measure their distance in pixels (red line on the left images in Fig. 16). Usually, more than one distance is measured

in order to minimize inaccuracies of the manual measurement. The average ratio of the landmark distances in the EM and LM images finally yields an upscaling factor, which is used to inflate the LM image. Now, the pixel size of the inflated LM image is identical to the EM image.

In the next step, any rotation between the two images can be compensated for. However, this is not absolutely necessary as the algorithm can compensate for up to 10° rotation. Finally, the EM FoV is cropped out from the LM image with the same image dimensions as the EM image. This gives us two images of the same dimension and pixel size, which we can then transfer to our software for registration.

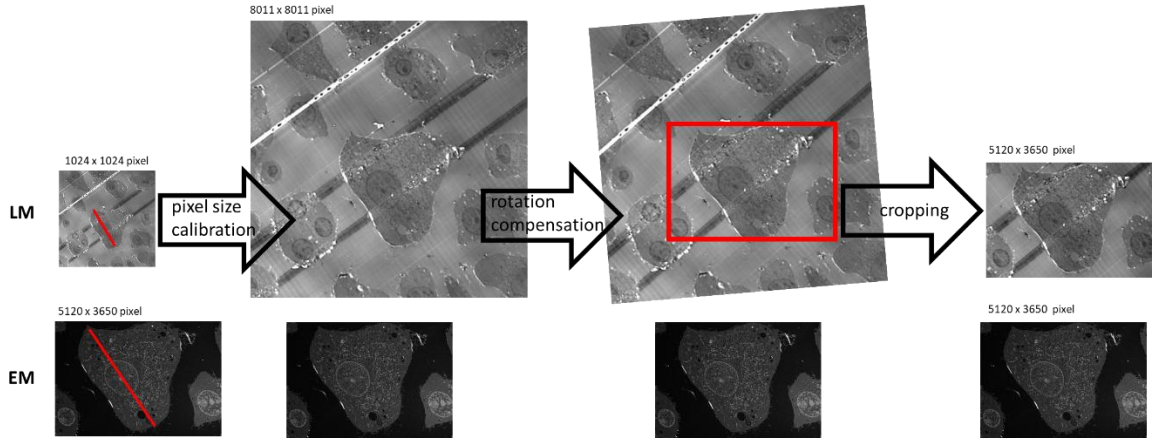


*Fig. 15: Image pre-processing: The raw images coming from the microscopes need to be preprocessed prior to being transferred to my software. In the first step, the pixel size of the LM image has to be adjusted to the EM image. This is done by measuring the distance of common landmarks, visible in both images (red line). From the ratio an upscaling factor is calculated and applied to the LM image. Here, the measured upscaling factor is 7.82. This inflates the raw LM image to a dimension of 8011 x 8011 pixels. In the next step, any rotational difference between the EM and the LM image can be compensated for. However, this step is optional. Finally, the FoV is cropped from the LM image, yielding two images containing the same FoV and the same pixel dimensions as the EM image.*

As already mentioned above, the pixel size calibration is a crucial step in the image pre-processing. Usually, the raw microscopy images already come with the calibrated pixel size. This calibration of the microscope is done once using magnification calibration standards like e.g. cross-grating patterns for TEM, patterned Silicon wafers for SEM and diamond-ruled micrometers for optical microscopy. However, these calibrations should not be relied upon, as demonstrated in Fig. 10. The problem stems from the different microscope resolutions, especially when combining light and electron microscopic data.

The two images shown in Fig. 16 a and b are the raw image files, the pixel calibration from the microscope is 81.62 nm/pixel and 9.92 nm/pixel for the LM and the EM image, respectively. This yields an upscaling factor of 8.22. However, when using this factor, as given by the microscope's calibration, our software fails to register both images, as shown in Fig. 16 c. Closer inspection by manually measuring common landmarks, e.g. the one indicated by the red

line, I measure a distance of 473 and 3692 pixels for the LM and EM images, respectively. This yields an upscaling factor of 7.81 and finally a successful registration (Fig. 9 e).

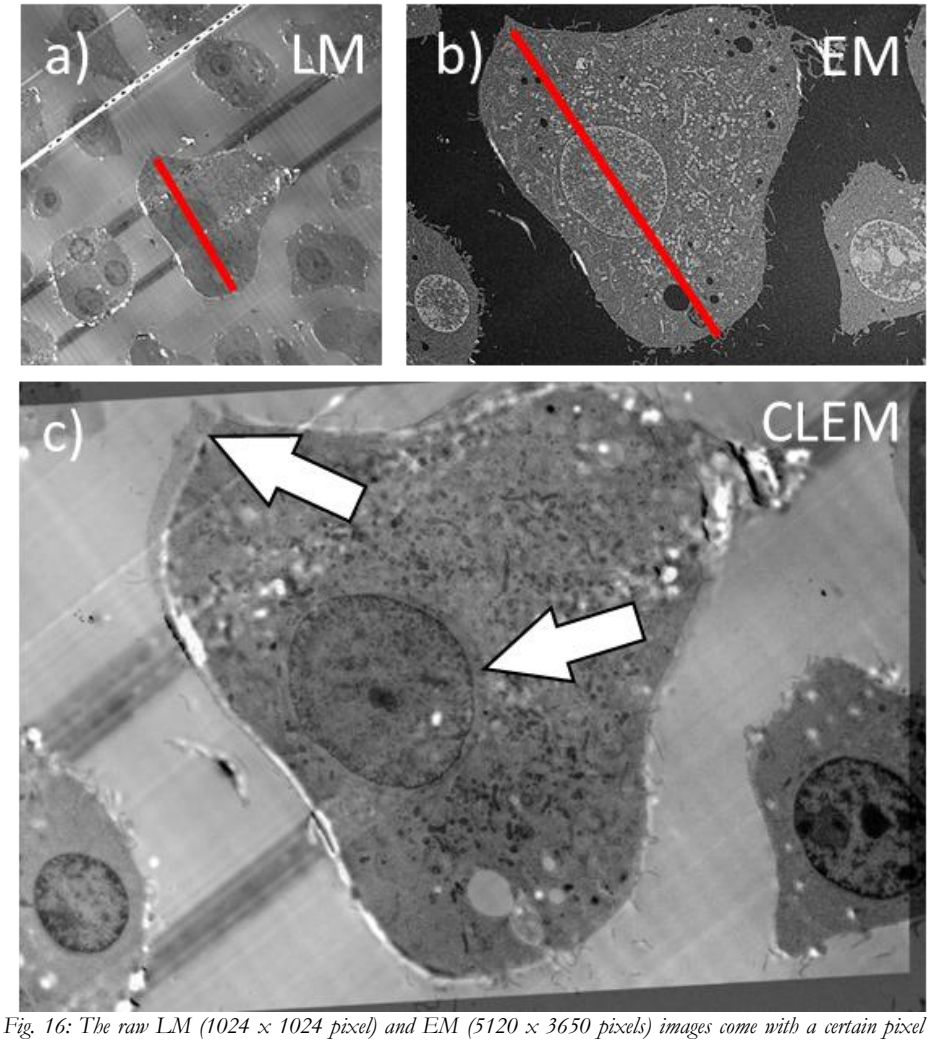


*Fig. 16: The raw LM (1024 x 1024 pixel) and EM (5120 x 3650 pixels) images come with a certain pixel size calibration, done by the microscope software. The original pixel size is 81.63 nm/pixel and 9.92 nm/pixel for the LM and EM image, respectively. When using these calibrations, the upscaling factor for the LM image is 8.22. c) When using this value for image pre-processing, our software fails to register the two images, as can be clearly seen from the overlay of the reflected channel with the EM image, indicated by the arrows.*

## Maximum image dimensions and importance of the pixel size calibration

The maximum dimension of the input images is determined by the GPU memory. On a common NVIDIA RTX4090 or RTX3080, we can process images up to 5120 x 3650 pixels and each optimization process requires approx. 3.5 h computation time (Table 6). When using a NVIDIA H100 GPU, even larger images can be processed. This is of course of particular interest as the development of electron microscopes in particular is moving towards ever larger image dimensions. This has the advantage that an image contains both the overall environment as well as the details of interest. Hence, I tested the registration of an 11k x 11k pixel image, as shown in Fig. 17. The initial EM image is a montage image with image dimension of 41722

x 40981 pixels and the LM image has an image dimension of 2048 x 2048 pixels. However, I had to downscale the EM image to a 10k x 10k image with an additional padding of 500 pixels to each side. Larger image dimensions resulted in an out-of-memory error even on the NVIDIA H100. However, after thorough manual calibration of the pixel size, I found that the registration of the two images was unsatisfactory. The mismatch increased from the upper left to the lower right corner of the image. This can already be seen in the enlarged images in Fig. 17 e to h. The further away from the upper left corner of the image, the greater the mismatch. I measured a total of 33 distinctive landmarks and displayed the displacement vector in Fig. 17 i. For better recognizability, the displacement vector is shown enlarged by a factor of 10. Altogether, the displacement results in an almost linear dependency. The greater the distance from the top left corner of the image (pixel count 0, 0), the greater the deviation in the registration (Fig. 17 j). The slope of the linear fit is approximately 0.01 and the intercept is -26 pixels. I suspect, that the pixel size calibration is the dominant factor here. The larger the image dimensions, the more precisely the pixel size must be calibrated. A deviation of only 0.01 in the pixel calibration between EM and LM image thus results in a deviation of 0.01 x 11k pixels = 110 pixels on a 11k x 11k pixel image. This is exactly the dependency that I recognize from Fig. 17 j.

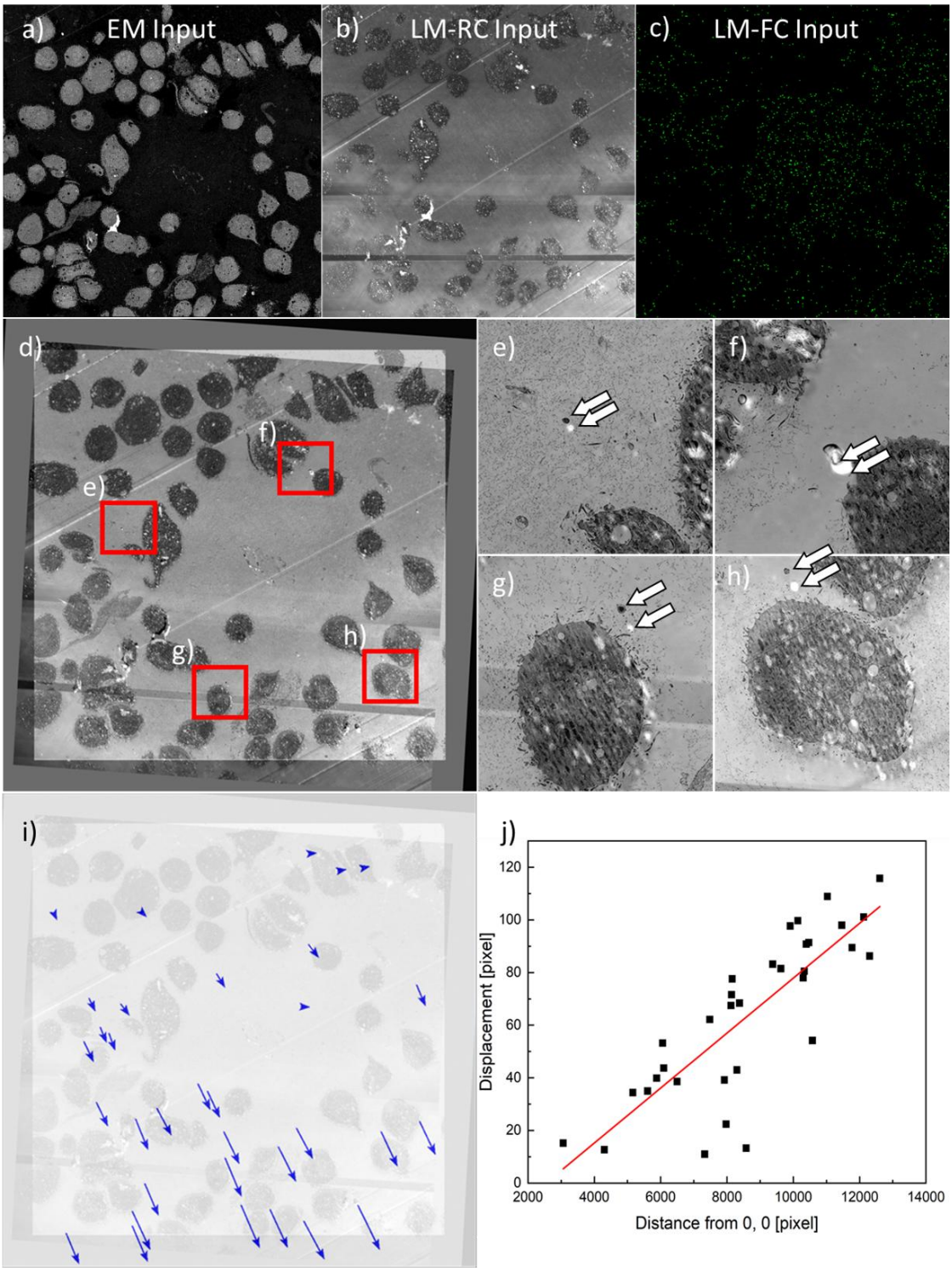


*Fig. 17: Registration result using the maximum dataset size with an 11k x 11k pixel EM image, processed on a NVIDIA H100 GPU. a) to c) input images and d) the overlay of the EM and the reflective LM channel after full size registration. In order to be able to recognize an offset in this large image format, e) - h) show enlargements of the marked areas in d) (red boxes). Here, a shift between EM and LM image is clearly recognizable (arrows), if one inspects common, distinctive landmarks in both images. We measured the displacement at a total of 33 distinguished landmarks and displayed them as blue arrows in i). The length of the arrows is stretched by a factor of 10 for reasons of recognizability. It can already be seen here, that there is hardly any offset near the origin of the image (the 0, 0 pixel position is in the top left corner) and that this offset increases with increasing distance. This observation is shown graphically in j). It shows the length of the displacement vector as a function of the distance to the origin of the image. The fitting (red line) yields a slope of approximately 0.01 with an intercept at the y-axis at -26 pixels.*

To overcome this mismatch problem with very large image dimensions, we take advantage of the calculated transformation matrix and used a somewhat indirect approach. Instead of

registering the full-sized image pair, we reduced the image dimensions (binning down) prior to the registration with our software, as demonstrated in the right panel of Fig. 9. Here, with a binning factor of approximately 10, image registration is successful. Fig. 18 shows the result for a reduction factor of approx. 20, with a 500 x 500 pixel image pair being processed to calculate the transformation matrix. Application of this transformation matrix to the full-sized 11k x 11k image pair, yields a very good matching of EM and LM image registration.

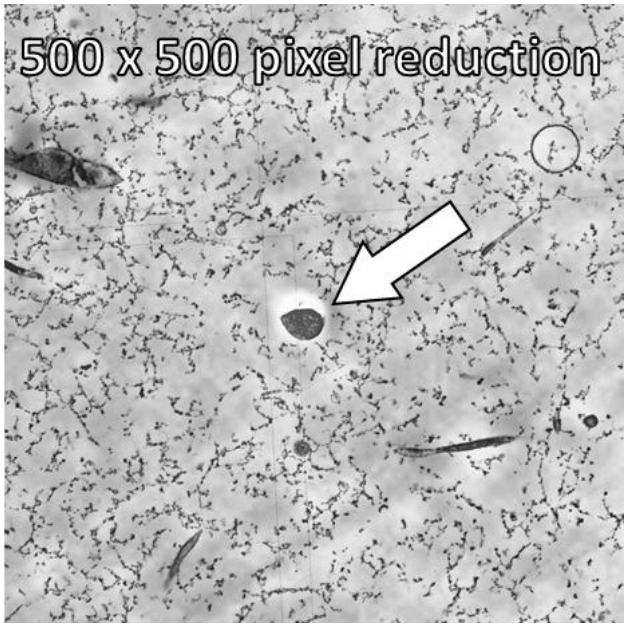


*Fig. 18: Effect of binning down full sized image prior to registration, showing the overlay of the reflective LM channel with the EM image. The image pair has been binned down to 500 x 500 pixels prior to registration. Using the calculated transformation matrix, the full sized images were superimposed. The image shows the same area as in Fig. 9 e.*

Although this approach is beneficial in terms of calculation time, there are limitations. Further reduction of the input image dimensions has to be taken with care. Since I use a 6-fold feature extraction in our software, which reduces the image dimensions by a binning factor of 2 at each level, image dimensions are reduced by a factor of $2^5 = 32$ for feature extraction. This means, a 250 x 250 pixel image will yield an 8 x 8 pixel image at feature extraction level 6. As can easily be seen in Fig. 19, this feature extraction does not contain any information (lower right panel in Fig. 19 with level 6 feature extraction from a 250 x 250 pixel image). Accordingly, our software seems to have a minimum image dimension, which might be around 500 x 500 pixels, as there is still some information in the level 6 feature extraction image.

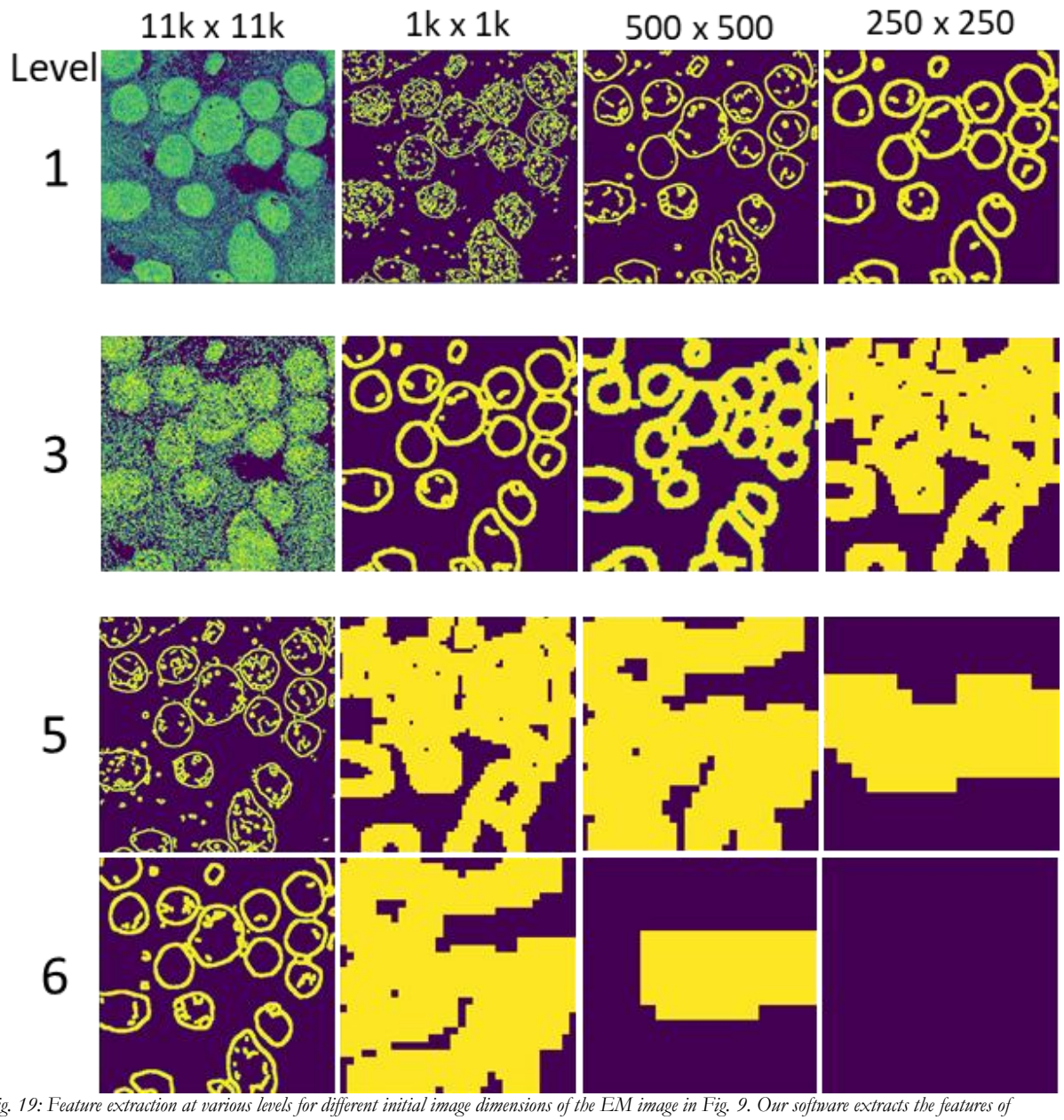


*Fig. 19: Feature extraction at various levels for different initial image dimensions of the EM image in Fig. 9. Our software extracts the features of the EM image using 6 levels in total, each level reducing the image dimensions by a factor of 2 x 2. The left column shows the feature extraction starting with the full sized 11k x 11k pixel EM image. At level 6, image dimensions were reduced to 340 x 340 pixels. On the left column, input image dimension is 250 x 250 pixels, resulting in a 7 x 7 pixel image at level 6 containing no information. Illustration of level 2 and 4 have been omitted here for reasons of clarity.*

## Shift and rotation tolerance

In order to test the tolerance against shift and rotation between the EM and LM input channel, I take advantage of the fact that the FoV of the LM channel is usually larger than that of the EM channel (Fig. 20 a, and b). The red box in Fig. 20 b displays the position and size of the EM channel information. To generate a shift, the LM image is now shifted by the corresponding pixels, in this case by + 250 pixels in the x-direction while retaining the position of the image section. This image section is then cut out and used for further processing. The

box marked in blue is cut out accordingly. For illustration purposes, however, the shift has been exaggerated. In this case, a shift of +250 pixels in x-direction, the registration fails, as clearly can be seen in Fig. 20 c.

The process to generate a rotation is similar. The LM image is rotated using image processing software (ImageJ) while the position of the cropping box is maintained, as illustrated by the green box.


*Fig. 20: Experimental approach to test shift and rotation tolerance: a) The EM image with image dimensions of 5120 x 3650 pixels has a smaller FoV than the LM image in b). After image Pre-processing and binning up of the LM image, the common FoV area is cropped out (red box). To generate a shift between both images, the LM image is shifted using image processing software but the cropping box is maintained (blue box). Thus, the shifting in positive x direction results in a left shift. The same procedure was applied for the generation of rotation (green box). c) The shift in +x direction of 250 pixels is not compensated by our software, as clearly visible from the overlay of the EM and the reflective LM channel.*

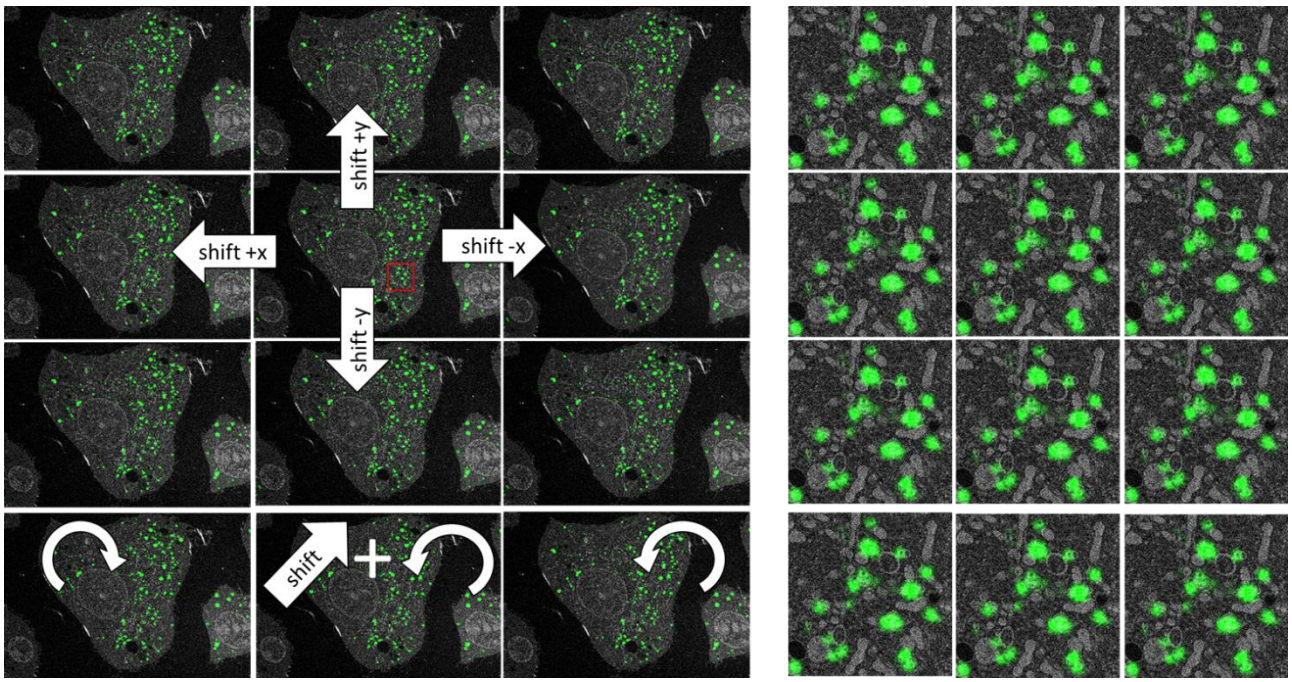


*Fig. 21: Tolerance against image shift and rotation. Left: same as Fig. 10 but with the fluorescence channel overlayed to the EM image. Right: Zoom in to the area marked by the red box in the central image.*

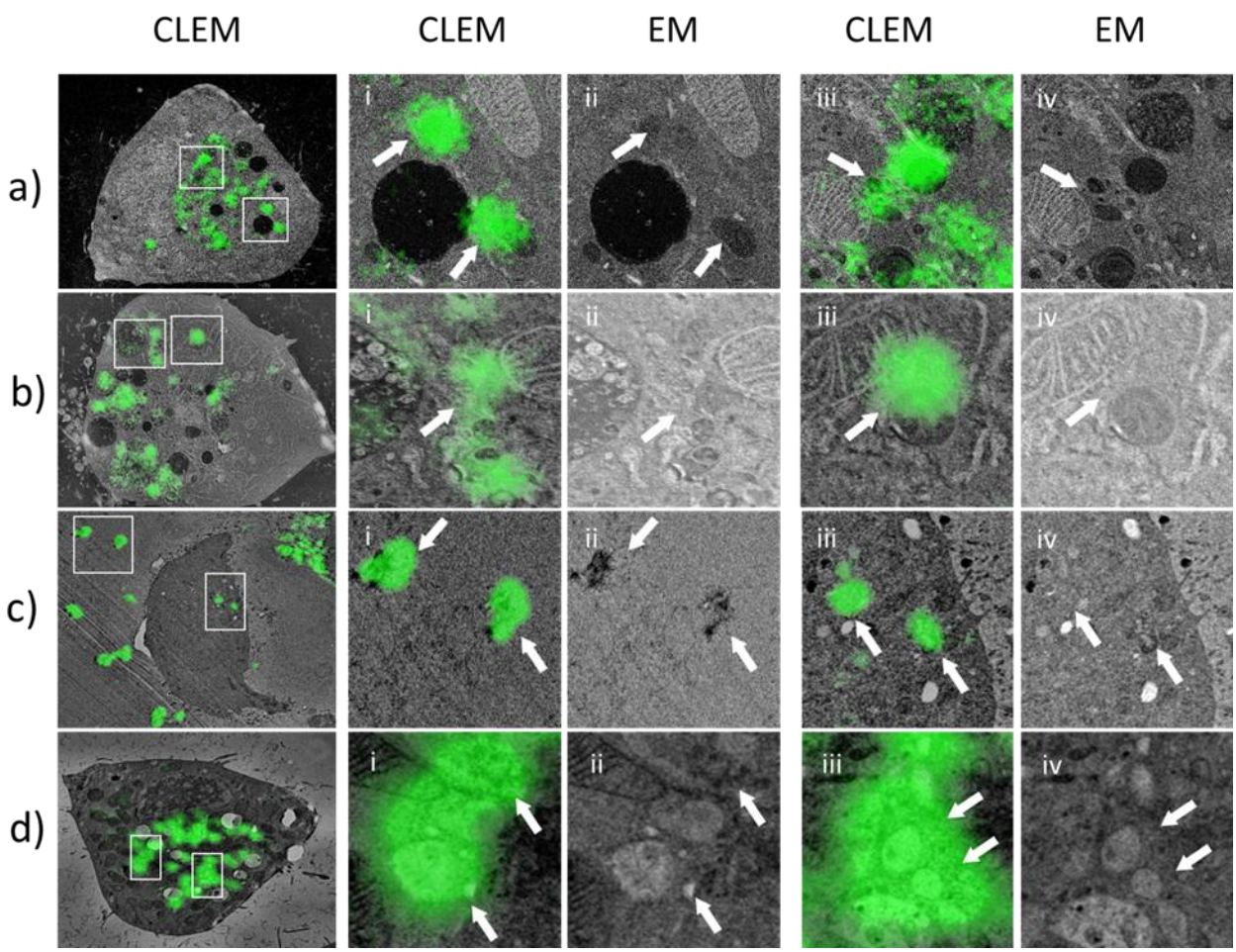


*Fig. 22: Detailed view of CLEM micrographs, which shows an enlarged view of the areas marked in the CLEM image (left column) compared to the same area of the EM input image. a) The nanogel particles are not recognizable in the EM image. However, the comparison of i) and ii) might suggest, that the features marked with the arrows in ii) represent the localization of the gel nanoparticles. However, when inspecting the area shown in iii), I can observe a fluorescence signal in a mitochondria, which is not expected for this experiment. b) no distinct features can be observed in the EM image at the localization of the FM signal in i) and ii). For the area shown in iii) however, it might be that the EM feature marked in iv) is the true localization of the fluorescent-labeled nanogel particles. If that´s the case, the CLEM registration at this position has a slight vertical offset. c) Using a TEM image as EM input data does not affect the performance of the DNN, which was optimized using a SEM image. The labeled objects are peptide nanofibres and they are not recognizable when being incorporated into a cell as shown in i) and ii). However, extracellular fibers /particles are clearly recognizable in the EM micrograph iv). The CLEM registration was successful, as shown in iii. d) Intracellular localization of nanogel particles (same as in a) and b)), using a TEM image as an EM input image. The fluorescently labeled nanogel particles do not show any distinct morphological feature in the TEM image, as indicated by the arrows.*

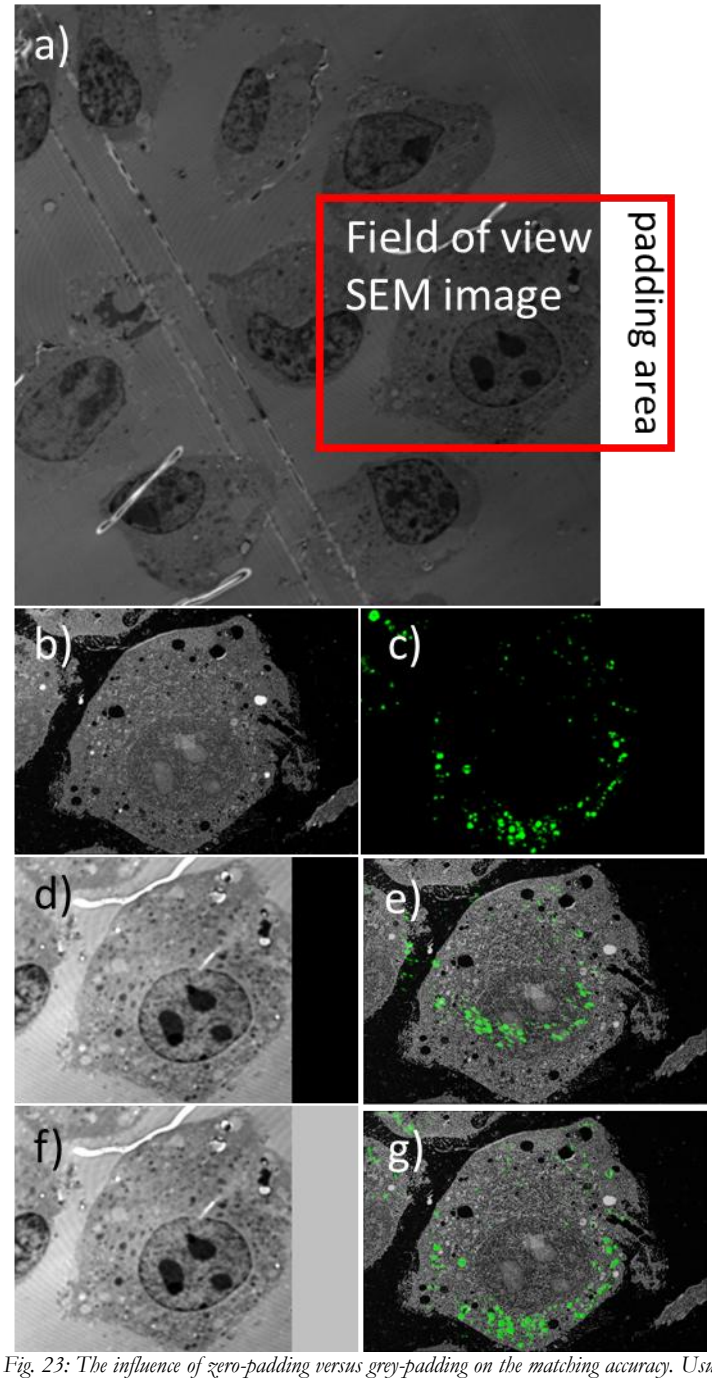


*Fig. 23: The influence of zero-padding versus grey-padding on the matching accuracy. Usually, the two input channels cannot be expected to match perfectly with respect to the Field of View. a) In this example, I show an LM image at low magnification, i.e. with a large Field of View. Here the reflected channel is shown. However, the corresponding SEM image b) is slightly shifted to the right over the Field of View of the LM image, as marked by the red box in a). However, since both input images must have the same pixel dimensions, the missing area must be filled in this case of the LM image. This is done by filling this missing area with black (0 value). This is referred to as zero-padding. For the FM channel of the LM image c) this is also possible without any problems, because this channel is not fed into the DNN. The reflected channel is used for registration, and if zero-padding is applied here d), the registration fails, as can be clearly seen in the CLEM image in e). However, if grey-padding is used for the reflected channel, i.e. if the missing area is filled with non-zero pixels, as shown in the reflected image in f), then the registration is successful, as shown in the CLEM image in g).*

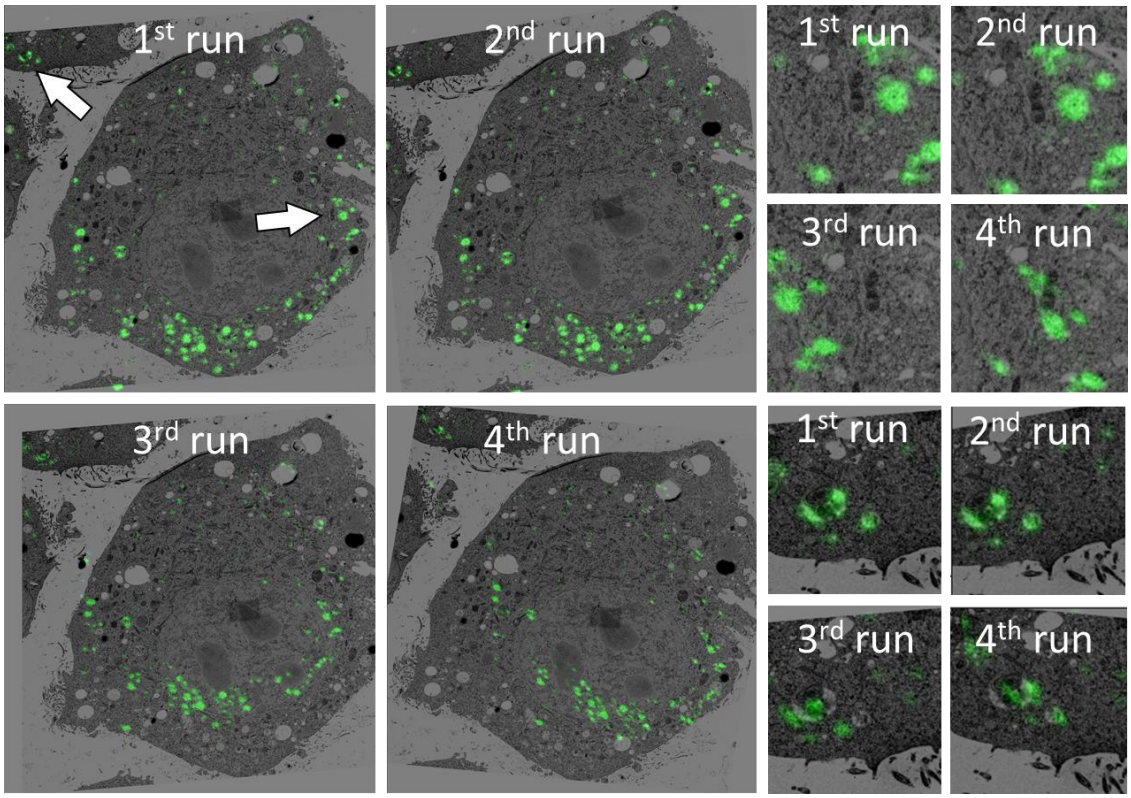


*Fig. 24: Effect of overheating the GPU demonstrated on the same image pair as in Fig. 8. The registration was done for 4 consecutive runs and already the 3rd run clearly fails, which has to be attributed to the overheating of the GPU.*

Table 6: List of image properties and GPU runtimes used for CLEM registration. The same CLEM image-pair run on different GPUs is indicated by * or **.

| Figure No | CLEM Image Dimensions [pixels] | Image Dimensions for GPU Computations [pixels] | GPU | Runtime [minutes] | MS-SSIM | MSE |
|---|---|---|---|---|---|---|
| 8 a) | *2194 x 2292* | *2194 x 2292* | *H100* | *50* | *0.9928* | *93.1* |
| 8 a)* | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.9928* | *93.1* |
| 9 a) | *10116 x 10080* | *10116 x 10080* | *H100* | *150* | *0.99569* | *39.3119* |
| 9 b)* | *10116 x 10080* | *1011 x 1008* | *RTX 3080* | *46* | *0.9951* | *32.7654* |
| 9 b)** | *10116 x 10080* | *1011 x 1008* | *H100* | *30* | *0.9951* | *32.7654* |
| 10 a)* | *5120 x 3560* | *5120 x 3560* | *RTX 3080* | *187* | *0.95613* | *2029.1* |
| 10 a)** | *5120 x 3560* | *5120 x 3560* | *H100* | *80* | *0.95613* | *2029.1* |
| 10 a) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9477* | *2775.8286* |
| 10 b) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9477* | *2834.6706* |
| 10 c) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.94299* | *3391.3574* |
| 10 d) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.95065* | *2327.2997* |
| 10 e) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.95613* | *2029.1* |
| 10 f) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.95095* | *2586.21* |
| 10 g) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9458* | *2770.265* |
| 10 h) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.95562* | *2145.3636* |
| 10 i) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9479* | *2634.626* |
| 10 j) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9448* | *3232.144* |
| 10 k) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9439* | *3375.2673* |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 l) | *5120 x 3560* | *1280 x 890* | *RTX 3080* | *45* | *0.9471* | *3232.871* |
| 11 a) | *1916 x 1190* | *1916 x 1190* | *RTX 3080* | *130* | *0.9959* | *70.9488* |
| 11 b) | *1017 x 1024* | *1017 x 1024* | *RTX 3080* | *45* | *0.9792* | *853.552* |
| 11 c) | *9992 x 9992* | *1011 x 1008* | *RTX 3080* | *45* | *0.9854* | *189.2263* |
| 11 d) | *5112 x 3558* | *1278 x 889* | *RTX 3080* | *45* | *0.97637* | *1153.2997* |
| 13 | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.9928* | *93.1* |
| 15 | *5120 x 3560* | *5120 x 3560* | *RTX 3080* | *187* | *0.95613* | *2029.1* |
| 16 | *5120 x 3560* | *5120 x 3560* | *RTX 3080* | *190* | *0.9779* | *836.6436* |
| 17 | *12000 x 12000* | *12000 x 12000* | *H100* | *160* | *0.8604* | *2660.895* |
| 17 | *10116 x 10080* | *1011 x 1008* | *H100* | *25* | *0.9951* | *32.7654* |
| 17* | *10116 x 10080* | *505 x 504* | *RTX 3080* | *30* | *0.9944* | *68.97* |
| 20 | *5120 x 3560* | *5120 x 3560* | *RTX 3080* | *187* | *0.95613* | *2029.1* |
| 22 a) | *4214 X 3552* | *1053 X 888* | *RTX 3080* | *45* | *0.9866* | *767.093* |
| 22 b) | *2560 X 1786* | *2560 X 1786* | *RTX 3080* | *84* | *0.9804* | *510.112* |
| 22 c) | *1024 X 1024* | *1024 X 1024* | *RTX 3080* | *44* | *0.9846* | *495.267* |
| 22 d) | *1017 x 1024* | *1017 x 1024* | *RTX 3080* | *44* | *0.9792* | *853.552* |
| 23 e) | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.98* | *516.548* |
| 23 g) | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.9836* | *521.41* |
| 24 a) | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.9928* | *93.1* |
| 24 b) | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.9928* | *93.1* |
| 24 c) | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.9964* | *141.641* |
| 24 d) | *2194 x 2292* | *2194 x 2292* | *RTX 3080* | *130* | *0.99634* | *125.31* |

Image registration using multi-modal image-pairs of different sources

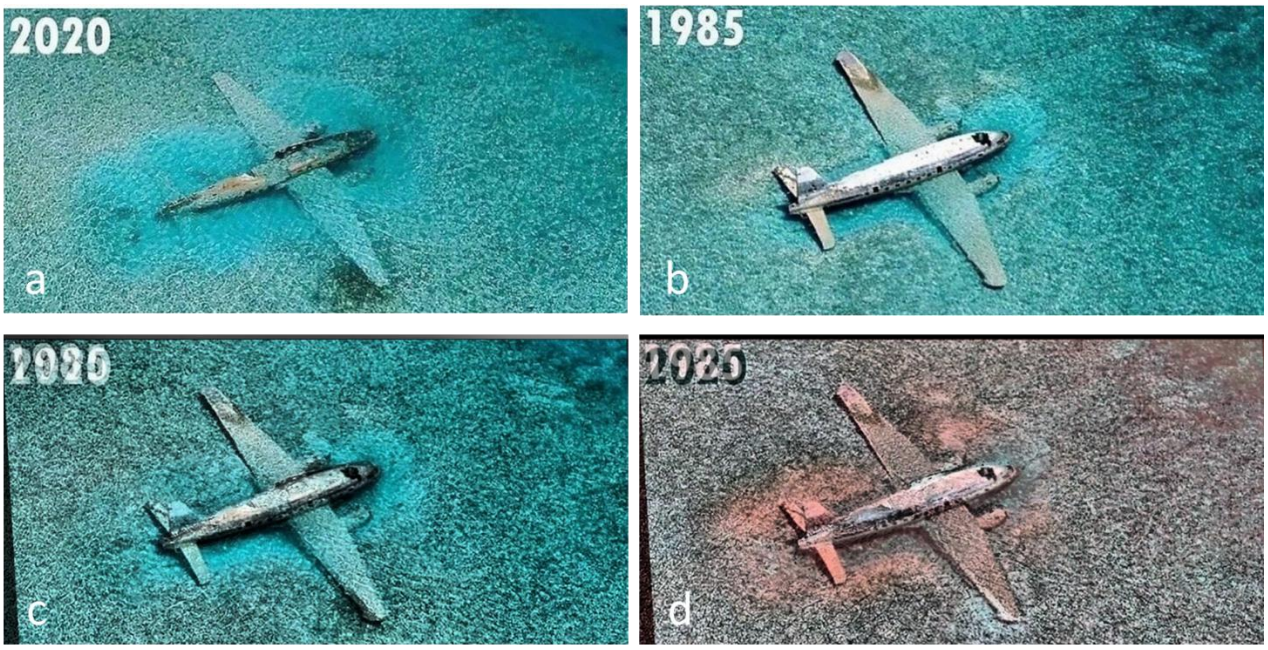


*Fig. 25: Image registration for the camera images of a crashed plane. a and b show photographs of Pablo Escobar's crashed plane near Norman's Cay in the Bahamas, captured in saltwater, with image dimensions of 700 × 347 pixels. These are used as input image pairs, where b serves as the fixed image and a as the moving image. c represents the registered image pair, and d highlights pixel-wise intensity variations, emphasizing structural discrepancies between corresponding regions of the two images. © Reprinted with permission [111]*

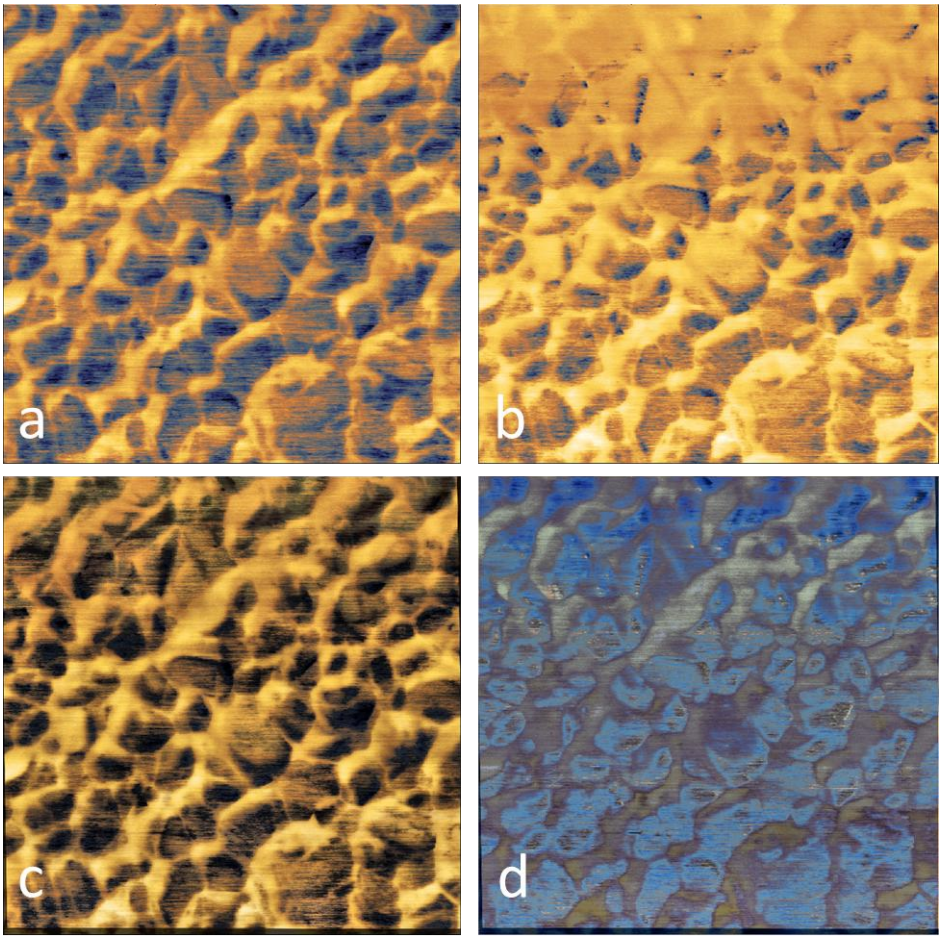


*Fig. 26: Image registration using atomic force microscopy (AFM) images. a and b are AFM images of the same FoV, showing height- and phase contrast modes, respectively. The image dimension is 2565 x 2568 pixels in both images. a and b are used as input image pairs, where a serves as the fixed image and b as the moving image. c represents the registered image pair, and d highlights pixel-wise intensity variations, emphasizing structural discrepancies between corresponding regions of the two images.*

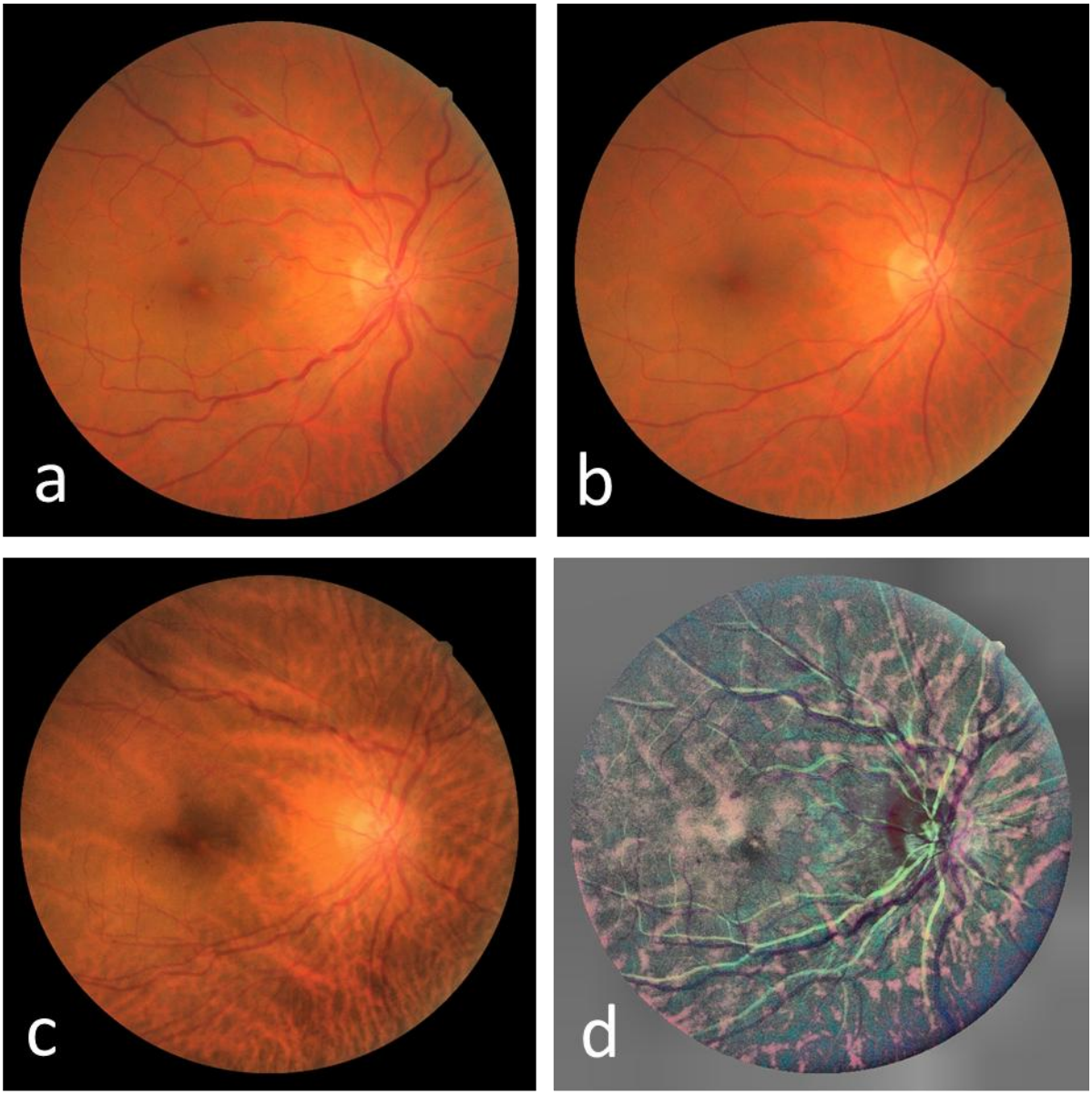


*Fig. 27: Image registration of retinal images. a and b are retinal images acquired with a Nidek AFC-210 fundus camera having image dimensions of 2912 x 2912 pixels and a FoV of 45° both in the x and y dimensions from Fundus Image Registration Dataset. a and b are used as input image pairs, where a serves as the fixed image and b as the moving image, c represents the registered image pair, and d highlights pixel-wise intensity variations, emphasizing structural discrepancies between corresponding regions of the two images*[112].

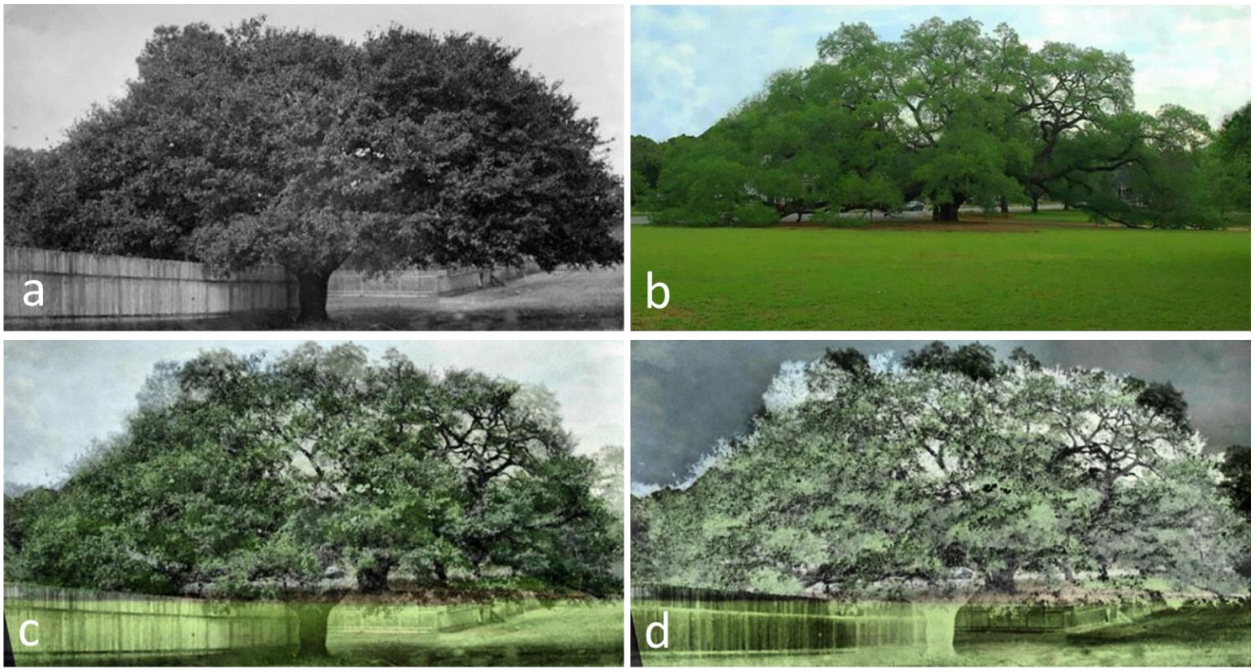


*Fig. 28: Image registration using common camera images of an oak tree from 1895 and 2020, respectively. a and b are used as input image pairs, where a serves as the fixed image and b as the moving image. c shows the result of the IR using my software. Although the FoV is not perfectly matching and the object – the oak tree – has significantly changed in morphology, the software is still able to yield a successful registration. d) When applying a subtractive overlay, the changes between the two input images can be seen even more clearly. © Reprinted with permission [111]*

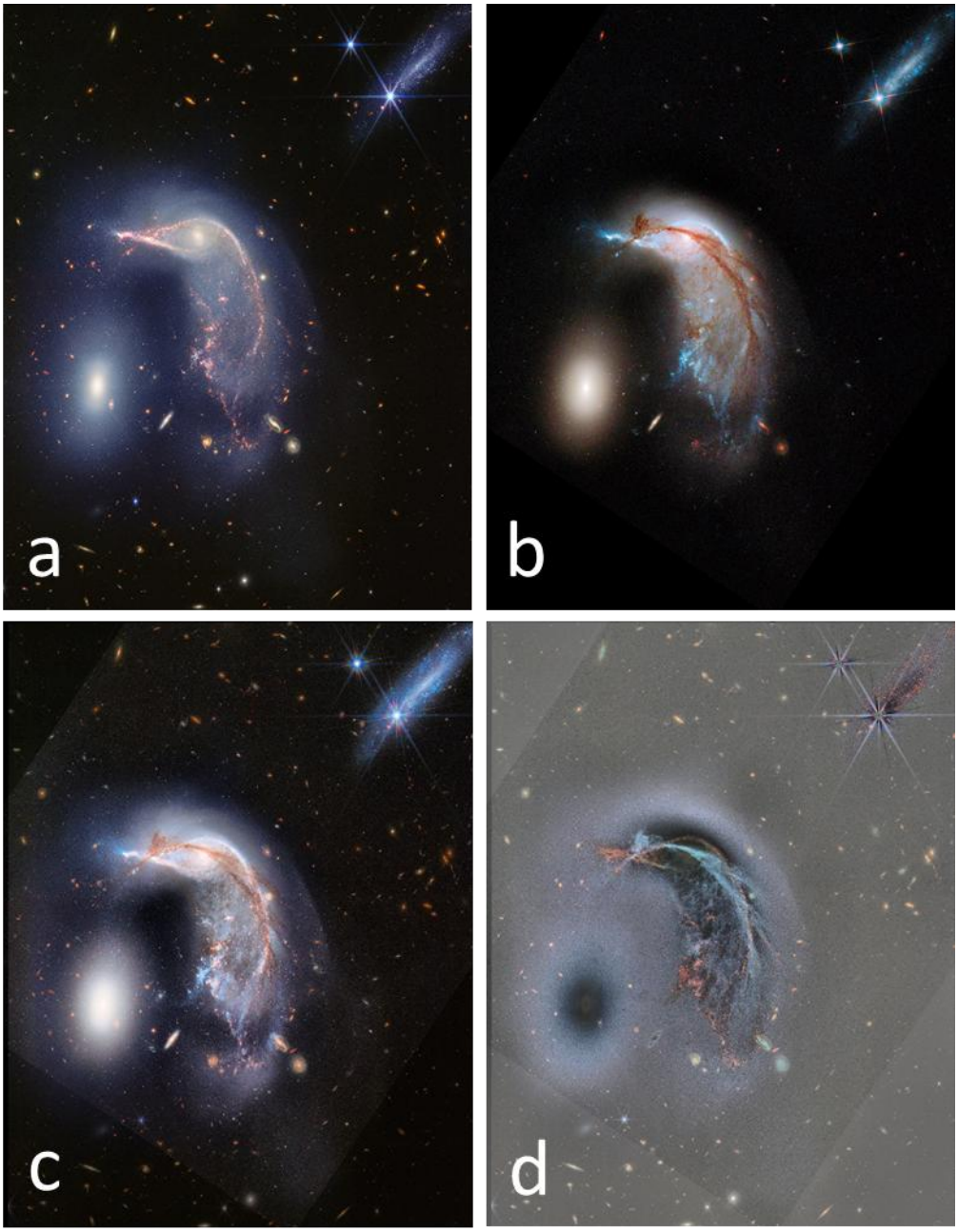


*Fig. 29: Image pair from interacting galaxies Arp 142 (Hubble and Webb Image) named "Two penguins and two eggs". a represents Webb's near-infrared image and b represents Hubble's visible light image. In Hubble's view, the Penguin is with a bright blue beak, and tail that is covered in an arc of bright brown dust while in Webb's near-infrared image shows the Penguin's shown in shades of pink. Its tail-like region is more diffuse, and a mix of lighter pinks as well as blues. Image dimension 3300 x 4260 pixels for both images. a and b are used as input image pairs, where a serves as the fixed image and b as the moving image. The IR result is shown in c and the subtractive overlay in d clearly shows the differences between the multimodal input images. © Reprinted with permission [113]*

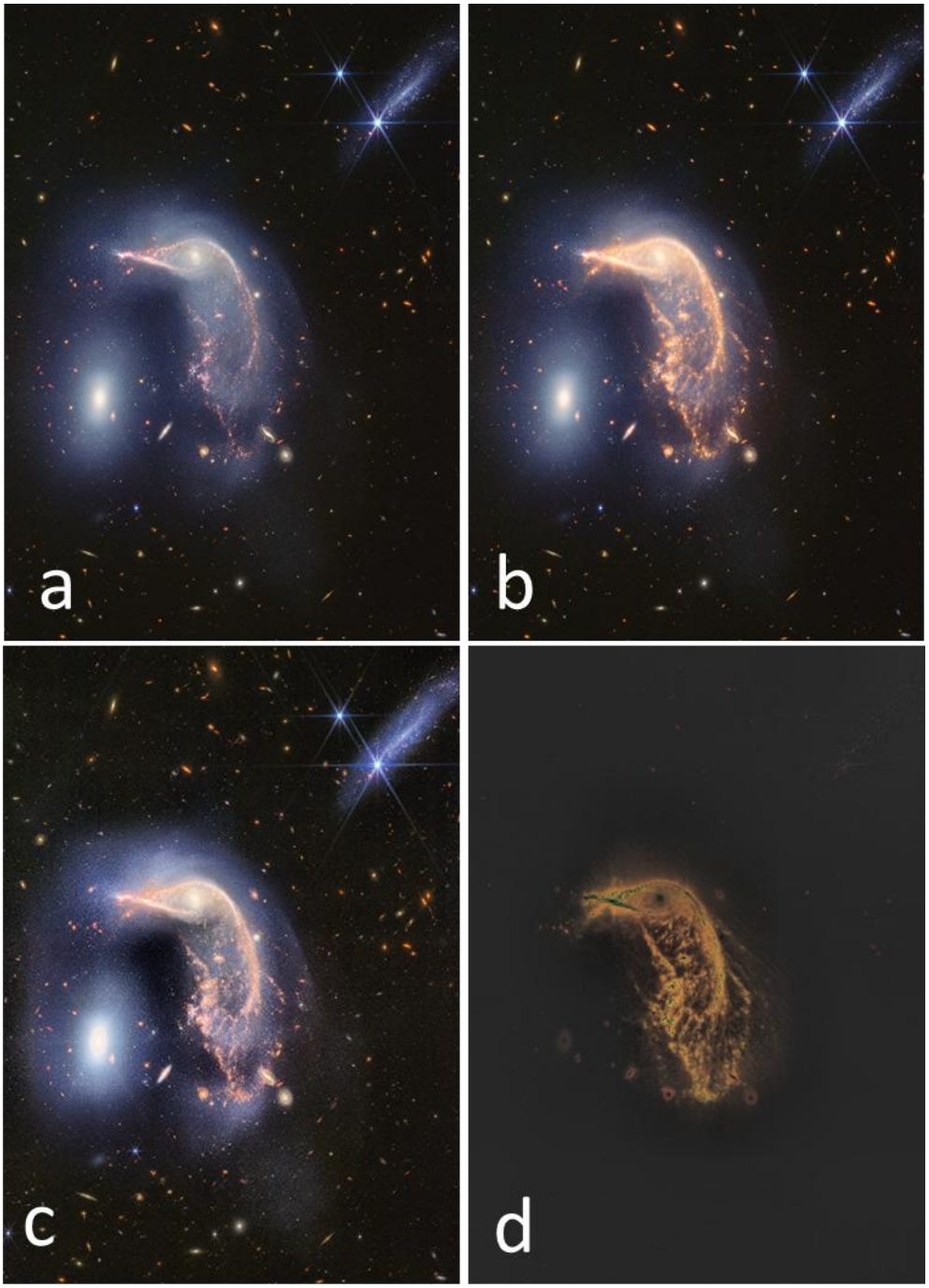

*Fig. 30: Same as in Fig. 29, but b shows the mid-infrared data from Webb telescope.*

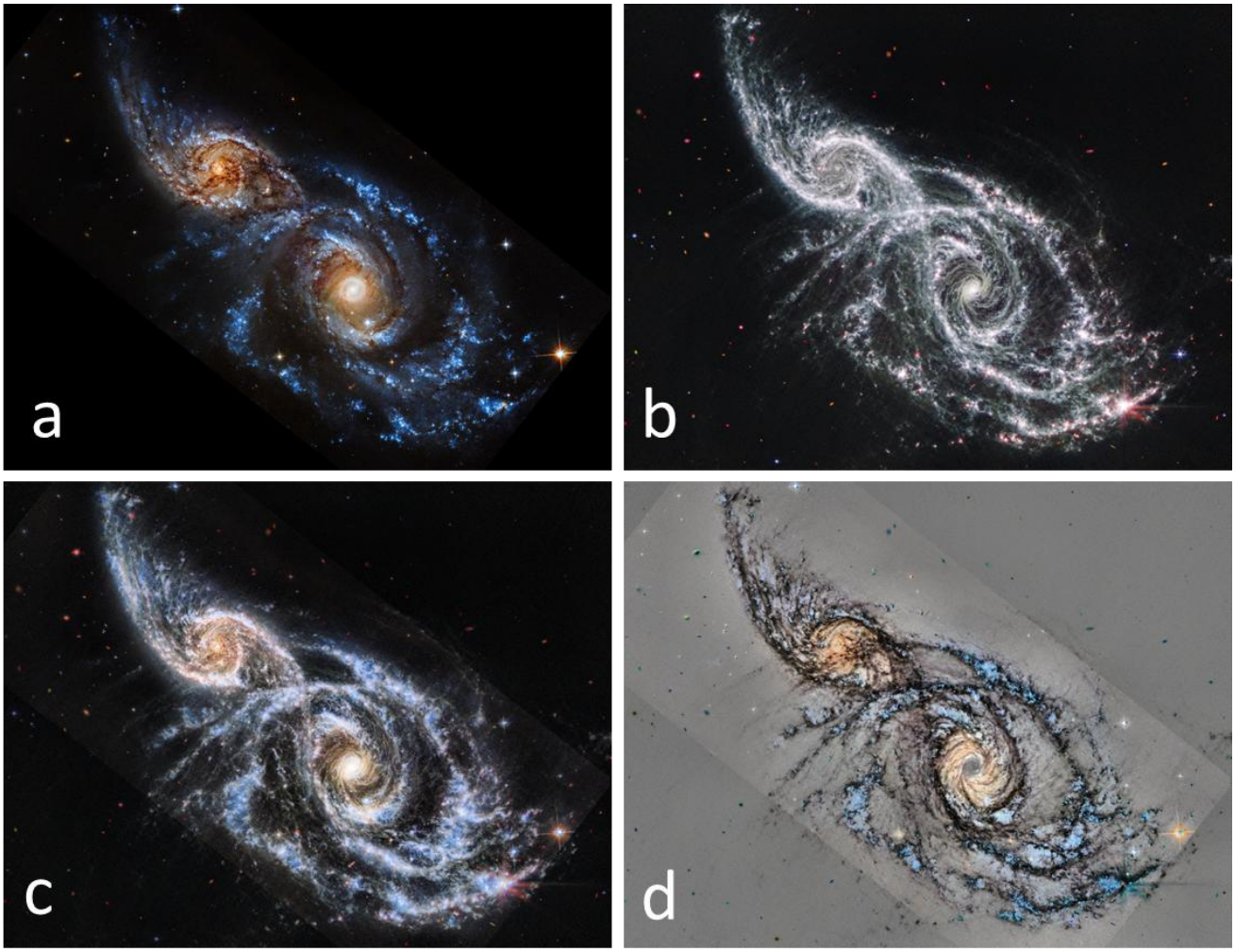


*Fig. 31: Two views of the same object are shown side by side, split evenly. Views of spiral galaxies IC 2163 and NGC 2207 captured by NASA's Hubble space telescope's ultraviolet- and visible-light view in a, and the James Webb space telescope's mid-infrared light view in b, spiral galaxies IC 2163 at top left, and NGC 2207 at bottom right in both images having image dimensions of 2532 x 1944 pixels. a and b are used as input image pairs, where a serves as the fixed image and b as the moving image, c represents the registered image pair, and d highlights pixel-wise intensity variations, emphasizing structural discrepancies between corresponding regions of the two images. © Reprinted with permission [113]*


*Fig. 32: Same as in Fig. 31, but b is the mid-infrared data from Webb telescope.*

# Source code for image registration algorithm (TFUDL-CLEM)

The algorithm is implemented in Jupyter Notebook using Anaconda navigator. Libraries are installed and updated using pip/conda. The source code is presented below in the cell format.

Step 1: Start with setting the environment and load the required libraries. It also checks whether cuda cores are using GPU or not, check output for cuda:0

```
from skimage import io, color
import numpy as np
import math
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.autograd import Variable
import torch.optim as optim
from skimage.transform import pyramid_gaussian
from skimage.filters import gaussian
from skimage.filters import threshold_otsu
from skimage.filters import sobel
from skimage.color import rgb2gray
from skimage import feature
from torch.autograd import Function
import cv2
from IPython.display import clear_output
import matplotlib.figure as fg
from skimage.io import imsave
from skimage.metrics import structural_similarity as ssim
from skimage import exposure as ex
from PIL import Image

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(device)
```

```
cuda:0
```

Step 2: Insert EM, Reflected and FM image-pair (EM - fixed image and LM - moving image)

```
I = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/Nanodiamonds/E1.tif").astype(np.float32) # fixed image
J = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/Nanodiamonds/R.tif").astype(np.float32) # ref_ image
FM = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/Nanodiamonds/F.tif").astype(np.float32) # FM image

fig=plt.figure()

plt.imshow(I/255)
plt.title("EM Image - fixed")
plt.show()
plt.imshow(J/255)
plt.title("Reflected Image - moving")
plt.show()
plt.imshow(FM/255)
plt.title("FM Image use for CLEM")
plt.show()
```

Step 3: Building up the image Gaussian pyramids for the image pair

```
downscale = 2.0
ifplot=True
if np.ndim(I) == 3:
    nChannel=I.shape[2]
    pyramid_I = tuple(pyramid_gaussian(gaussian(I, sigma=1, channel_axis=-1), downscale=downscale, channel_axis=-1))
    pyramid_J = tuple(pyramid_gaussian(gaussian(J, sigma=1, channel_axis=-1), downscale=downscale, channel_axis=-1))
elif np.ndim(I) == 2:
    nChannel=1
    pyramid_I = tuple(pyramid_gaussian(gaussian(I, sigma=1, channel_axis=-1), downscale=downscale, channel_axis=-1))
    pyramid_J = tuple(pyramid_gaussian(gaussian(J, sigma=1, channel_axis=-1), downscale=downscale, channel_axis=-1))
else:
    print("Unknown rank for an image")

%matplotlib inline
fig=plt.figure()
fig.add_subplot(1,2,1)
plt.imshow(pyramid_I[6]/255)
fig.add_subplot(1,2,2)
plt.imshow(pyramid_J[6]/255)
plt.show()
fig=plt.figure()
fig.add_subplot(1,2,1)
plt.imshow(pyramid_I[5]/255)
fig.add_subplot(1,2,2)
plt.imshow(pyramid_J[5]/255)
plt.show()
fig=plt.figure()
fig.add_subplot(1,2,1)
plt.imshow(pyramid_I[4]/255)
fig.add_subplot(1,2,2)
plt.imshow(pyramid_J[4]/255)
plt.show()
```

Step 4: Setting the class homography net for the neural network

```
class HomographyNet(nn.Module):
    def __init__(self):
        super(HomographyNet, self).__init__()
        # affine transform basis matrices

        self.B = torch.zeros(6,3,3).to(device)
        self.B[0,0,2] = 1.0
        self.B[1,1,2] = 1.0
        self.B[2,0,1] = 1.0
        self.B[3,1,0] = 1.0
        self.B[4,0,0], self.B[4,1,1] = 1.0, -1.0
        self.B[5,1,1], self.B[5,2,2] = -1.0, 1.0

        self.v1 = torch.nn.Parameter(torch.zeros(6,1,1).to(device), requires_grad=True)
        self.vL = torch.nn.Parameter(torch.zeros(6,1,1).to(device), requires_grad=True)

    def forward(self, s):
        C = torch.sum(self.B*self.vL,0)
        if s==0:
            C += torch.sum(self.B*self.v1,0)
        A = torch.eye(3).to(device)
        H = A
        for i in torch.arange(1,10):
            A = torch.mm(A/i,C)
            H = H + A
        return H
```

Step 5: Setting up model for mutual information neural estimation (MINE) neural network layers

```
n_neurons = 100
class MINE(nn.Module):
    def __init__(self):
        super(MINE, self).__init__()
        self.fc1 = nn.Linear(2*nChannel, n_neurons)
        self.fc2 = nn.Linear(n_neurons, n_neurons)
        self.fc3 = nn.Linear(n_neurons, 1)
        self.bsize = 1 # 1 may be sufficient

    def forward(self, x, ind):
        x = x.view(x.size()[0]*x.size()[1],x.size()[2])
        MI_lb=0.0
        for i in range(self.bsize):
            ind_perm = ind[torch.randperm(len(ind), device="cuda")]
            z1 = self.fc3(F.relu(self.fc2(F.relu(self.fc1(x[ind,:])))))
            z2 = self.fc3(F.relu(self.fc2(F.relu(self.fc1(torch.cat((x[ind,0:nChannel],x[ind_perm,nChannel:2*nChannel]),1))))))
            MI_lb += torch.mean(z1) - torch.log(torch.mean(torch.exp(z2)))

        return MI_lb/self.bsize
```

```
def AffineTransform(I, H, xv, yv):
    # apply affine transform
    xvt = (xv*H[0,0]+yv*H[0,1]+H[0,2])/(xv*H[2,0]+yv*H[2,1]+H[2,2])
    yvt = (xv*H[1,0]+yv*H[1,1]+H[1,2])/(xv*H[2,0]+yv*H[2,1]+H[2,2])
    J = F.grid_sample(I,torch.stack([xvt,yvt],2).unsqueeze(0), align_corners = True).squeeze()
    return J

def multi_resolution_loss():
    loss=0.0
    for s in np.arange(L-1,-1,-1):
        if nChannel>1:
            Jw_ = AffineTransform(J_lst[s].unsqueeze(0), homography_net(s), xy_lst[s][:,:,0], xy_lst[s][:,:,1]).squeeze()
            mi = mine_net(torch.cat([I_lst[s],Jw_],0).permute(1,2,0),ind_lst[s])
            loss = loss - (1./L)*mi
        else:
            Jw_ = AffineTransform(J_lst[s].unsqueeze(0).unsqueeze(0), homography_net(s), xy_lst[s][:,:,0], xy_lst[s][:,:,1]).squeeze()
            mi = mine_net(torch.stack([I_lst[s],Jw_],2),ind_lst[s])
            loss = loss - (1./L)*mi

    return loss
```

Step 6: Feature extraction and canny edge detection algorithm

```
L = 6

I_lst,J_lst,h_lst,w_lst,xy_lst,ind_lst=[],[],[],[],[],[]
for s in range(L):
    I_ = torch.tensor(cv2.normalize(pyramid_I[s].astype(np.float32), None, alpha=0, beta=1, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)).to(device)
    J_ = torch.tensor(cv2.normalize(pyramid_J[s].astype(np.float32), None, alpha=0, beta=1, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)).to(device)

    if nChannel>1:
        I_lst.append(I_.permute(2,0,1))
        J_lst.append(J_.permute(2,0,1))
        h_, w_ = I_lst[s].shape[1], I_lst[s].shape[2]

        edges_grayscale = cv2.dilate(cv2.Canny(cv2.GaussianBlur(rgb2gray(pyramid_I[s]),(21,21),0).astype(np.uint8), 0, 30),
                                np.ones((5,5),np.uint8),
                                iterations = 1)
        ind_ = torch.nonzero(torch.tensor(edges_grayscale).view(h_*w_)).squeeze().to(device)[:1000000]
        ind_lst.append(ind_)
    else:
        I_lst.append(I_)
        J_lst.append(J_)
        h_, w_ = I_lst[s].shape[0], I_lst[s].shape[1]

        edges_grayscale = cv2.dilate(cv2.Canny(cv2.GaussianBlur(rgb2gray(pyramid_I[s]),(21,21),0).astype(np.uint8), 0, 30),
                                np.ones((5,5),np.uint8),
                                iterations = 1)
        ind_ = torch.nonzero(torch.tensor(edges_grayscale).view(h_*w_)).squeeze().to(device)[:1000000]
        ind_lst.append(ind_)
    plt.imshow(edges_grayscale)
    plt.show()
    h_lst.append(h_)
    w_lst.append(w_)

    y_, x_ = torch.meshgrid([torch.arange(0,h_).float().to(device), torch.arange(0,w_).float().to(device)])
    y_, x_ = 2.0*y_/(h_-1) - 1.0, 2.0*x_/(w_-1) - 1.0
    xy_ = torch.stack([x_,y_],2)
    xy_lst.append(xy_)
```

Step 7: Optimizing MINE Loss along with homography hyperparameters

```
%%time
homography_net = HomographyNet().to(device)
mine_net = MINE().to(device)

optimizer = optim.Adam([{'params': mine_net.parameters(), 'lr': 1e-5},
                        {'params': homography_net.vL, 'lr': 5e-4},
                        {'params': homography_net.v1, 'lr': 1e-5}], amsgrad=True)
mi_list = []
for itr in range(87000):
    optimizer.zero_grad()
    loss = multi_resolution_loss()
    mi_list.append(-loss.item())
    loss.backward()
    optimizer.step()
    """
    clear_output(wait=True)
    plt.plot(mi_list)
    plt.title("Optimized MINE parameters")
    plt.xlabel("No of iterations")
    plt.ylabel("MINE Loss")
    plt.show()
    """
```

```
CPU times: total: 1h 25s
Wall time: 1h 34s
```

Step 8: Information for the transform matrix

```
#I = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset/E8_O.tif").astype(np.float32)
#FM = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset/F8_R.tif").astype(np.float32)
I_t = torch.tensor(I).to(device) # Original input images
J_t = torch.tensor(J).to(device) # Original input images
H = homography_net(0)
print(H)
print(H.grad_fn)
J_F = torch.tensor(FM).to(device) # without Gaussian FM chanel
if nChannel>1:
    J_w = AffineTransform(J_t.permute(2,0,1).unsqueeze(0), H, xy_lst[0][:,:,0], xy_lst[0][:,:,1]).squeeze().permute(1,2,0)
else:
    J_w = AffineTransform(J_t.unsqueeze(0).unsqueeze(0), H , xy_lst[0][:,:,0], xy_lst[0][:,:,1]).squeeze()
```

```
tensor([[ 0.9940, -0.0456, -0.0315],
        [ 0.0869,  0.9954,  0.0052],
        [ 0.0000,  0.0000,  1.0067]], device='cuda:0', grad_fn=<AddBackward0>)
<AddBackward0 object at 0x000001BFA33C82E0>
```

Step 9: Applying transform matrix to the fluorescence channel

```
#FM channel

if nChannel>1:
    J_1 = AffineTransform(J_F.permute(2,0,1).unsqueeze(0), H, xy_lst[0][:,:,0], xy_lst[0][:,:,1]).squeeze().permute(1,2,0)
else:
    J_1 = AffineTransform(J_F.unsqueeze(0).unsqueeze(0), H , xy_lst[0][:,:,0], xy_lst[0][:,:,1]).squeeze()
```

```
I_t = torch.tensor(I).to(device)    # Original input images
J_F = torch.tensor(FM).to(device)   # Original input images

print(J_F) # original FM image

print(J_F.grad_fn)
print(J_1) # FM image multiplied with Transformation matrix
print(J_F.grad_fn)

Tr_R = J_1 - J_F
print(Tr_R) # resultant transformation matrix
print(J_F.grad_fn)
```

Step 10: Saving the output CLEM images with fluorescence channel

```
I_t = torch.tensor(I).to(device) # Original input images
J_F = torch.tensor(FM).to(device) # Original input images

D = J_F + I_t
D_w = J_1 + I_t

D = (D - torch.min(D))/(torch.max(D) - torch.min(D))
D_w = (D_w - torch.min(D_w))/(torch.max(D_w) - torch.min(D_w))

fig=plt.figure(figsize=(15.00,15.00))
plt.imshow(D.cpu().data, interpolation='nearest')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_1.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()
fig=plt.figure(figsize=(15.00,15.00))
plt.imshow(D_w.cpu().data, interpolation='nearest')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_2.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()

eq1 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_1.tif")
eq2 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_2.tif")
eq_img1 = ex.equalize_adapthist(eq1)
fig=plt.figure(figsize=(15.00,15.00))
plt.imshow(eq_img1, interpolation='nearest')
plt.axis('off')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_1.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()

eq_img2 = ex.equalize_adapthist(eq2)
fig=plt.figure(figsize=(15.00,15.00))
plt.imshow(eq_img2, interpolation='nearest')
plt.axis('off')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_2.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()
```

```
# Assuming D and D_w are your tensors (after normalization)
# Bring D and D_w to CPU and convert to NumPy
D = D.cpu().detach().numpy()  # Remove gradient tracking and move to CPU
D_w = D_w.cpu().detach().numpy()

# Rescale the data back to [0, 255] for saving as an image (assuming the range is [0, 1] after normalization)
D = np.clip(D * 255, 0, 255).astype(np.uint8)
D_w = np.clip(D_w * 255, 0, 255).astype(np.uint8)


# Apply adaptive histogram equalization
D_equalized = ex.equalize_adapthist(D)
D_w_equalized = ex.equalize_adapthist(D_w)

# Convert the equalized images back to the range [0, 255] and to uint8
D_equalized = np.clip(D_equalized * 255, 0, 255).astype(np.uint8)
D_w_equalized = np.clip(D_w_equalized * 255, 0, 255).astype(np.uint8)

# Convert the equalized NumPy arrays back to PIL Images
D_equalized_image = Image.fromarray(D_equalized)
D_w_equalized_image = Image.fromarray(D_w_equalized)

# Save the equalized images
D_equalized_image.save("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/output_image_CL_n_1.tif")
D_w_equalized_image.save("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/output_image_CL_n_2.tif")
```

Step 11: Calculating the MSE and MS-SSIM values of the CLEM images with fluorescence channel

```
def mse(imageA, imageB):
    # the 'Mean Squared Error' between the two images is the
    # sum of the squared difference between the two images;
    # NOTE: the two images must have the same dimension
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
    err /= float(imageA.shape[0] * imageA.shape[1])

    # return the MSE, the lower the error, the more "similar"
    # the two images are
    return err

def compare_images(imagea, imageb):
    # compute the mean squared error and structural similarity
    # index for the images
    m = mse(imagea, imageb)
    s = ssim(imagea, imageb)

    # setup the figure
    %matplotlib inline
    fig = plt.figure(figsize=(10,10))
    #plt.suptitle("SSIM: %.2f MSE: %.2f" % (s, m))
    #print(s[0])
    #print("SSIM: %f" % (s[0]))
    #print("MSE: %f" % (m))
    print("SSIM: %f MSE: %f" % (s, m))
    fig.add_subplot(1,2,1)
    plt.imshow(image1)
    fig.add_subplot(1, 2, 2)
    plt.imshow(image2)



image1 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_1.tif") # first image
image2 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/CL_n_2.tif") # CLEM image

compare_images(image1, image2)
```

Step 12: From here, the following source code provides the information for the reflected channel & for saving the output CLEM images with the reflected channel

```
I_t = torch.tensor(I).to(device) # Original input images
J_t = torch.tensor(J).to(device) # Original input images

D = J_t - I_t
D_w = J_w - I_t

D = (D - torch.min(D))/(torch.max(D) - torch.min(D))
D_w = (D_w - torch.min(D_w))/(torch.max(D_w) - torch.min(D_w))
%matplotlib inline
fig=plt.figure(figsize=(15,15))
plt.imshow(D.cpu().data, interpolation='nearest')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_1.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()
fig=plt.figure(figsize=(15,15))
plt.imshow(D_w.cpu().data, interpolation='nearest')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_2.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()

eq3 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_1.tif")
eq4 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_2.tif")
eq_img1 = ex.equalize_adapthist(eq3)
fig=plt.figure(figsize=(15.00,15.00))
plt.imshow(eq_img1, interpolation='nearest')
plt.axis('off')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_1.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()

eq_img2 = ex.equalize_adapthist(eq4)
fig=plt.figure(figsize=(15.00,15.00))
plt.imshow(eq_img2, interpolation='nearest')
plt.axis('off')
plt.savefig("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_2.tif", dpi='figure', bbox_inches='tight')  #give the path for saving
plt.show()
```

```
# Assuming D and D_w are your tensors (after normalization)
# Bring D and D_w to CPU and convert to NumPy
D = D.cpu().detach().numpy()  # Remove gradient tracking and move to CPU
D_w = D_w.cpu().detach().numpy()

# Rescale the data back to [0, 255] for saving as an image (assuming the range is [0, 1] after normalization)
D = np.clip(D * 255, 0, 255).astype(np.uint8)
D_w = np.clip(D_w * 255, 0, 255).astype(np.uint8)


# Apply adaptive histogram equalization
D_equalized = ex.equalize_adapthist(D)
D_w_equalized = ex.equalize_adapthist(D_w)

# Convert the equalized images back to the range [0, 255] and to uint8
D_equalized = np.clip(D_equalized * 255, 0, 255).astype(np.uint8)
D_w_equalized = np.clip(D_w_equalized * 255, 0, 255).astype(np.uint8)

# Convert the equalized NumPy arrays back to PIL Images
D_equalized_image = Image.fromarray(D_equalized)
D_w_equalized_image = Image.fromarray(D_w_equalized)

# Save the equalized images
D_equalized_image.save("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/output_image_RC_n_1.tif")
D_w_equalized_image.save("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/output_image_RC_n_2.tif")
```

Step 13: Calculating the MSE and MS-SSIM values of the CLEM image with reflected channel

```
def mse(imageA, imageB):
    # the 'Mean Squared Error' between the two images is the
    # sum of the squared difference between the two images;
    # NOTE: the two images must have the same dimension
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
    err /= float(imageA.shape[0] * imageA.shape[1])

    # return the MSE, the lower the error, the more "similar"
    # the two images are
    return err

def compare_images(imagea, imageb):
    # compute the mean squared error and structural similarity
    # index for the images
    m = mse(imagea, imageb)
    s = ssim(imagea, imageb)

    # setup the figure
    %matplotlib inline
    fig = plt.figure(figsize=(10,10))
    #plt.suptitle("SSIM: %.2f MSE: %.2f" % (s, m))
    #print("SSIM: %.2f MSE: %.2f" % (s, m))
    print("SSIM: %f MSE: %f" % (s, m))
    fig.add_subplot(1,2,1)
    plt.imshow(image1)
    fig.add_subplot(1, 2, 2)
    plt.imshow(image2)

image1 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_1.tif") # first image
image2 = io.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D7/Res/RC_n_2.tif") # CLEM image
compare_images(image1, image2)
```

# Chapter 5: Weakly supervised Image Pre-processing pipeline

The following chapter 5 is based on the manuscript in preparation "A Weakly Supervised Pre-processing Pipeline for Multi-Modal Image Pair Generation for Image Registration" submitted to Computer Vision and Image Understanding. For the thesis, this chapter was extended with additional details. This chapter explains the role of image pre-processing in automated CLEM and how to successfully generate the different modalities image-pairs having same pixel sizes. The verbatim use of the manuscript text is indicated by the use of the **Garamond** font.

## Image pre-processing and weakly supervised generated image-pairs

Template matching (TM) is the fundamental method in digital image processing for finding the similar or identical structure of a template image inside a large source image. This approach is useful for identifying or detecting the object, scene, motion, similarity or edge detection in the multiple images.

To address the fundamental challenges associated with raw image handling and to improve the overall imaging workflow, we propose a novel image preprocessing pipeline (Fig. 33). The pipeline is hardware-independent, easy to integrate into existing imaging systems, and designed to be broadly applicable.

In the first step, the different pixel size calibrations, which originate directly from the physical calibration of both microscopes as metadata, must be matched. However, the initial microscope calibrations can deviate considerably from each other. I) In order to obtain a precisely matching calibration of both images, we need a common feature in both images from which we can determine the pixel size ratio (PSR) between both images. II) The LM and EM images together with the PSR are the input for the following, unsupervised pipeline. III) In order to generate a suitable template from the EM input image, we reduce the EM image dimensions by binning down according to the calculated PSR value, while the LM image serves as the source image. IV) Subsequently, we apply the area based template matching to localize the template position within source image. V) Finally, the identified area is cropped from the LM source image and resized using the determined PSR value. Together with the raw EM image, this results in an image pair that shares the same field of view (FoV) and has a calibrated, matching pixel size-while preserving the original resolution of the EM image.

The core of this approach is based on TM, which has the unique ability to search for and locate a template image within a larger image[104,106]. We tested two template matching approaches to solve the multi-modal problem for generating an image pair, either using normalized cross-correlation method or normalized correlation coefficient method[104,105]. Here, we ultimately chose the latter method due to its superior performance compared to the cross-correlation approach, particularly in cases where the template and image regions exhibit differing lighting contrasts, as it effectively normalizes intensity variations. This process determines the spatial location of a specified pattern / template via a pixel-wise correlation between the source input image and a predefined template encapsulating the target structure. The template is translated by *u* discrete units along the x-axis and *v* units along the y-axis of the image, with similarity metrics computed over the template region at each spatial offset. The standard normalized cross correlation *R(u,v)* in template matching is mathematically represented as:

$R(u,v) = \frac{\sum\sum((I(x+u,y+v)-I')\,(T(x,y)-T'))}{\sqrt{\sum(I(x+u,y+v)-I')^2}.\sqrt{\sum(T(x,y)-T')^2}}$, where I´ and T´ is the mean region in the source and template image respectively.

As the multi-modal images are influenced by intensity conditions, the mean subtraction be avoided. This is the key strategy here that we are not going to compensate for intensity change, as it has to be determined through the low dimensional image. Therefore, the standard normalized cross correlation coefficient is modified as:

$$R'(u,v) = \frac{\sum\sum\big((I(x+u,y+v)\big)\,(T(x,y))}{\sqrt{\sum(I(x+u,y+v))^2}.\sqrt{\sum(T(x,y))^2}}$$

With the help of normalized correlation coefficient TM, we can match the template image within the source image. The obtained matching result provides the precise location of the region that has to be cropped out from the LM image.

This pre-processing pipeline can be implemented for generating any multi-modal image pair of identical features e.g. in medical, biological or satellite image analysis[6,54]. Moreover, it is truly independent of any image metadata parameters.

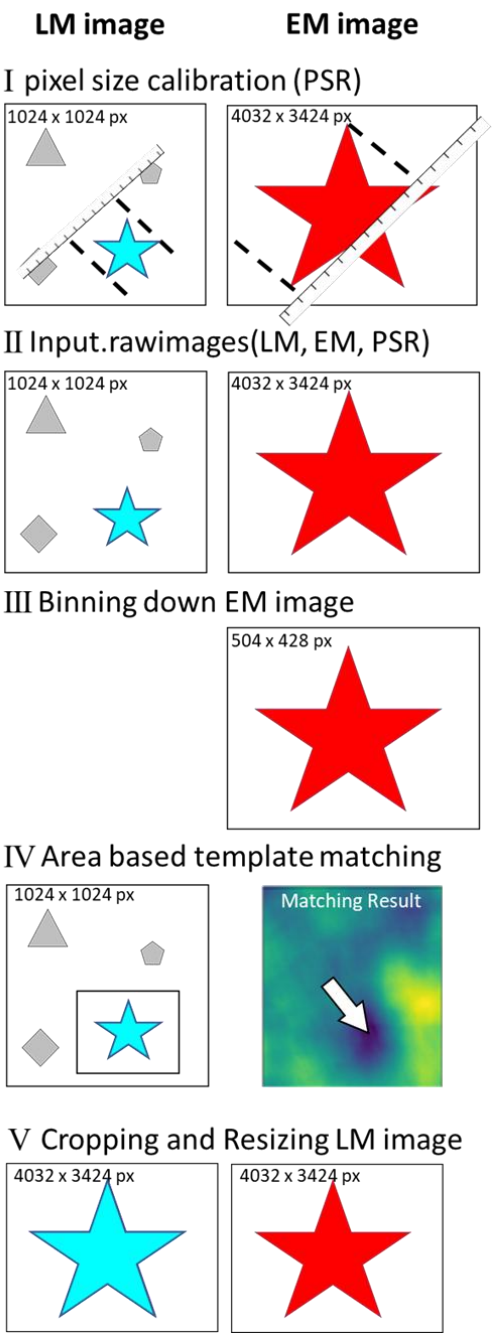


*Fig. 33: Graphical representation of the workflow of weakly supervised image pre-processing algorithm. I) Before applying the automated pre-processing algorithm, the exact pixel size calibration ratio (PSR) between the two input channels has to be determined manually by using internal, common landmarks. II) The input feed comprises of the LM and EM raw images of different image dimensions and the PSR in addition. Usually, the image dimension of the EM image is significantly larger than that of the LM image. III) The first step of the pre-processing algorithm is to bin down the EM raw image to a pixel dimension of approximately 0.5 times of the LM image dimension. IV) With this reduced EM image the area based template matching is applied using normalized cross-correlation. V) After successful matching, the respective area is cropped from the raw LM image and is finally resized to the EM image dimension using the calculated PSR.*

## Solutions to false matching problems and correct pixel size mapping

In Fig. 34 the initial situation for the two input images is displayed. In columns a and b of the initial input EM and LM images are shown. As can be seen, the FoV in the EM image is significantly smaller than the FoV of the LM image. However, the task of our presented workflow is to precisely locate and crop the EM image FoV from the LM image (the red box in Fig. 34). As for a successful application for subsequent machine learning registration approaches, the generated image pairs need to have an identical FoV and equal pixel dimensions. This requires a thorough determination of the pixel size ratio (PSR) between both image modalities. The PSR between the two microscopes is defined as the quotient of LM and EM pixel size. During our analyses, we found that the pixel size calibrations, as coming from the two different microscope metadata, do not match precisely, as clearly visible when comparing panels 1a and 1c. Hence, the cell appears to be much larger in the LM image compared to the EM image and the cropping window does not match (white arrows).

Moreover, the two EM-images presented in Fig. 34 were acquired at different magnifications having a microscope pixel calibration of 15.7 nm/px and 12.7 nm/px for the upper and lower EM image, respectively. In combination with the LM resolution of 83 nm/px, this yields a PSR of 5.28 and 6.55, respectively. Column c displays the result of resizing the cropped LM images using the metadata obtained PSR. It clearly shows a discrepancy of the FoV with respect to the EM image. However, because I am aiming to extract the same feature from both images, and also have an internal calibration standard. The process of calibrating the PSR is by simply comparing the distance of identifiable landmarks in both images, as indicated by the red arrows in Fig. 34. Here, the ratio of the distances in pixels is used to calculate the PSR, which finally leads to an equivalent FoV, with the landmark distances having the same pixel length, as displayed in column d, with the manually calculated PSR values of 5.1253 and 6.095, which deviate from the microscope calibrations by 3% and 7%, respectively.

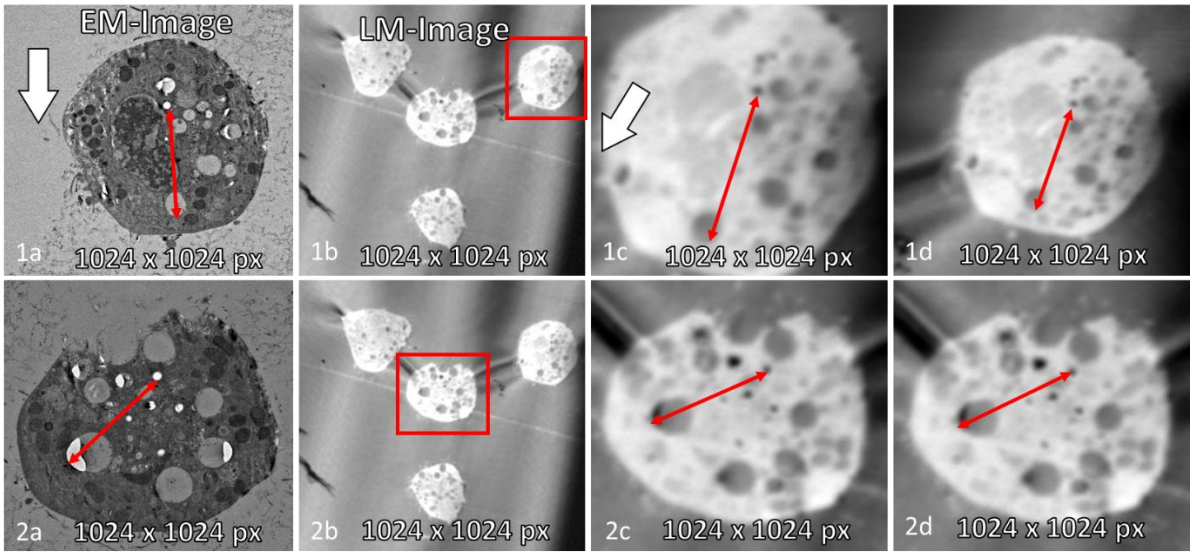


*Fig. 34: Pixel size calibration process: Columns a) and b) display the raw EM and LM input images. It should be noted, that the EM images have been acquired at different magnifications. As can be seen the FoV of the LM image is considerably larger than the EM image; the area corresponding to the EM image a) is marked by the red box in the LM image b). Both input images have the same image dimension of 1024 x 1024 px. In order to maintain the image dimensions, we need to rescale the marked area in the LM image to achieve the same image dimension of the EM input. c) Rescaling the LM image using the microscope pixel calibration values yields a distinct deviation in the FoV for the upper panel images, as can be seen by the white arrows. d) Manually calibrated pixel size ratio by using common landmarks in both on the images. This finally yield a match in the FoV of both images. Although 2c) visually seems to display the similar FoV, the measured length of the landmarks (red lines in 2a and 2c) deviate by 7% and needs to be recalibrated as shown in 2d), where the mismatch of the landmarks is compensated.*

Subsequently, a template for the area based TM has to be prepared from the lower FoV input image, in this case from the EM image. This involves downscaling the EM image to a certain factor. Fig. 35 displays the influence of this downscaling factor, which has to be carefully determined. Depending on the downscaling factor, the image dimension of the generated template (Fig. 35 b) can be adjusted. Here, we used different downscaling factors, ranging from 1/7 to 1/10, yielding template dimensions between 732 x 508 px and 512 x 354 px, respectively. With these templates we applied the area based TM using normalized cross correlation coefficient method. The respective results for the crop-out in the LM image are displayed in Fig. 35 c. Obviously, the 1/10 downscaling approach yields a cut-off in the FoV and so does the 1/9 downscaling. On the other side, the 1/7 downsizing leads to a mislocalization (blue box in Fig. 35 c). The optimal result is achieved using a downsizing factor of 1/8 (Fig. 35 d). This corresponds exactly to the PSR value of 8.05 between the two input images. Therefore, our observations show that exact resizing of the template with regards to the source-image is an essential prerequisite for successful TM. Hence, we implement the PSR as the crucial parameter for any resizing procedures. At this point, it is worth mentioning again, that the determination of the PSR still is a manual process in our workflow. The development of an unsupervised, automatic PSR determination will be the last step towards a fully autonomous image pair generation.

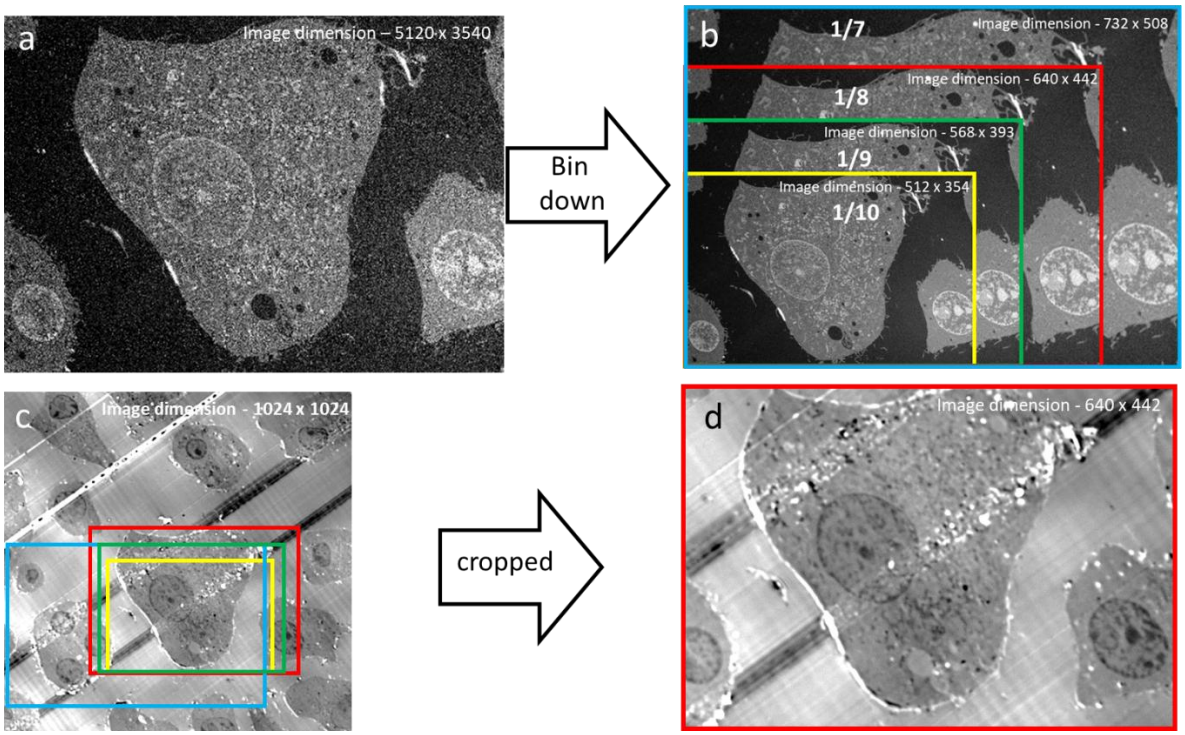


*Fig. 35: Binning, matching and cropping operations: The raw EM input image a) has to be binned down before the area based matching. However, the binning factor plays a crucial role for successful matching. b) Testing different bin-down factors (1/8, 1/9 and 1/10) for the EM-image. c) The matching results obtained for the different binned down EM images are indicated by the coloured boxes. As can be seen, the best matching result is obtained for the 1/8 times binning. This equals exactly the PSR value. For higher bin-down factors the matching fails, as indicated by the yellow and green box. d) displays the cropping result showing a perfect matching between LM and EM image feature.*

Usually, TM approaches are applied on single modal datasets, i.e. that the template image is extracted from the source image itself[104,106]. Therefore, there are naturally no differences between the template and the source in terms of pixel size and information content. However, if the template and source originate from different microscopes, then these two have different characteristics, especially in terms of pixel size. As we have demonstrated in Fig. 35, the precise resizing of the template with regards to the source image pixel size is of utmost importance. Moreover, due to the multimodal nature of source and template, the two input images are looking quite different and yet our matching approach is successful. The underlying key factor for this is passing the template with a Gaussian filter, which enhances its characteristic features and reduces image noise. These enhanced features finally enable successful localization of the template position within the source image, which finally results in the generation of a multi-modal image pair that can be further used in different applications.

Fig. 36 shows two more examples of cropping an EM template image from an LM image source. After resizing the LM image, the image pair has identical image dimensions, e.g. the image dimension of the raw EM input image and common landmarks in both images have the same distance (yellow markers in Fig. 36). However, close inspection of this image pair reveals a very subtle translational shift between both images, as indicated by the white arrow.

Moreover, as can be concluded from the example in the lower pane of Fig. 36, the area based TM can even successfully handle artefacts, which appear only in one of the input images. The bright areas in Fig. 36 d are caused by wrinkling of the thin section specimen during sample preparation and thus induce a high signal in the reflected channel of the cLSM. Accordingly, the TM approach is very robust, even if one of the input images contains additional features, such as the wrinkles in this case.

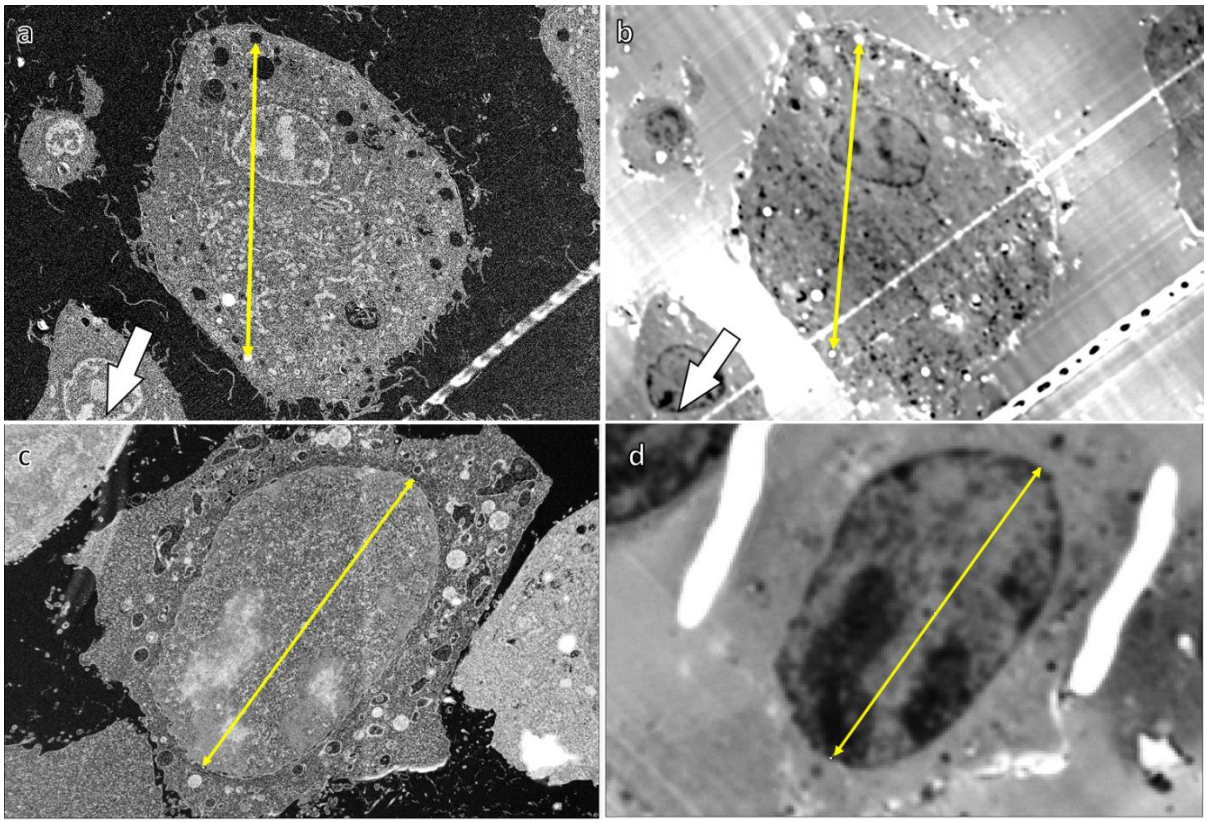


*Fig. 36: Examples of processed image pairs, readily precisely cropped and pixel size calibrated. Images a and c display the input EM image, with pixel dimensions of 5120 x 3540 px, b and d show the precisely cropped area from the raw LM image with equal dimension of 5120 x 3540 px. Due to the prior determination of the exact pixel size ratio between both input images, the LM and EM images have exactly the same pixel calibration. This can be easily checked by measuring the distance between common features, as shown by the yellow line. These two distances in the two pairs of images have the same length in pixels (landmark distance). Furthermore, it is worth mentioning that the area-based template matching still works, even if the image pairs differ significantly due to artefacts, as can be seen from the two bright objects in d. These are imaging artefacts caused by wrinkling of the thin sections in the LM, which are not visualized by the EM.*

The upscaled and cropped LM image (including all channels, reflected and fluorescence) with correspondence to the FoV of the original EM image contribute an image pair, that can be further used for image registration applications. As displayed in Fig. 37, the simple overlay of the image pair already yields a reasonable registration. To illustrate the correspondence between the two images, I have superimposed the EM and the reflected channel of the LM image. This representation is the best way to recognize shifts between the two images and the slight misalignments of the simple overlay are clearly visible, as indicated by the white arrows

in Fig. 37. Using the fluorescence channel, however, it is almost impossible to judge the quality and precision of the overlay, as can be seen in Fig. 52 and 53.

In order to demonstrate the use of this pre-processing in the framework of an automated registration routine, we have applied our custom made machine learning based registration software (https://keeper.mpdl.mpg.de/f/6ad35c6b234f45838a08/) (Appendix III) to the generated image pairs. As shown in the right panel of Fig. 37, the slight mismatch between the input images can subsequently be eliminated by ML based approaches. However, I found that even for ML approaches a low mismatch error below 1% is required for successful image registration. Therefore, the precise rescaling of the images in a pre-processing step is of utmost importance. The total computation time of our pipeline from starting to end is less than a minute. Hence, it is fast, easy to implement and reliable on a common desktop PC. I have developed our supervised image pre-processing pipeline in the Jupyter Notebook using Python on the Windows 11 Pro Desktop. The configuration consists of 11$^{th}$ Gen Intel(R) core™ CPU i9-11900k @ 3.50 GHz 3.50 GHz with 128 GB RAM having Nvidia GeForce RTX 3080 Graphic card. As a note, I want to highlight that all the operations of our pipeline are executed only on the CPU core, hence no GPU compatibility is required.

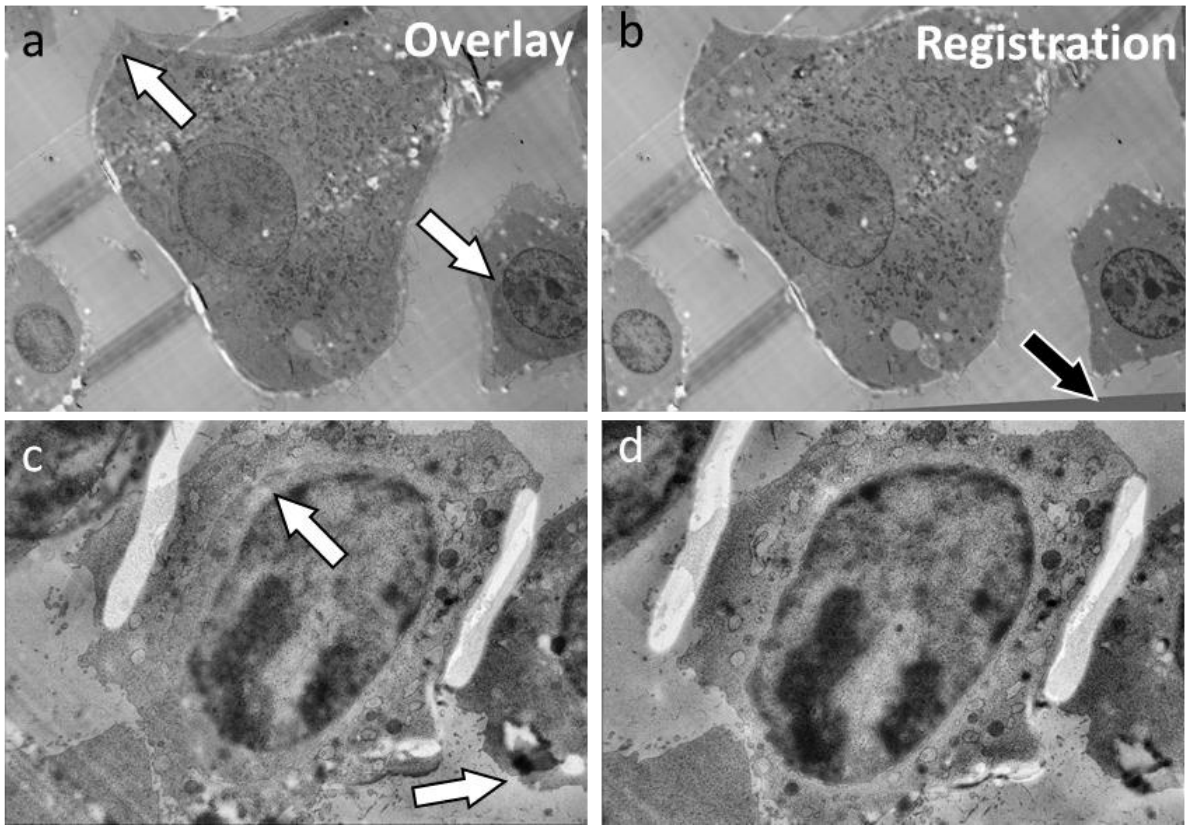


*Fig. 37: Registration results of cropped image pairs from Fig. 35 a and d, and Fig. 36 c and d. Here, the images display an overlay of the EM and the reflective channel of the LM image. This type of representation makes it possible to identify shifts between the two images. In the left column*

*(a and c) the cropped image pairs are simply overlaid, without any further image registration. The white arrows indicate features, where the offset between both images is clearly visible. In the right column (b and d), I applied our ML registration software to align both input images, yielding a precise matching. Even a slight rotation between the two input images can be compensated for with such an approach (black arrow).*

During the development of this weakly supervised preprocesscing pipeline, we have focused on two aspects in particular: Time efficiency and improving accuracy with easy implementation on different operating systems (OS) e.g. Windows and Linux, therefore enabling straightforward operation.

As the IR workflow presented in Chapter 3 calculates a transformation matrix, the precise cropping of the common FoV from EM and LM images is not only a matter of convenience, but it can significantly reduce computational load. With the creation of image pairs without translational offset, e.g. shift in x- and y-direction, the computational costly determination of the transformation is necessary for the first image pair, only. Any other image pairs, extracted from the same dataset, have the same rotational offset and hence, the already calculated transformation matrix can be applied to any of those image pairs. This is demonstrated in Fig. 38. The initial image pair 1a and 1b has been used to train the transformation matrix. Then a smaller part of the input images was cropped using my software approach (marked by the yellow box). The IR in sub-image 2c of Fig. 38 was done just using the already available transformation matrix yielding a perfect registration of the LM and the EM image. Accordingly, the computing time for the following image pairs is reduced to just a few seconds. However, this requires that the rotational offset of all data sets is consistent. However, this is usually the case for microscopic examination of a sample. The rotation occurs when the specimen is transferred from one microscope to another, but it then remains constant for all images captured. Therefore, my software is ideal for fast batch processing of many image pairs.

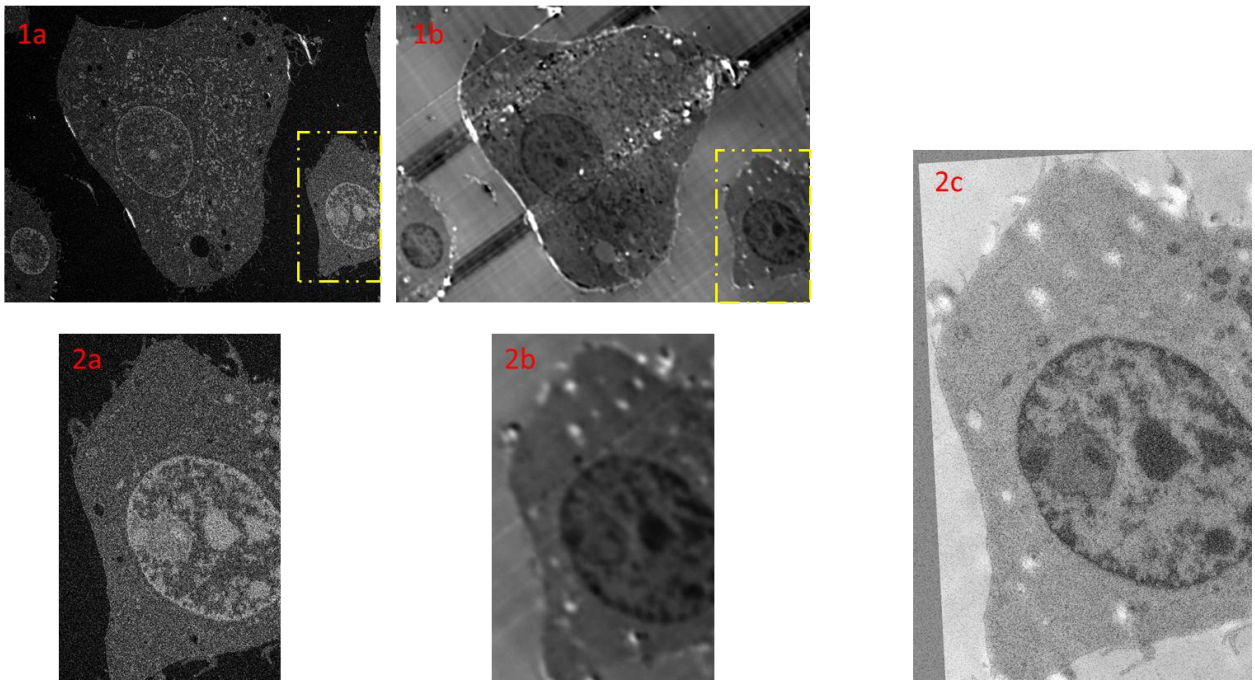


*Fig. 38: 1a and 1b represent the EM-LM image pair highlighting the rectangular region to be cropped, 2a and 2b are the processed EM and LM image-pair from 1a and 1b, and 2c represents CLEM registered image-pair generated with 2a and 2b image-pair, using the same transformation matrix as for 1a and 1b.*

## Super-resolution generative adversarial network (SRGAN)

The super-resolution generative adversarial network (SRGAN) algorithm has been optimized for the upscaling of images, e.g. to increase the image dimensions. However, this is equivalent to a decrease of the pixel size and hence an increase in resolution. It is preset to increase the image dimension by a factor of 4. I have tested the SRGAN on a SEM image in Fig. 39. The upper panel shows the original EM image at full resolution (a) and with reduced image dimension (b). The latter (reduced) image was used for the SRGAN reconstruction in c), generating a 4820 x 3556 pixel image from the 1205 x 889 pixel input. However, at this display size, no differences are noticeable. Therefore, the middle panel displays the zoom-in to the area marked by the red box. d) in the zoom the original image shows a quite high noise level, especially in the area of the cytoplasm. By reducing the image dimension (increasing the pixel size) the image is blurred and the noise is reduced (e). The SRGAN reconstructed image in f, shows very fine details that cannot be corroborated from the original image, e.g. there is a certain fine structure in the displayed mitochondria that makes no sense. On the other side, some subtle structures, visible in the original image (e.g. the elliptical object marked by the white arrow in d) are not properly reconstructed by this generative model.

The reason for this behavior is shown in the lower panel of Fig. 39. The generative model is optimized using common RGB photographs, like in Fig. 39 g. Here, the SRGAN adds common, averaged patterns to compensate the missing image information, as shown by the comparison of Fig. 39 h and i. However, for any scientific analysis, this “educated guess” to fill missing information is useless.

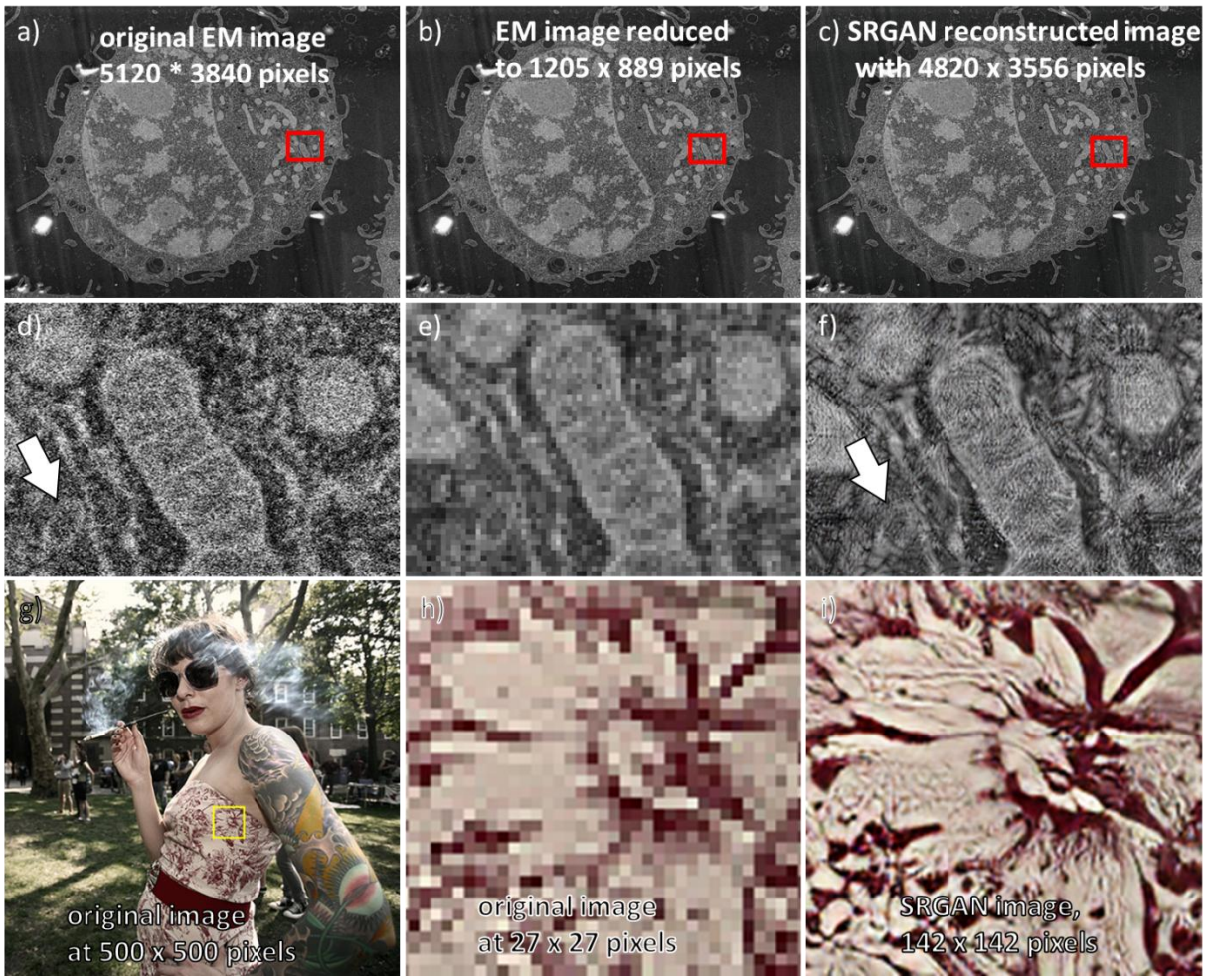


*Fig. 39: Testing the SRGAN performance on a SEM greyscale image: a) raw and b) downscaled EM image. The downscaled SEM image is then used to reconstruct the missing information and to upscale the image again (c). However, at this display size no differences are visible. For this, it is necessary to zoom in. d) to f) show the area marked by the red boxes. The original EM image is quite noisy, but fine structure details are visible. e) at larger pixel size (lower resolution), the image appears more blurry, similar to the Gaussian blur. f) SRGRAN reconstruction from the downscaled EM image. Very fine image features have been added compared to the original image and on the other side, some features go lost (e.g. the elliptic object marked by the white arrow). The reason for this behavior might be attributed to the fact, that the SRGRAN is optimized using common, RGB photographs, like in g). Here, the comparison of the original, low resolution details (h) with the reconstructed image (i) is impressive, but this is not a real improvement of resolution but an image post-processing using averaged, most probable expectation. Accordingly, SRGAN is not suitable for any scientific meaningful image resolution enhancement.*

But SRGAN can nevertheless be used usefully for the IR of CLEM images, as demonstrated in Fig. 40. As the LM image always has a larger pixel size it needs to be upscaled to fit the

EM image. And here either normal upscaling software can be used, or this can be done using SRGAN. The result for the CLEM IR is equivalently perfect for both approaches.

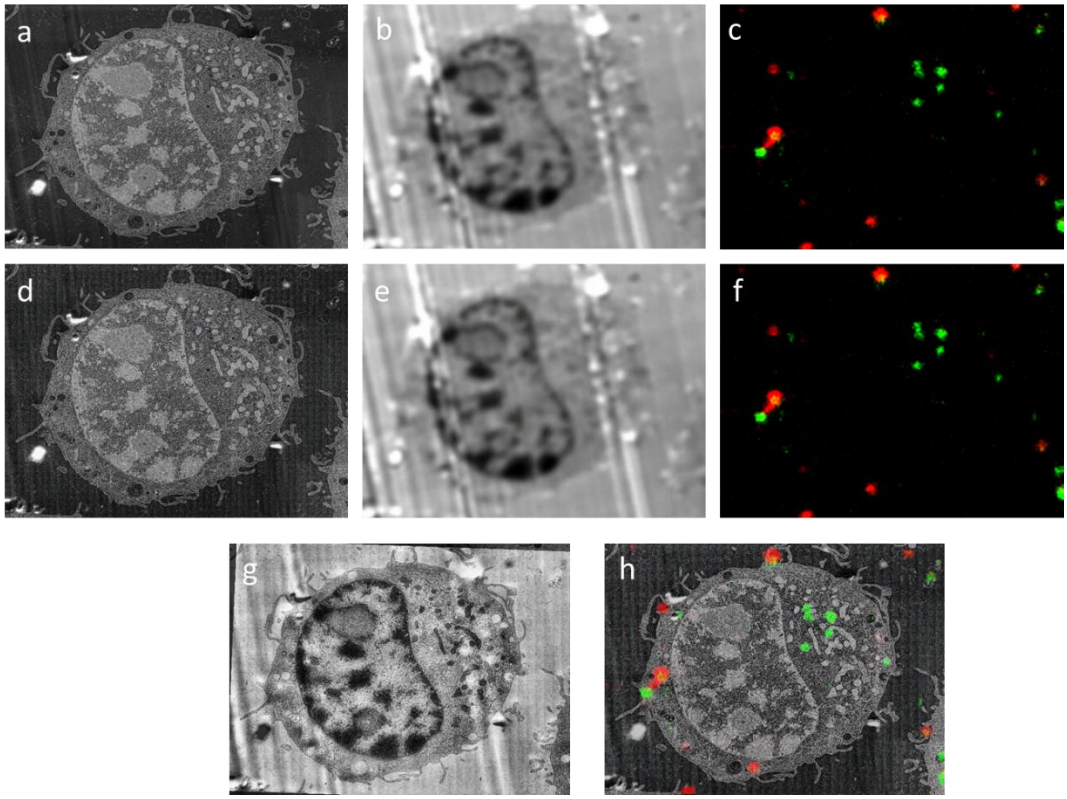


*Fig. 40: CLEM set of LM and EM micrographs generated using SRGAN model. a), b) and c) are 1k x 1k original microscopy images and d), e) and f) are the corresponding 4k x 4k images, up scaled using the SRGAN model. Finally, g) and h) are the IR CLEM results using the SRGAN up scaled images. The overlay of the EM with the LM reflected channel shows perfect matching. Hence, the two color FM overly in h) is a precise registration as well.*

This chapter 5 is adapted from a submitted manuscript and its supplementary information.

*"A Weakly Supervised Pre-processing Pipeline for Multi-Modal Image Pair Generation for Image Registration"*

D. Daksh, A. Kaltbeitzel, G. Glaßer, K. Landfester, I. Lieberwirth

Paper Contributions:

Daksh (first author) - conceptualization, methodology, coding and designing the neural network pipeline, setup different test cases, writing, preparing and editing of manuscript, data analysis and interpretation of all corresponding results, Anke Kaltbeitzel & Gunnar Glaßer - acquire the different sample images of LM and EM, Katharina Landfester & Ingo Lieberwirth - acquiring funding for the project, design and discussion of the concept, data analysis and interpretation of experimental results, and editing of the manuscript.

# Source code for weakly supervised image pre-processing pipeline

The algorithm is implemented in Jupyter Notebook using Anaconda navigator. Libraries are installed and updated using pip/conda. The source code is presented below in the cell format.

Step 1: Setting up the environment, libraries and import source and reference images

```
import os
import math
import numpy as np
import cv2
import random

import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/C4/R.tif")
img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

print('Original Confocal Image Dimensions:',img.shape)
plt.imshow(img)
plt.show()
img2 = img.copy()

resized = img

# Read the template
template = cv2.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/E41_1.tif")
template = cv.cvtColor(template, cv.COLOR_BGR2GRAY)
print('Template EM Image Dimensions:',template.shape)
plt.imshow(template)
plt.show()
```

Step 2: Applying template matching method for matching between source and template

```
w, h = template.shape[::-1]

# Choose the matching method
method = cv.TM_CCORR_NORMED

# Apply template Matching
res = cv.matchTemplate(resized, template, method)

# Find the location of the best match
min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
```

Step 3: Setting up the rectangle marking the matched region and crop out the matched region

```
threshold = 0.1  # Try lowering if necessary
if max_val >= threshold:
    top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(resized, top_left, bottom_right, color=(255, 0, 0), thickness=10)
else:
    print("No good match found.")

# Display the results
plt.subplot(121), plt.imshow(res)
plt.title('Matching Result'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(resized)
plt.title('Detected Point'), plt.xticks([]), plt.yticks([])
plt.show()

# Print the coordinates
print("Coordinates for the rectangle:", top_left, bottom_right)

x1, y1 = top_left  # Top-left corner coordinates
x2, y2 = bottom_right  # Bottom-right corner coordinates

img = cv.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/C4/R.tif")

im1 = img[y1:y2, x1:x2]           # cv.TM_CCORR_NORMED  (45, 471) (685, 916)

plt.imshow(im1)
plt.show()
print(im1.shape)
cv2.imwrite("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/R_crop.tif", im1)

img = cv.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/C4/F.tif")

im1 = img[y1:y2, x1:x2]           # cv.TM_CCORR_NORMED  (45, 471) (685, 916)

plt.imshow(im1)
plt.show()
print(im1.shape)
cv2.imwrite("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/F_crop.tif", im1)


img = cv2.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/R_crop.tif", cv2.IMREAD_UNCHANGED)

print('Original Dimensions : ',img.shape)
```

Step 4: Scaling up of the cropped reflected and fluorescence region with the precise pixel size ratio

```
plt.imshow(img)
plt.show()

#scale_percent = 12.724
#scale_percent = 6.362
scale_percent = 8.05 # percent of pixel size in both images
#scale_percent = 12.8404 # percent of pixel size in both images
width = int(img.shape[1] * scale_percent)
height = int(img.shape[0] * scale_percent)
dim = (width, height)

# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

print('Resized Dimensions : ',resized.shape)

#cv2.imshow("Resized image", resized)
#cv2.waitKey(0)
#cv2.destroyAllWindows()

plt.imshow(resized)
plt.show()


J = resized[0:3540, 0:5120]
cv2.imwrite('C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/R_1.tif', J)
print('new Dimensions : ',J.shape)
plt.imshow(J)
plt.title("LM Image - new")
plt.show()

img = cv2.imread("C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/F_crop.tif", cv2.IMREAD_UNCHANGED)

print('Original Dimensions : ',img.shape)

plt.imshow(img)
plt.show()
```

```
#scale_percent = 12.724
#scale_percent = 6.362
#scale_percent = 6.4458 # percent of pixel size in both images
#scale_percent = 12.48 # percent of pixel size in both images
width = int(img.shape[1] * scale_percent)
height = int(img.shape[0] * scale_percent)
dim = (width, height)

# resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

print('Resized Dimensions : ',resized.shape)

#cv2.imshow("Resized image", resized)
#cv2.waitKey(0)
#cv2.destroyAllWindows()

plt.imshow(resized)
plt.show()


J = resized[0:3540, 0:5120]
cv2.imwrite('C:/Users/daksh/Desktop/Jupyter notebook/Data/Datasets/new D2/Crop-Dataset2/IPs/F_1.tif', J)
print('new Dimensions : ',J.shape)
plt.imshow(J)
plt.title("LM Image - new")
plt.show()
```

# Chapter 6: Conclusions

The automated registration of multimodal image data is still a challenge for modern image analysis methods. The problem arises from the fact that images from different sources represent complementary, sometimes inconsistent, and content. However, the crucial point for CLEM is, that it yields added value by merging molecular and morphological information. In this work, I presented a deep learning based workflow to register EM and LM image pairs in combination with a detailed pre-processing, TM based procedure. Although the development and the testing has been done on microscopy images only, I can also envisage my software being used in completely different areas, such as astronomy, satellite-based earth observation or autonomous, self-driving systems.

The input images can therefore be quite different and represent different data content. This is even the case for the EM and LM images I examined. For this reason, I opted for an unsupervised, training-free approach to keep the software as flexible as possible. However, it has been shown that the depth of information of the fluorescence image is not sufficient to ensure this flexibility. In contrast, the reflected LM image has much more information overlap with the EM image. Therefore, the choice of the reflected LM image was ultimately the path to successful automated registration. A very important aspect for any automated registration is prior image processing, in particular matching the pixel size in both input channels as precisely as possible. The larger the dimension of the image, the more precisely the pixel size must be calibrated in both input channels, as shown in Fig. 9. Another new approach in the software is the calculation of a transformation matrix. This allows to flexibly overlay each channel of the LM image with the EM image. Even more, I can reduce the size of the input images (reduce the image dimension) and then apply the resulting transformation matrix to the original size, which significantly reduces the calculation time. But the once-only determination of the transformation matrix is also a significant advantage for batch processing: it is no problem to take large-format microscopic images with a very large image dimension of 11k x 11k pixels (or images generated by stitching). If it is possible to automatically cut out pieces with the exact same FoV from each of these data sets, then it is sufficient to calculate a single transformation matrix, which is then applied to all pairs of images cropped out, as the relative rotation and shifting between the two input sources remains constant.

With this, I am presenting a novel and easily implemented weakly supervised image pre-processing algorithm that efficiently generates image-pairs of different modalities and imaging systems. It not only solves the challenges and hurdles of having consistent resolution but also serves as starting point for analysis of image processing. It is worth stating that the precise determination of the PSR between both image modalities is of utmost importance for all subsequent image analysis procedures and that the pixel calibration metadata coming from both microscopes may not be sufficiently accurate. By enabling the seamless integration of images of different resolutions, my pipeline enhances multimodal analysis, facilitates high-throughput processing, improves data accuracy, and supports the development of multiscale models. Additionally, my pre-processing system might indicate the general structure for workflows that are more efficient, reduce human error, and enable innovative research that requires precise and accurate image fusion. I hope that my work will further be used in different applications and helps in producing fruitful scientific breakthroughs in image analysis with easing the starting steps of image pre-processing.

## Open ongoing challenges & Future research

I have faced many challenges that accidentally arise while working on automated CLEM registration. From keeping the full image information without losing any pixel to successfully save the whole CLEM micrograph with all the mathematical operations is a big challenge. I found out the solutions to handle the missing information image-pair by giving the concept of padding and how to implement the padding. Meanwhile, I highlight the issues with hardware as overuse or excess use of Nvidia GPU causes generation of extreme heat and needs to be cooled down, otherwise the generated heat can be a reason for bad alignments. And also the prolonged use of Nvidia GPU card need to be replaced or it can show the degradation effect. The one challenge that is still to be considered is the batch processing of image-pairs for image registrations. As my software/tool needs manual starting points (the landmark measuring shown in Fig. 36), it is less robust because the user interface (UI) overhead leads to a performance impact when handling either larger number of operations or data. The UI overhead demonstrates as slow response times and increased memory usage. That means UI overhead needs to be optimized for batch processing involving minimum number of user interactions and optimizing overall performance with potentially offloading the unnecessary tasks in the background processes. I am hoping that

in the future either via using scheduler with airflow or command line script usage can show up the potential in this regard. This is a highly challenging task that necessitates the use of high-performance computing on powerful servers, capable of running processes in parallel and periodically optimizing them in the background.

# References


1 Petrou, M. M. & Petrou, C. *Image processing: the fundamentals*. (John Wiley & Sons, 2010).

2 Gonzales, R. C. & Wintz, P. *Digital image processing*. (Addison-Wesley Longman Publishing Co., Inc., 1987).

3 Brown, L. G. A survey of image registration techniques. *ACM computing surveys (CSUR)* **24**, 325-376 (1992).

4 Chen, J. *et al.* A survey on deep learning in medical image registration: New technologies, uncertainty, evaluation metrics, and beyond. *Medical Image Analysis*, 103385 (2024).

5 Bharati, S., Mondal, M., Podder, P. & Prasath, V. Deep learning for medical image registration: A comprehensive review. *arXiv preprint arXiv:2204.11341* (2022).

6 Salvi, M., Acharya, U. R., Molinari, F. & Meiburger, K. M. The impact of pre-and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. *Computers in Biology and Medicine* **128**, 104129 (2021).

7 Bow, S. T. *Pattern recognition and image preprocessing*. (CRC press, 2002).

8 Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *nature* **529**, 484-489 (2016).

9 Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *nature* **596**, 583-589 (2021).

10 *The Nobel Prize in Chemistry 2024*, <https://www.nobelprize.org/prizes/chemistry/2024/summary/> (2025).

11 Murphy, D. B. & Davidson, M. W. *Fundamentals of light microscopy and electronic imaging*. (John Wiley & Sons, 2012).

12 Slayter, E. M. & Slayter, H. S. *Light and electron microscopy*. (Cambridge University Press, 1992).

13 Abbe, E. Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung. *Archiv für mikroskopische Anatomie* **9**, 413-468 (1873).

14 Egerton, R. F. *Physical principles of electron microscopy*. Vol. 56 (Springer, 2005).

15 Goodhew, P. J. & Humphreys, J. *Electron microscopy and analysis*. (CRC press, 2000).

16 Merchant, F. & Castleman, K. *Microscope image processing*. (Academic press, 2022).

17 Goodfellow, I. *et al.* Generative adversarial networks. *Communications of the ACM* **63**, 139-144 (2020).

18 Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving language understanding by generative pre-training. (2018).

19 Marcus, G., Davis, E. & Aaronson, S. A very preliminary analysis of DALL-E 2. *arXiv preprint arXiv:2204.13807* (2022).

20 Dargan, S., Kumar, M., Ayyagari, M. R. & Kumar, G. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of computational methods in engineering* **27**, 1071-1092 (2020).

21 Razghandi, M., Zhou, H., Erol-Kantarci, M. & Turgut, D. in *ICC 2022-IEEE International Conference on Communications.* 4781-4786 (IEEE).

22 Bachimanchi, H. & Volpe, G. Diffusion Models to Enhance the Resolution of Microscopy Images: A Tutorial. *arXiv preprint arXiv:2409.16488* (2024).

23 Van der Walt, S. *et al.* scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014).
24 Ledig, C. *et al.* in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 4681-4690.
25 Wang, X. *et al.* in *Proceedings of the European conference on computer vision (ECCV) workshops.* 0-0.
26 Lu, Y., Wu, S., Tai, Y.-W. & Tang, C.-K. in *Proceedings of the European conference on computer vision (ECCV).* 205-220.
27 Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. in *Proceedings of the IEEE international conference on computer vision.* 2223-2232.
28 Khemakhem, F. & Ltifi, H. Neural style transfer generative adversarial network (NST-GAN) for facial expression recognition. *International Journal of Multimedia Information Retrieval* **12**, 26 (2023).
29 Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 1125-1134.
30 Karras, T., Laine, S. & Aila, T. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 4401-4410.
31 Tripathy, S., Kannala, J. & Rahtu, E. in *Proceedings of the IEEE/CVF winter conference on applications of computer vision.* 1329-1338.
32 Armanious, K. *et al.* MedGAN: Medical image translation using GANs. *Computerized medical imaging and graphics* **79**, 101684 (2020).
33 Shahbazi, M. *et al.* in *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2888-2898.
34 Wu, J., Zhang, C., Xue, T., Freeman, B. & Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems* **29** (2016).
35 Ronneberger, O., Fischer, P. & Brox, T. in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18.* 234-241 (Springer).
36 Wang, Y., Friedrich, H., Voets, I. K., Zijlstra, P. & Albertazzi, L. Correlative imaging for polymer science. *Journal of Polymer Science* **59**, 1232-1240 (2021).
37 Zhang, X., Zhang, W., Peng, J. & Fan, J. Automatic Image Labelling at Pixel Level. *arXiv preprint arXiv:2007.07415* (2020).
38 de Beer, M. *et al.* Precise targeting for 3D cryo-correlative light and electron microscopy volume imaging of tissues using a FinderTOP. *bioRxiv*, 2022.2007. 2031.502212 (2022).
39 Goshtasby, A. A. *Image registration: Principles, tools and methods*. (Springer Science & Business Media, 2012).
40 Nag, S. Image registration techniques: a survey. *arXiv preprint arXiv:1712.07540* (2017).
41 Sarvaiya, J. N., Patnaik, S. & Bombaywala, S. in *2009 international conference on advances in computing, control, and telecommunication technologies.* 819-822 (IEEE).
42 Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J. & Dalca, A. V. Voxelmorph: a learning framework for deformable medical image registration. *IEEE transactions on medical imaging* **38**, 1788-1800 (2019).
43 Fu, Y. *et al.* Deep learning in medical image registration: a review. *Physics in Medicine & Biology* **65**, 20TR01 (2020).

44 De Vos, B. D. *et al.* A deep learning framework for unsupervised affine and deformable image registration. *Medical image analysis* **52**, 128-143 (2019).
45 De Chaumont, F. *et al.* Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods* **9**, 690-696 (2012).
46 Szeliski, R. *Computer vision: algorithms and applications*. (Springer Nature, 2022).
47 Rivenson, Y. *et al.* Deep learning microscopy. *Optica* **4**, 1437-1443 (2017).
48 Zhou, Z.-H. *Machine learning*. (Springer nature, 2021).
49 Mahesh, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]* **9**, 381-386 (2020).
50 Slomka, P. J. & Baum, R. P. Multimodality image registration with software: state-of-the-art. *European journal of nuclear medicine and molecular imaging* **36**, 44-55 (2009).
51 Simonovsky, M., Gutiérrez-Becker, B., Mateus, D., Navab, N. & Komodakis, N. in *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III 19.* 10-18 (Springer).
52 Seifert, R. *et al.* DeepCLEM: automated registration for correlative light and electron microscopy using deep learning. *F1000Research* **9**, 1275 (2023).
53 Zou, J., Gao, B., Song, Y. & Qin, J. A review of deep learning-based deformable medical image registration. *Frontiers in Oncology* **12**, 1047215 (2022).
54 Chen, J. *et al.* Transmorph: Transformer for unsupervised medical image registration. *Medical image analysis* **82**, 102615 (2022).
55 Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging* **29**, 196-205 (2009).
56 Krentzel, D. *et al.* CLEM-Reg: An automated point cloud based registration algorithm for correlative light and volume electron microscopy. *BioRxiv*, 2023.2005. 2011.540445 (2023).
57 Pluim, J. P., Maintz, J. A. & Viergever, M. A. Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging* **22**, 986-1004 (2003).
58 Gaens, T., Maes, F., Vandermeulen, D. & Suetens, P. in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* 1099-1106 (Springer).
59 O'shea, K. & Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458* (2015).
60 Hu, Y. *et al.* Weakly-supervised convolutional neural networks for multimodal image registration. *Medical image analysis* **49**, 1-13 (2018).
61 Zhou, S. *et al.* in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22.* 478-486 (Springer).
62 Zhu, Y., Zhou Sr, Z., Liao Sr, G. & Yuan, K. in *Medical Imaging 2020: Image Processing.* 596-603 (SPIE).
63 Ma, K. *et al.* in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20.* 240-248 (Springer).
64 Potier, G., Lavancier, F., Kunne, S. & Paul-Gilloteaux, P. in *2021 IEEE International Conference on Image Processing (ICIP).* 131-135 (IEEE).

65 Xiao, H., Ren, G. & Cai, J. A review on 3D deformable image registration and its application in dose warping. *Radiation Medicine and Protection* **1**, 171-178 (2020).
66 Haskins, G., Kruger, U. & Yan, P. Deep learning in medical image registration: a survey. *Machine Vision and Applications* **31**, 1-18 (2020).
67 Serra Lleti, J. M. *et al.* CLEM Site, a software for automated phenotypic screens using light microscopy and FIB-SEM. *Journal of Cell Biology* **222**, e202209127 (2022).
68 Abràmoff, M. D., Magalhães, P. J. & Ram, S. J. Image processing with ImageJ. *Biophotonics international* **11**, 36-42 (2004).
69 Paul-Gilloteaux, P. *et al.* eC-CLEM: flexible multidimensional registration software for correlative microscopies. *Nature methods* **14**, 102-103 (2017).
70 Fu, Z. *et al.* mEosEM withstands osmium staining and Epon embedding for super-resolution CLEM. *Nature methods* **17**, 55-58 (2020).
71 Hoopes, A., Hoffmann, M., Fischl, B., Guttag, J. & Dalca, A. V. in *Information Processing in Medical Imaging: 27th International Conference, IPMI 2021, Virtual Event, June 28–June 30, 2021, Proceedings 27.* 3-17 (Springer).
72 Nan, A., Tennant, M., Rubin, U. & Ray, N. in *Medical Imaging with Deep Learning.* 527-543 (PMLR).
73 Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* **40**, 834-848 (2017).
74 Khadangi, A., Boudier, T. & Rajagopal, V. EM-stellar: benchmarking deep learning for electron microscopy image segmentation. *Bioinformatics* **37**, 97-106 (2021).
75 Long, J., Shelhamer, E. & Darrell, T. in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 3431-3440.
76 Toyooka, K. MirrorCLEM: a seamless system for correlative light and electron microscopy. *The HITACHI Scientific Instrument News* **15** (2020).
77 Pape, C. *et al.* MoBIE: a Fiji plugin for sharing and exploration of multi-modal cloud-hosted big image data. *nature methods* **20**, 475-476 (2023).
78 Vergara, H. M. *et al.* Whole-body integration of gene expression and single-cell morphology. *Cell* **184**, 4819-4837. e4822 (2021).
79 Nan, A., Tennant, M., Rubin, U. & Ray, N. Ordinary Differential Equation and Complex Matrix Exponential for Multi-resolution Image Registration. *arXiv preprint arXiv:2007.13683* (2020).
80 Pal, S., Tennant, M. & Ray, N. in *2022 26th International Conference on Pattern Recognition (ICPR).* 3391-3398 (IEEE).
81 Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Physical review E* **69**, 066138 (2004).
82 DeTone, D., Malisiewicz, T. & Rabinovich, A. Deep image homography estimation. *arXiv preprint arXiv:1606.03798* (2016).
83 Belghazi, M. I. *et al.* in *International conference on machine learning.* 531-540 (PMLR).
84 Snaauw, G. *et al.* Mutual Information Neural Estimation for Unsupervised Multi-Modal Registration of Brain Images. *Annu Int Conf IEEE Eng Med Biol Soc* **2022**, 3510-3513 (2022). https://doi.org/10.1109/EMBC48229.2022.9871220
85 Liu, Z. *et al.* A survey on applications of deep learning in microscopy image analysis. *Computers in biology and medicine* **134**, 104523 (2021).

86 Roche, A., Malandain, G., Pennec, X. & Ayache, N. in *Medical Image Computing and Computer-Assisted Intervention—MICCAI'98: First International Conference Cambridge, MA, USA, October 11–13, 1998 Proceedings 1*. 1115-1124 (Springer).
87 Liu, Z. *et al.* Correlative image-based release prediction and 3D microstructure characterization for a long acting parenteral implant. *Pharmaceutical Research* **38**, 1915-1929 (2021).
88 Hardoon, D. R., Saunders, C., Szedmak, S. & Shawe-Taylor, J. in *Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006 Proceedings 2*. 681-692 (Springer).
89 von Chamier, L. *et al.* Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nature communications* **12**, 2276 (2021).
90 Bischof, J. *et al.* Multimodal bioimaging across disciplines and scales: challenges, opportunities and breaking down barriers. npj Imaging 2. (2024).
91 Voulodimos, A., Doulamis, N., Doulamis, A. & Protopapadakis, E. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience* **2018**, 7068349 (2018).
92 Gómez-de-Mariscal, E. *et al.* DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nature Methods* **18**, 1192-1195 (2021).
93 Schroeder, A. B. *et al.* The ImageJ ecosystem: Open-source software for image visualization, processing, and analysis. *Protein Science* **30**, 234-249 (2021).
94 Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nature methods* **9**, 671-675 (2012).
95 Chavez, C. & Faulkner, A. *Adobe Photoshop Classroom in a Book (2021 release)*. (Adobe Press, 2021).
96 Fedorov, A. *et al.* 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic resonance imaging* **30**, 1323-1341 (2012).
97 Gilat, A. *MATLAB: An introduction with Applications*. (John Wiley & Sons, 2017).
98 Yaniv, Z., Lowekamp, B. C., Johnson, H. J. & Beare, R. SimpleITK image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of digital imaging* **31**, 290-303 (2018).
99 Lowekamp, B. C., Chen, D. T., Ibáñez, L. & Blezek, D. The design of SimpleITK. *Frontiers in neuroinformatics* **7**, 45 (2013).
100 Moore, J. *et al.* in *Medical imaging 2015: image processing*. 37-42 (SPIE).
101 Hiner, M. C., Rueden, C. T. & Eliceiri, K. W. SCIFIO: an extensible framework to support scientific image formats. *BMC bioinformatics* **17**, 1-5 (2016).
102 Ibanez, L., Schroeder, W., Ng, L. & Cates, J. *The ITK software guide*. Vol. 2 (Kitware Clifton Park, NY, 2005).
103 Yoo, T. S. *et al.* in *Medicine Meets Virtual Reality 02/10* 586-592 (IOS press, 2002).
104 Brunelli, R. *Template matching techniques in computer vision: theory and practice*. (John Wiley & Sons, 2009).
105 Briechle, K. & Hanebeck, U. D. in *Optical pattern recognition XII*. 95-102 (SPIE).
106 Hashemi, N. S., Aghdam, R. B., Ghiasi, A. S. B. & Fatemi, P. Template matching advances and applications in image analysis. *arXiv preprint arXiv:1610.07231* (2016).
107 Paddock, S. W. Confocal laser scanning microscopy. *Biotechniques* **27**, 992-1004 (1999).

108 Rella, E. M., Chhatkuli, A., Liu, Y., Konukoglu, E. & Van Gool, L. Zero pixel directional boundary by vector transform. *arXiv preprint arXiv:2203.08795* (2022).
109 Wells III, W. M., Viola, P., Atsumi, H., Nakajima, S. & Kikinis, R. Multi-modal volume registration by maximization of mutual information. *Medical image analysis* **1**, 35-51 (1996).
110 Reddi, S. J., Kale, S. & Kumar, S. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237* (2019).
111 *These 30 Photos Taken At The Same Location At Different Points In Time*, <https://121clicks.com/inspirations/photos-taken-same-location-at-different-points-in-time/> (2021).
112 Hernandez-Matas, C. *et al.* FIRE: fundus image registration dataset. *Artificial Intelligence in Vision and Ophthalmology* **1**, 16-28 (2017).
113 *Webb Image Galleries*, <https://science.nasa.gov/mission/webb/multimedia/images/>.

# Appendices

## Appendix I: Mathematical analysis of transformation matrix & additional CLEM examples

Technically, I used the key concept of computer graphics and advanced theoretical geometry functions for performing the task of vector calculus, differential geometry and tensors. Image registration is the process of aligning two or more images of the same sample or scene taken at different times, from different viewpoints, and by different sensors or cameras. For this reason, the images show different modalities, spatial relationships and finally the geometric alignment of images is performed using a world coordinate system and a transformation matrix.

The World Coordinate System (WCS) is a fixed, global reference frame (concrete coordinate system) used to define the positions and orientations of objects in a 2D or 3D space. All objects and cameras in a scene are ultimately located and oriented with respect to this system. WCS usually implemented via mathematical models, and these transformation to world coordinates is represented in graphical (Fig. 41) and equation form as:

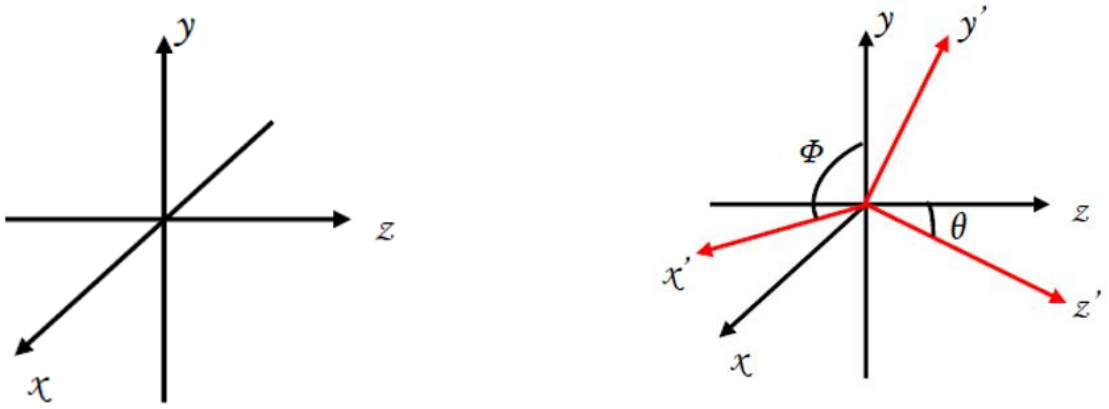


*Fig. 41: Image plane showing the transformation in respect to terms of θ and ϕ.*

So, Transformation matrix is shown as,

$$T = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where, (x, y) is the original coordinates, (x', y') is the transformed coordinates with reference to fixed (x, y) coordinates, $a_{11}$, $a_{12}$, $a_{21}$, $a_{22}$ represent the rotation, shear and scale factors and $t_x$, $t_y$ is translation factors. With respect to the combined scaling and the degree of the rotation θ factor, the transformation matrix can be represented as:

$$T = \begin{bmatrix} s_x.\cos\theta & -s_y.\sin\theta & t_x \\ s_x.\sin\theta & s_y.\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

, where $s_x$ and $s_y$ are combined scaling factors, and θ is the rotation angle.

My algorithm employs matrix exponential to compute the transformation matrix, which are then further applied to the moving image to align it with the fixed image. The whole process involves explicit coordinate mappings, and transformations that defines the characteristic feature of a world coordinate system approach.

Optimizing the Transformation matrix using TFUDL approach:

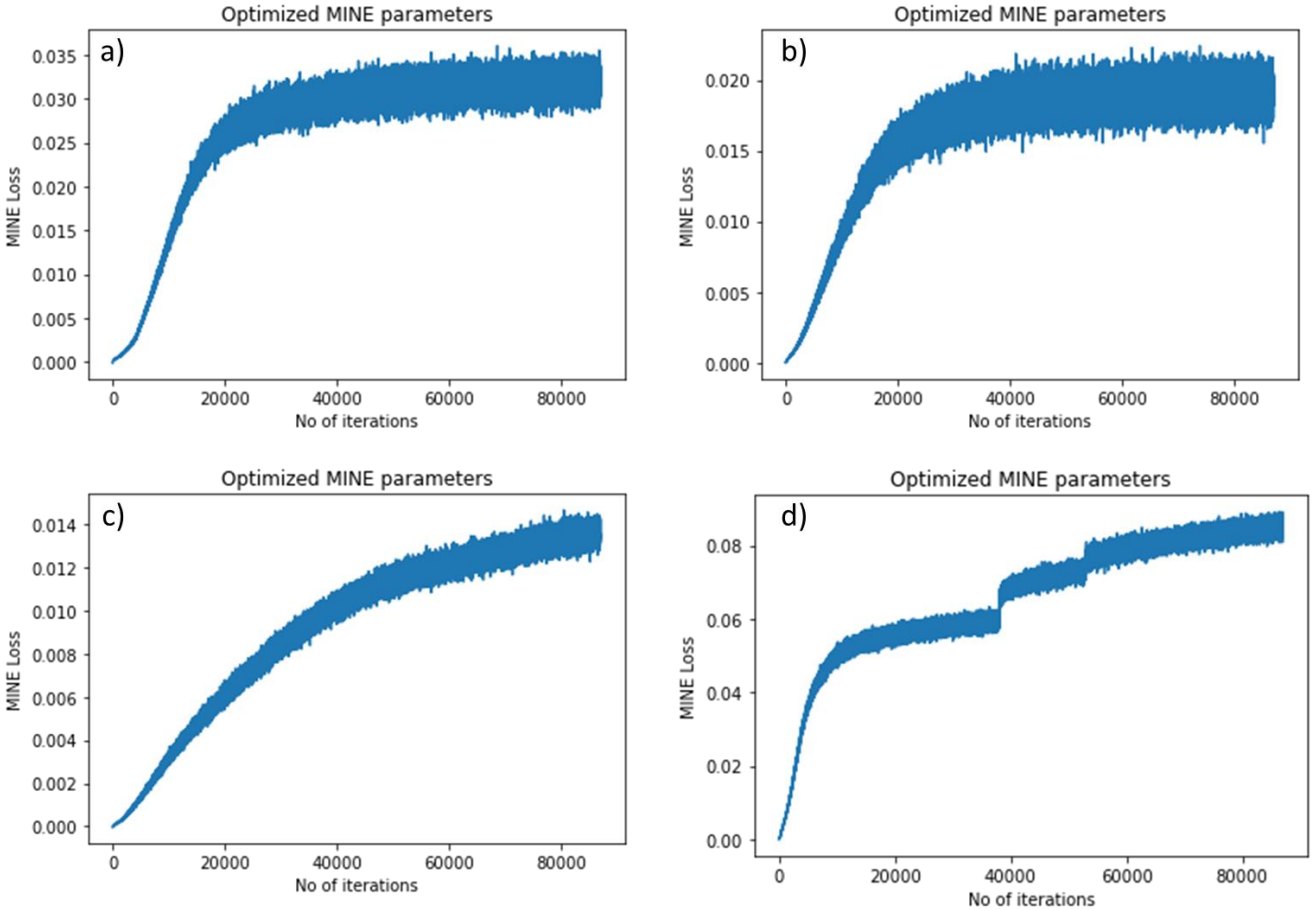


*Fig. 42: a), b) and c) represent the optimized MINE Loss against the number of iterations with a very low or negligible error rate. d) represents the sudden increase in MINE loss (error rate) due to overheating of the GPU.*

# More CLEM micrographs generated using different Image-pairs and conditions

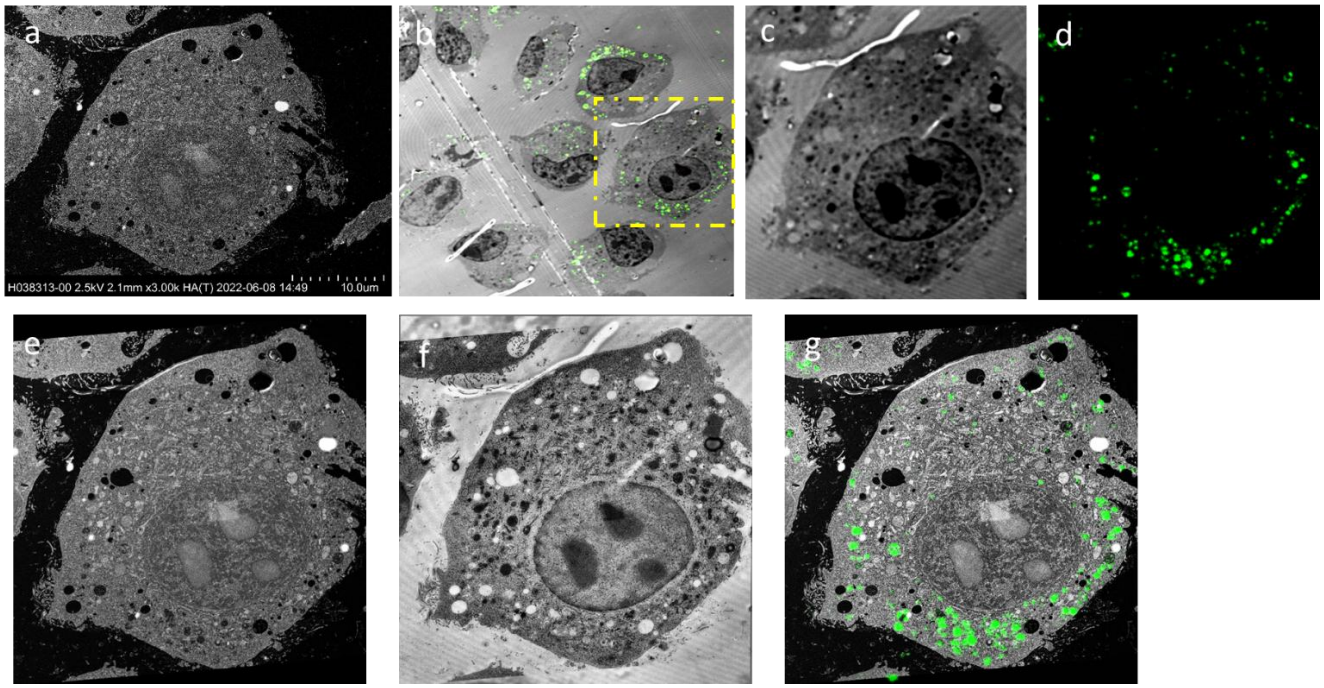


*Fig. 43: A typical representation of a CLEM micrograph from a raw EM-LM image-pair, a) represents the raw EM image, b) represents the raw LM image, c), d) and e) display the respective input images. f) represents the image registration of e) and c) and g) shows the image registration between e) and d) using the generated transformation matrix as calculated in f).*

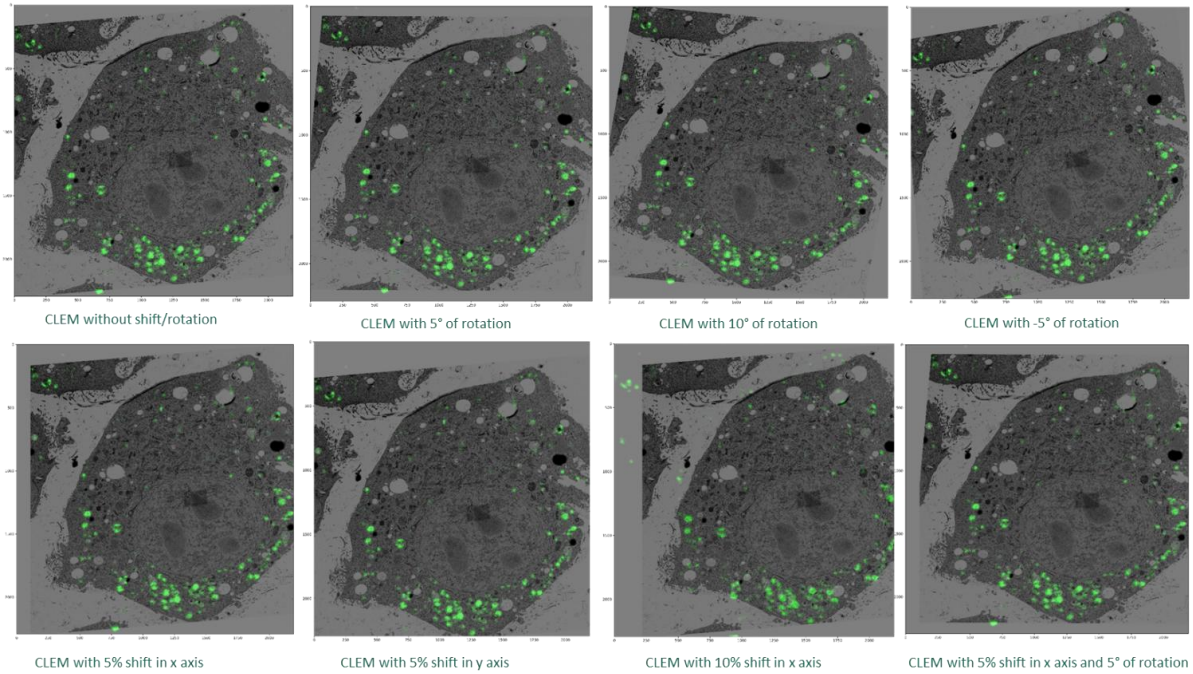


*Fig. 44: CLEM micrographs applied to different shift and rotation offsets, demonstrating the stability and limits of my IR approach. The 10% shift in x-direction is above the tolerance limit and leads to a corrupt IR result.*

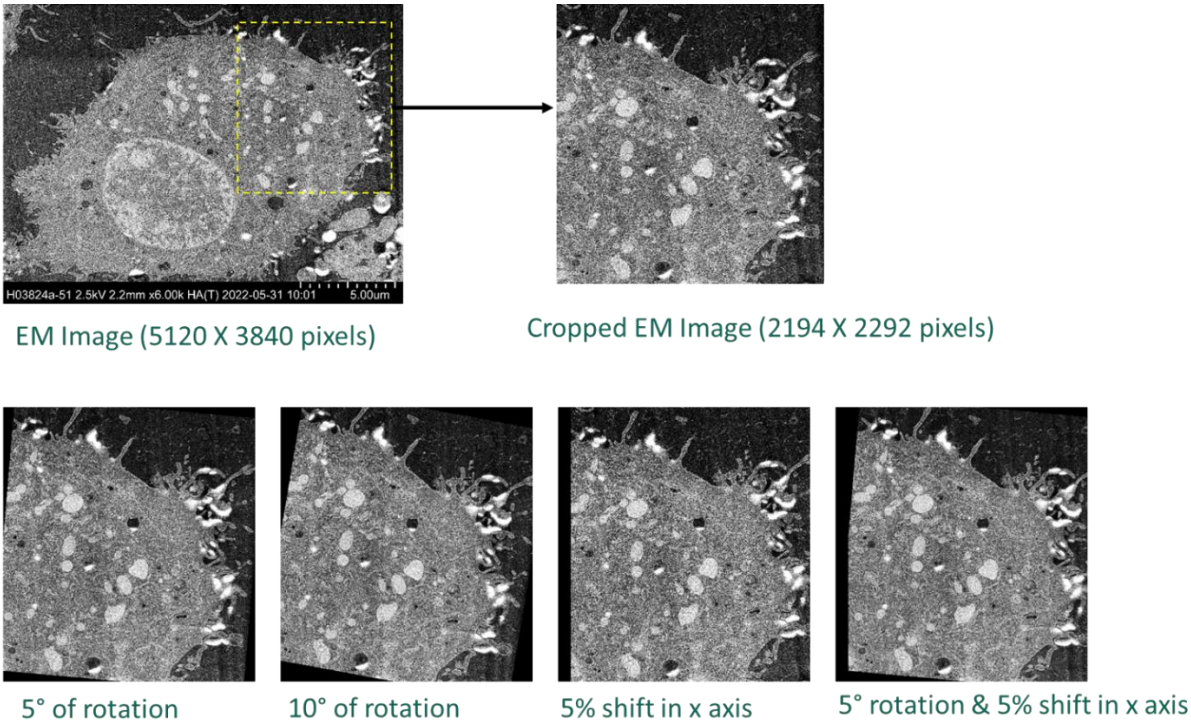


*Fig. 45: Demonstration of the image pair offset: On top row EM image obtained via cropping the area marked by the yellow box. The bottom row displays the result of manually adding shift & rotation operations to generate an offset with regards to the fixed LM image.*

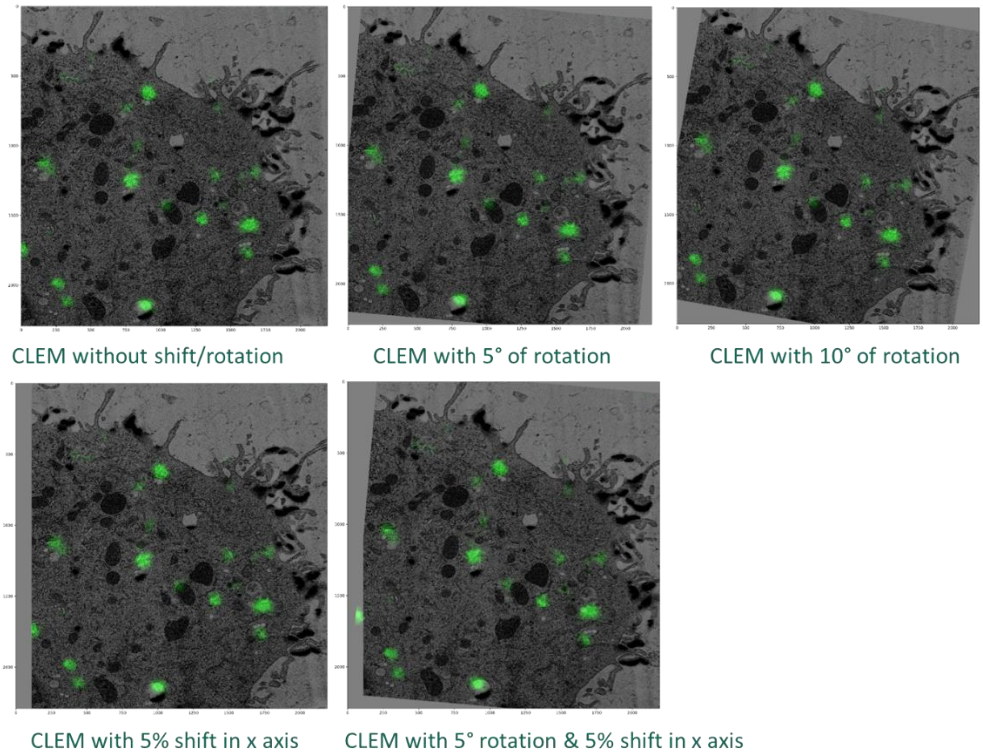


*Fig. 46: IR results from the offsets generated in Fig. 45. Within the tolerable limits of shift and rotation, the registration is successful.*

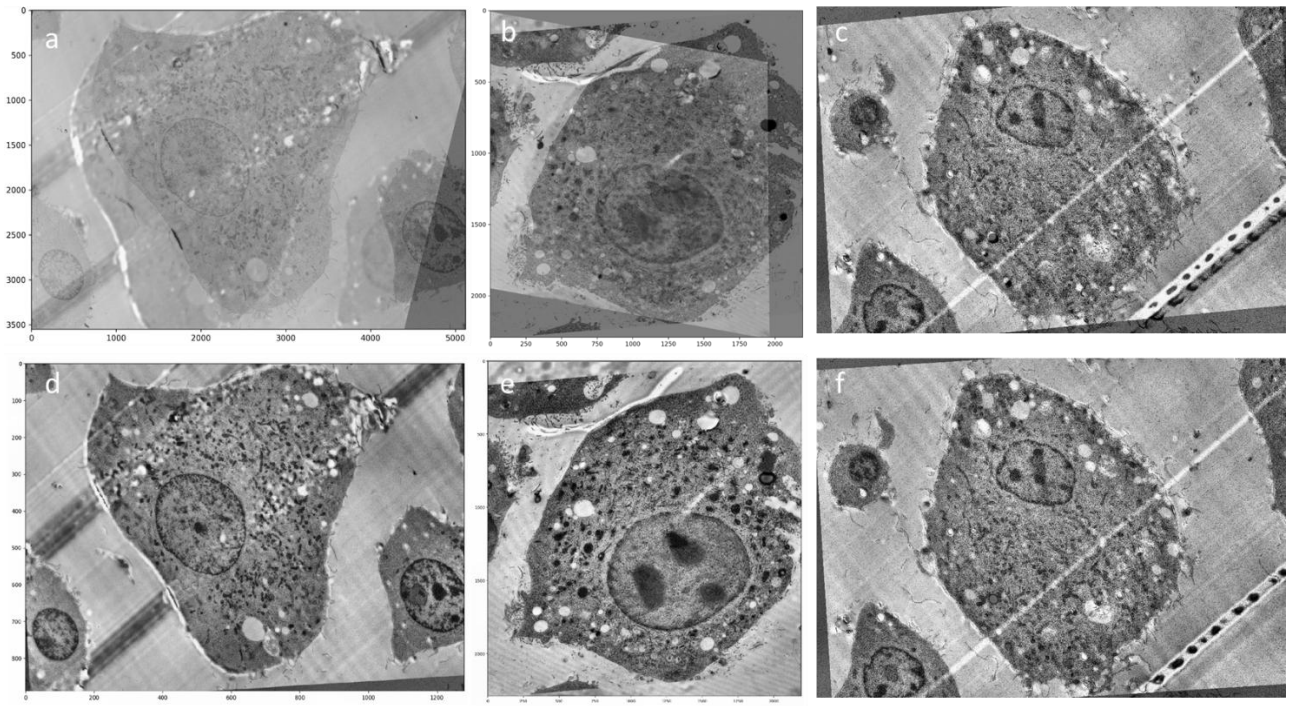


*Fig. 47: Effect of GPU overheating on different Ips displayed in the upper panel. For comparison, the lower panel shows the successful registration using a "cold" GPU.*

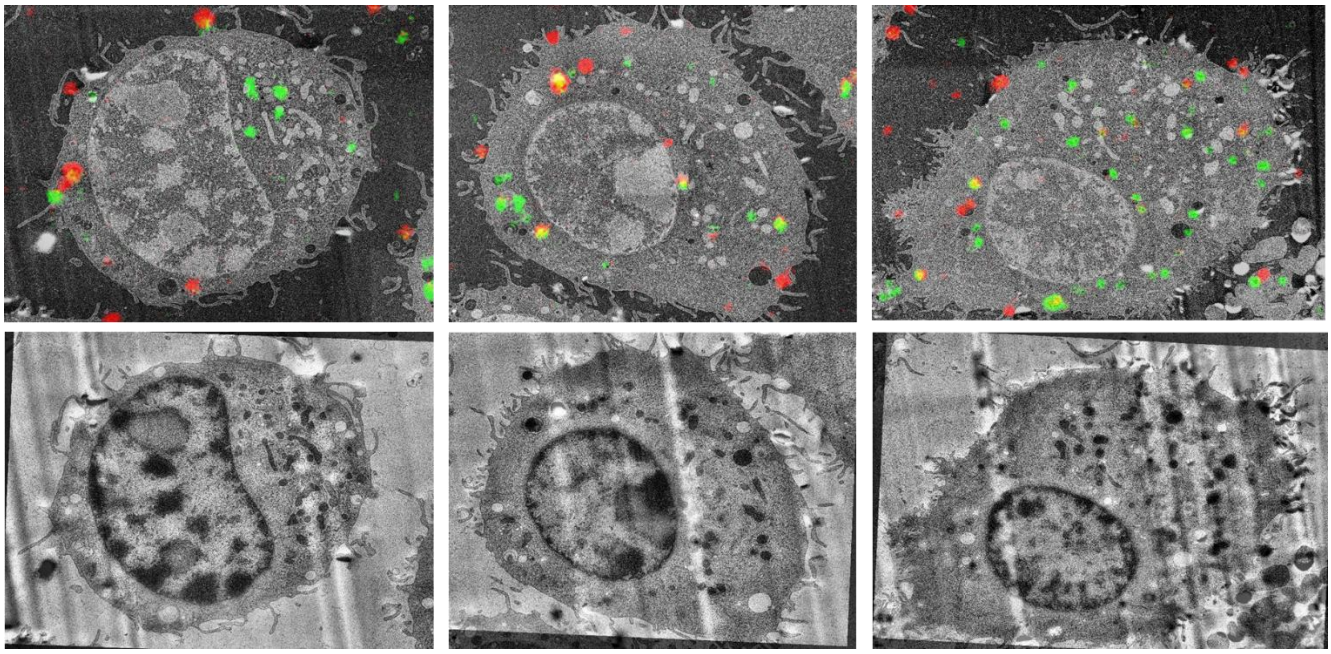

*Fig. 48: Dual color CLEM micrograph IR results for various datasets. The top row shows the CLEM micrographs with fluorescence multicolor channel and the lower row shows the corresponding overlay of the EM image with the reflected channel. This display method is the best way to visually evaluate the accuracy of the registration.*

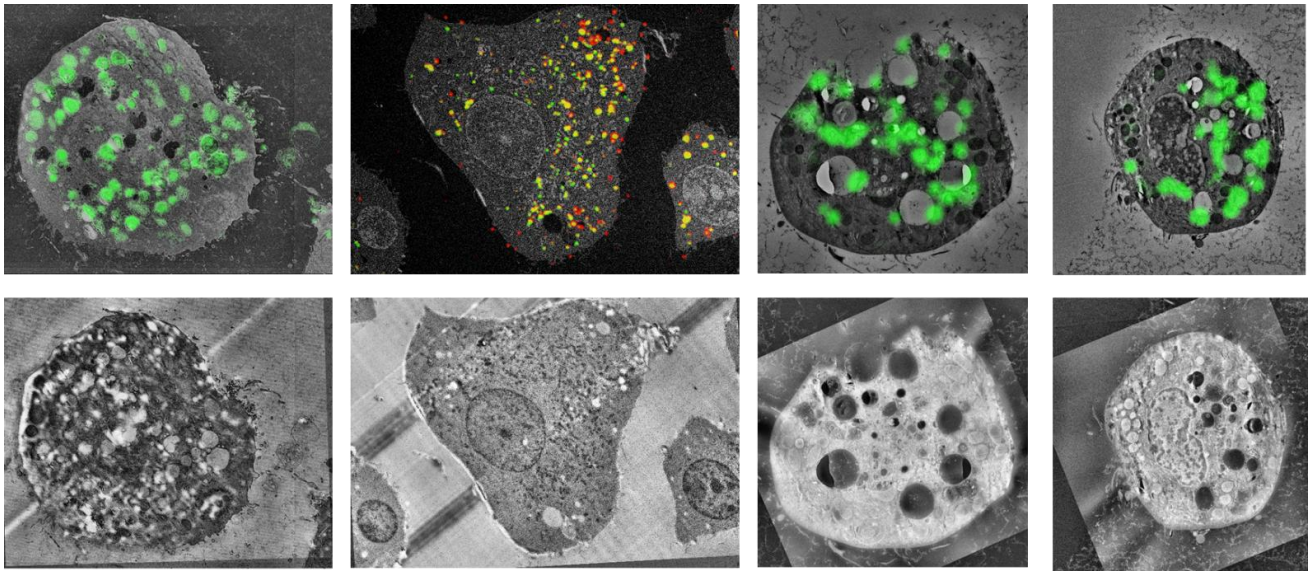

*Fig. 49: CLEM micrograph IR results obtained from different SEM & TEM datasets. The top row shows the CLEM micrographs with the fluorescence channel and the lower row shows the corresponding overlay of the EM image with the reflected channel.*

# Appendix II: Examples and references for the image pre-processing pipeline

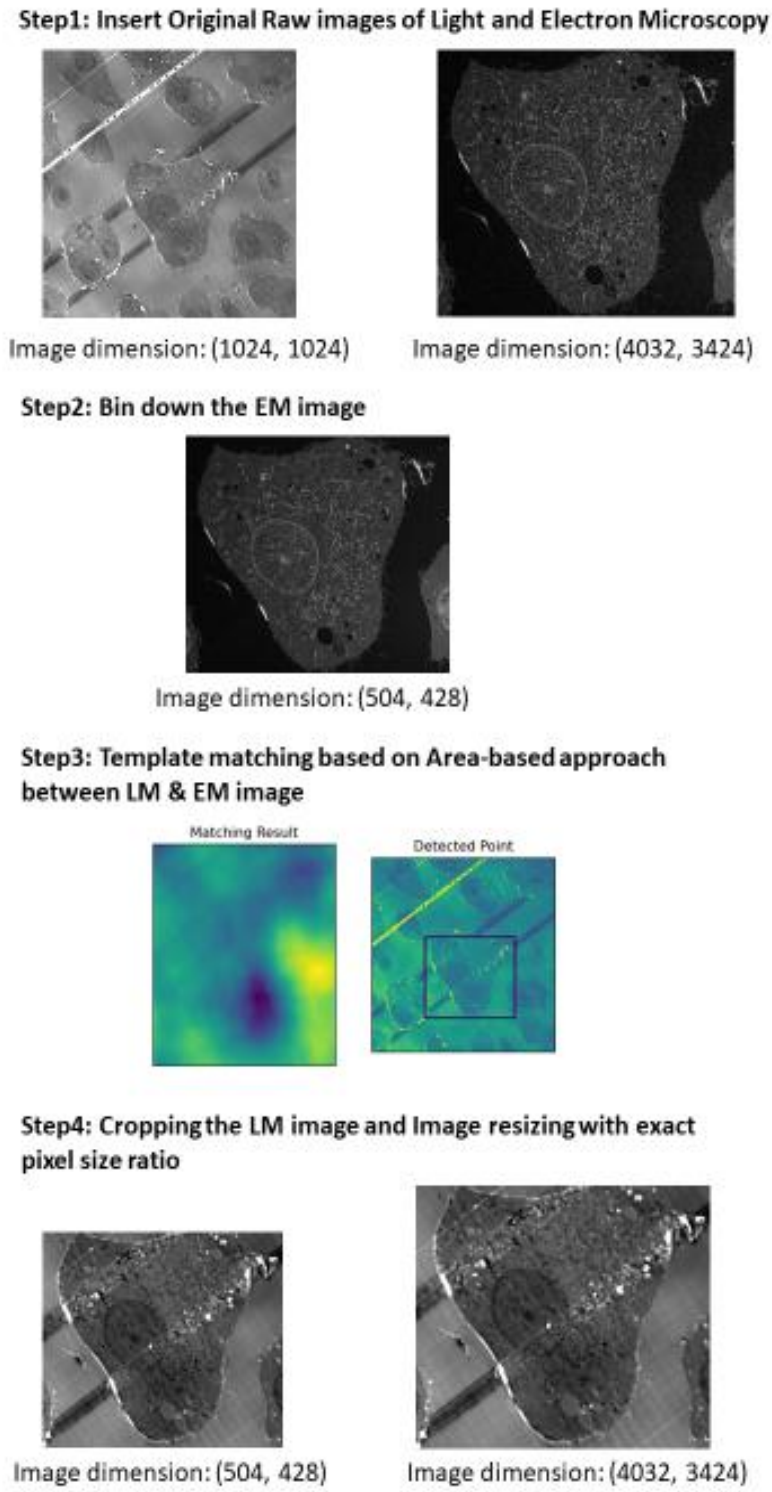


*Fig. 50: Weakly supervised image pre-processing workflow for the generation of multi-modal or CLEM image pairs. The aim is to extract the very same FoV from the LM image. For this, a TM approach is applied yielding the exact position within the LM image. Finally, this part of the LM image is cropped to yield an EM-LM image pair with matching FoV.*

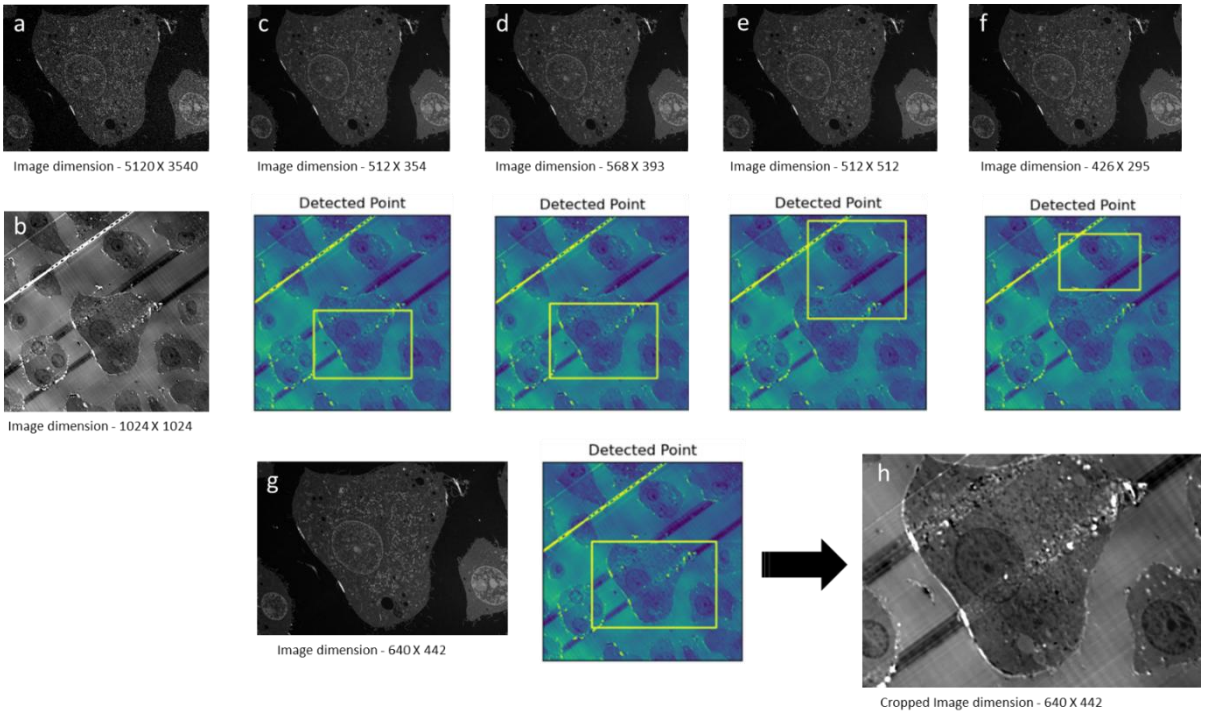


*Fig. 51: Detailed analysis of TM results with different templates, e.g. image dimensions, of the EM image. a and b represent the raw EM and LM images, c, d, e, and f represent false matching results due to mismatch in information in template for area-based matching. The PSR between the images is not matching, and g represents the best template (best PSR) for template matching and the cropped image is represented in h.*

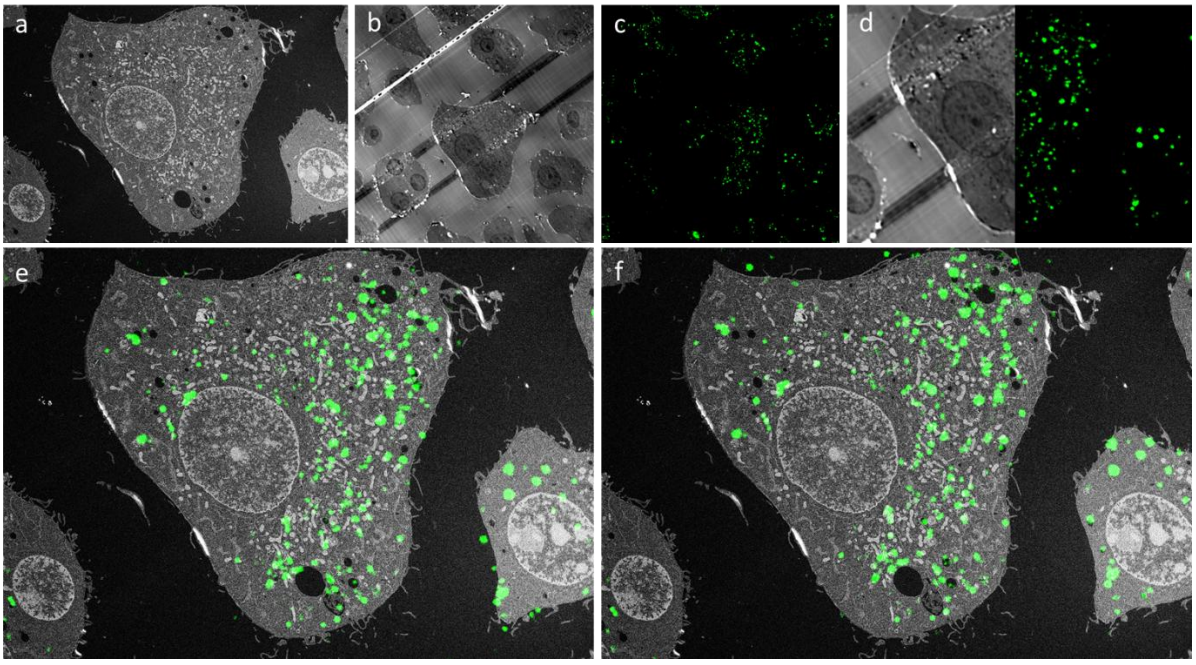


*Fig. 52: Comparison using the reflected and the fluorescence channel for IR. a) to d) show the raw image data used as input. In e) the software was optimized using the fluorescence channel and in f) the reflective channel. As from prior analysis the IR in e) using the FM channel only, has failed. This can be attributed to the fact, that the FM channel does not contain any smart features that match the EM image.*

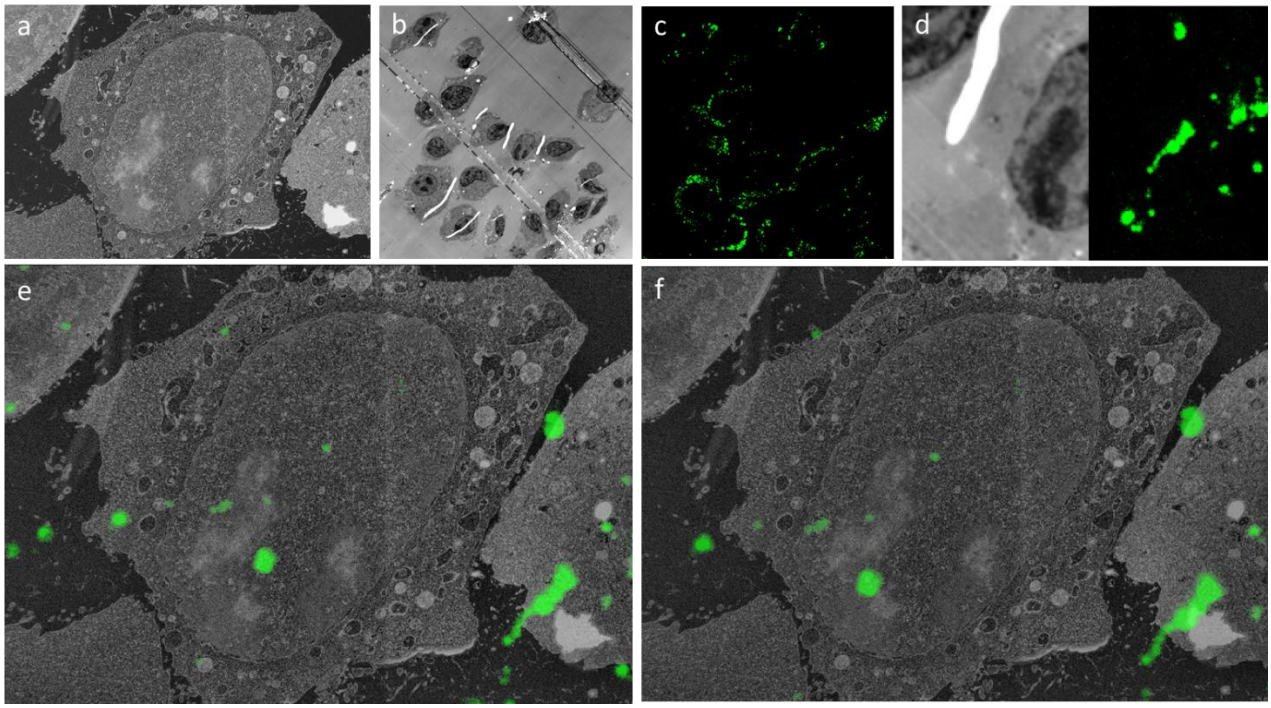


*Fig. 53: Another example as in Fig. 52 but for the dataset presented in the lower panel of Fig. 37. e) Here again, the IR fails when using the FM channel only.*

# Appendix III: Use of AI tools

Table a. List of AI tools / software I used in my thesis for writing and arranging the images.

| **AI tools** | **Used for** | **Why** | **When** |
|---|---|---|---|
| DeepL Translate | Translation between English to German and vice versa | German is not my native language | For abstract and other translations |
| DeepL Write | Rephrasing & reformulation of texts | Better readability | Throughout the entire work |
| ChatGPT & Google search | Searching the topic in depth with concept explained, finding research articles and up to date new/old information | Find relevant articles for my research, and fundamental key terms | Literature research and all relevant information |
| MS-Office based grammar & spell check | Rephrasing & reformulation of texts | Better readability, fix grammatical mistakes and spelling checks | Throughout the entire work |

I used ChatGPT and google with google scholar for searching the literature, understanding the meanings and describing different scientific terms. All the texts and pages are written using MS-Word and in-built spell check is used for typos, spellings and grammar. The formulation of sentences and paraphrasing is checked with DeepL and ChatGPT for polishing or arranging in better way. I present some examples how I use these tools.

All presented images are original micrographs generated directly by the algorithm, without any editing, cropping, or post-processing. They are arranged in the desired order and compiled using Microsoft PowerPoint.

Table b. Examples of using search engine and AI-based tools in this thesis.

| **Search Engines** | **Example texts or words** |
|---|---|
| ChatGPT & Google search (how do I do search) | Light microscopy, electron microscopy, machine learning, image registration. I used the words for understanding the meanings. |
| Rephrasing or polishing using ChatGPT | All the presented images are directly the micrographs obtained from the algorithm itself without editing, cropping or polishing. These images are arranged in desire order and assembled together using MS-PowerPoint. (Original text wrote by me)<br><br>All presented images are original micrographs generated directly by the algorithm, without any editing, cropping, or post-processing. They are arranged in the desired order and compiled using Microsoft PowerPoint. (Text polished using ChatGPT) |

# Appendix IV: Curriculum Vitae

**Daksh Daksh**
PhD Candidate, MPIP Mainz, Germany (daksh-daksh.github.io)
daksh@mpip-mainz.mpg.de +49 177 9228749 de.linkedin.com/in/ddaksh

---

Computer vision researcher specializing in machine and deep learning, generative AI, and super-resolution imaging. Focused on scalable and automated machine learning for science.

**EDUCATION**

**Max Planck Institute for Polymer Research,** Mainz, Germany
PhD Computer Science Oct 2021 – present
Advisor: Prof. Katharina Landfester & Dr. Ingo Lieberwirth

**National Forensic Sciences University (AIR – 159 in Computer Sciences),** Gandhinagar, India
MS - Forensic Nanotechnology (Nano-engineering) Jul 2014 – May 2016
Graduated in First class with Distinction

**Lovely Professional University,** Phagwara, India
B.Tech (Honors) Computer Science & Engineering Jul 2008 – May 2012
Graduated in First class with Distinction

**Experience**

**PhD Researcher & affiliated with CRC 1066, MPIP Mainz** 2021 – Present

- Working on automated image registration for correlative microscopy using deep learning.
- Developed Python tool for training free multi-modal image registration of light and electron microscopy and supervised image Pre-processing tool for generating the precised image-pair.
- Worked on deep Learning for image analysis and microscopy data analysis using Machine Learning e.g. for generation of super resolution microscopy images

**Web group Coordinator, Max Planck PhDnet, MPG** 01/2023 – 01/2025

- Development and maintenance of the PhDnet website and mailing lists.

**Research fellow in CSIE & IEO dept., NTNU, Taiwan** 05/2018 – 07/2020 Jointly worked on project with NVIDIA AI Tech, Taiwan

- Worked on optical defect detection using supervised deep learning for Digital Holography.
- Worked on data Augmentation using GAN i.e. AC-GAN and DC-GAN and for Deep Learning Based on Digital Holograms.
- Worked on deep Learning for image analysis and generating different holographic image datasets.

**Scientific Co-worker, University of Greifswald, Germany** 06/2017 – 02/2018

- Worked on computational modeling and simulations for bio-molecules.

**Setup configuration specialist, Aon Hewitt Pvt. Ltd., India (got selected among 5000 students & even before final semester)** 11/2012 – 07/2014

- Worked on IBM mainframes, Lotus notes, C++, ASP .net (C#), VB, and SQL

**Publications**

**TFUDL-CLEM: A Training-Free Unsupervised Deep Learning Registration Method for Correlative Light and Electron Microscopy.** D. Daksh, A. Kaltbeitzel, G. Glaßer, K. Landfester, I. Lieberwirth. Nature Communications Biology (Under Peer Review)

**A Weakly Supervised Pre-processing Pipeline for Multi-Modal Image Pair Generation for Image Registration.** D. Daksh, A. Kaltbeitzel, G. Glaßer, K. Landfester, I. Lieberwirth. (Submitted)

**Conferences**

**Multi-resolution Cross-modality Image Registration Using Unsupervised Deep Learning Approach.** D Daksh, A Kaltbeitzel, K Landfester, I Lieberwirth. Microscopy and Microanalysis: Microscopy Society of America 2023

**Unsupervised Deep Learning approach for image registration in Correlative Microscopy for the localization of Nanoparticles.** D Daksh, A Kaltbeitzel, G Glaßer, I Lieberwirth, K Landfester. Invited Speaker, BIO Web of Conferences. European Microscopy Congress 2024

**Correlative Microscopy Strategies for the Identification of Intracellular Nanoparticles and their Cellular Processing.** I Lieberwirth, S Han, A Kaltbeitzel, G Glaßer, D Daksh, K Landfester. Microscopy and Microanalysis: Microscopy Society of America 2024

**Patent**

**Multimodal Training-Free Registration Using Mutual Image Information.** Application submitted for patent & commercialization with Max Planck Innovations 2025.

| | |
|---|---|
| **Summer schools** | **Microscopy data analysis: machine learning and the BioImage Archive. EMBL Heidelberg, Germany** May 2022<br>**First EMBL Imaging centre Symposium: Enabling imaging across scales. EMBL Heidelberg, Germany** May 2022 Oral short talk<br>**SFB 1066 and 1278 Joint Summer school. Fulda, Germany** July 2022 Poster Presentation<br>**Joint Symposium of RMaP and SFB 1066. Mainz, Germany** July 2022 Poster Presentation<br>**EMBO workshop: from Cryo-EM to multi-scale modelling. EMBL Heidelberg, Germany** February 2023 Poster Presentation<br>**Deep Learning & computer vision (DLCV) school. Genova, Italy** June 2023<br>Selected among 1% with full scholarship to attend. Won the first prize for the most novel idea and best presentation.<br>**Joint Symposium of SFB1066 and TRR319. Mainz, Germany** April 2024 Poster Presentation<br>**Internal Symposium of SFB 1066. Mainz, Germany** July 2024 Poster Presentation<br>**International Symposium of SFB 1066. Mainz, Germany** January 2025 Poster Presentation |
| **Certificates** | **Optimizing writing strategies. MPIP Mainz, Germany** Feb. 2024<br>**German B1. In-house course in MPIP Mainz, Germany** Sept. 2024<br>**Python for HPC. MPCDF Garching, Germany** Nov. 2024<br>**Parallel Computing With Matlab. MPCDF Garching, Germany** Dec. 2024<br>**Workshop on AI and Research in MPG. Berlin, Germany** Dec. 2024 |
| **Skills** | **Programming Languages & Tools:** Python, PyTorch, Pandas, Anaconda navigator with different IDEs, Matlab, Napari, ImageJ, Bash, Git, LaTeX, TensorFlow, Scikit-learn, Scipy, Numpy<br>**Languages:** English (fluent), Hindi (native), German (B1), French (beginner) |

# Appendix V: List of Publications

## Publications

- TFUDL-CLEM: A Training-Free Unsupervised Deep Learning Registration Method for Correlative Light and Electron Microscopy. D. Daksh, A. Kaltbeitzel, G. Glaßer, K. Landfester, I. Lieberwirth. Nature Communications Biology. (under peer-review)

- A Weakly Supervised Pre-processing Pipeline for Multi-Modal Image Pair Generation for Image Registration. D. Daksh, A. Kaltbeitzel, G. Glaßer, K. Landfester, I. Lieberwirth. (Submitted)

## Conferences

- Multi-resolution Cross-modality Image Registration Using Unsupervised Deep Learning Approach. D. Daksh, A. Kaltbeitzel, K. Landfester, I. Lieberwirth. Microscopy and Microanalysis: Microscopy Society of America 2023.

- Unsupervised Deep Learning approach for image registration in Correlative Microscopy for the localization of Nanoparticles. D. Daksh, A. Kaltbeitzel, G. Glaßer, I. Lieberwirth, K. Landfester. Invited Speaker, BIO Web of Conferences. European Microscopy Congress 2024.

- Correlative Microscopy Strategies for the Identification of Intracellular Nanoparticles and their Cellular Processing. I. Lieberwirth, S. Han, A. Kaltbeitzel, G. Glaßer, D. Daksh, K. Landfester. Microscopy and Microanalysis: Microscopy Society of America 2024.