UNIVERSITY OF GENOVA

MASTER DEGREE IN ROBOTICS

Parameters optimization and image condition analysis for focus assessment with a machine learning proof-of-concept study

by

Mohammad Reza Haji Hosseini

Thesis submitted for the degree of *Master of Robotic*

October 25, 2023

Carolina Scandellari — Supervisor from Foresee Biosystems

Michele Dipalo — Supervisor from Foresee Biosystems

Carmine Recchiuto — Supervisor from Unige

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Mohammad Reza Haji Hosseini
October 2023

Acknowledgements

I wish to extend my sincere gratitude to all those who have been instrumental in supporting and guiding me on this transformative journey of completing my master's thesis.

First and foremost, I am deeply appreciative of my esteemed thesis supervisor, Carolina Scandellari. Her unwavering guidance, profound insights, and consistent encouragement have been invaluable to the development of my research and the refinement of this thesis's quality. Their expertise has truly been the cornerstone of my academic growth.

Equally deserving of recognition are the distinguished faculty members of the Robotic Engineering Department. Their adept teachings have not only laid a robust foundation for my comprehension but also equipped me with the essential tools to embark on this research odyssey. Their ardor for knowledge and dedication to education have been exceptionally motivating.

My heartfelt thanks extend to my mother and father, whose unyielding support and unwavering faith in my capabilities have been a wellspring of strength. Their ceaseless encouragement has propelled me through challenges and bolstered my resolve during moments of self-doubt.

Abstract

This master's thesis project records a period of internship conducted at Foresee Biosystems, a cutting-edge startup nurtured by the Italian Institute of Technology. The core mission of this startup revolves around the creation of IntraCell, an innovative apparatus that employs advanced LASER-based technology to assess the impacts of various substances on cardiac cells. Over the course of five months, the main goal of the internship was to concentrate on enhancing the autofocus algorithm for IntraCell—a critical element for achieving accurate cell observation and optimal cell optoporation at the correct focal point.

The research involved evaluating various methodologies, including the Sobel operator, for constructing a "Performance curve", a critical component for focus detection. Integration of the most accurate methods into Machine Learning algorithms, such as Support Vector Regression, significantly expanded the effective focus range. Additionally, a Convolutional Neural Network was integrated for image classification, substantially improving autonomous focus determination. The study also highlighted the significance of closed-loop control and applied fuzzy control strategies.

The autofocus procedure involves a series of tasks, including handling Z motor commands, ongoing image capture, image processing, creating performance curves, predicting performance curve trends, employing neural networks for image classification, and implementing fuzzy control.

This research highlights the seamless amalgamation of cutting-edge technologies and intricate machine learning methodologies. This synergy has culminated in the refinement of autofocus capabilities within the fields of computer vision, image processing, and robotics, marking a significant stride forward.

Table of contents

List of figures

List of tables

Nomenclature

Greek Symbols

Δ	Delta
ε	Epsilon
γ	Gamma
μ	Micro
∇	Nabla
∂	Partial derivative
π	Number Pi
σ	Small sigma
$\sum$	Capital sigma

Acronyms / Abbreviations

BCIs	Brain-computer interfaces
BPF	Band pass filter
C	Cost parameter
CL	Closed Loop
dB	Decibels
DFT	Discrete Fourier Transform
DPI	Dots Per Inch

DW Depht-wise convolution

ER Error Rate

FC Fully Connected

FFT Fast Fourier Transform

G Gradient magnitude

G_x Gradient X

G_y Gradient Y

GANs Generative Adversarial Networks

GHK The Goldman-Hodgkin-Kats Equation for the calculation of membrane potential for K+, Na+, and Cl- ions

GUI Graphical User Interface

HPF High pass filter

IDFT Convolutional Neural Network

IDFT Inverse Discrete Fourier Transform

L Laplacian

LPF Low pass filter

MCS Multi Channel Systems

MF Median filter

ML Machine Learning

NE The Nernst Equation for membrane voltage at equilibrium

OL Ohm's law for potential difference calculation

PW Point-Wise convolution

RBF Radial Basis Function

ReLU Rectified Linear Unit

RMS Root Mean Square

RMSE Root Mean Squared Error

SA Software Architecture

SMA Shape Memory Alloy

SNR Signal Noise Ratio

STD Standard Deviation

SVM Support Vector Machines

SVR Support Vector Regression

VoG Variance of Gradient

Introduction

For this master's thesis project, I did an internship at Foresee Biosystems, a high-tech start-up incubated by the Italian Institute of Technology. This start-up is currently working on a ground-breaking device, called IntraCell to assess the toxicological outcomes of a drug or chemical in in-vitro cardiac cells (Iachetta et al., 2021, 2022; Melle et al., 2020) based on a new patented LASER-based technology that allows simultaneous recordings of action potentials from hundreds of cardiomyocytes, offering unprecedented electrophysiological measurements with high quality, high parallelization, and extreme reliability. This laser technology is combined with the Multi-Channel Systems (MCS) microelectrode array (MEA) allowing for non-invasive evaluation of cardiac action potential over weeks.

The laser technique used to obtain the cells' signal is called Optoporation, while MEAs are used in electrophysiology to obtain extracellular recordings. The system is also composed of an optic microscope and a camera, which allow the user to witness what is happening in the cellular culture. The microscope, in conjunction with the laser, is governed by three motors along the x-y-z axes, enabling the precise movement of the optical and laser components across these three dimensions. Nevertheless, for precise focusing, it's essential to make adjustments along the Z-axis, necessitating additional Z-axis commands. If the sample is not on the right focus, the user cannot see it properly, and the optoporation has no effect on the cells. My main goal during this period was to provide a faster and more robust algorithm for the autofocus assessment of the system.

This thesis delves into the exploration of multiple techniques aimed at constructing an autonomous algorithm for achieving accurate focus. The primary focus of this research was to comprehensively evaluate five distinct methodologies for constructing a focus state representation curve. This curve, referred to as the *"Performance curve"*, was developed using the following five methods: Sobel filtering and correction, Canny edge detector filtering, Fast Fourier Transform, Noise processing, and Saturation analysis. The intention behind this approach was to identify critical points on the curve that potentially indicate optimal focus.

Notably, these techniques enable precise focus detection within a substantial range of ± 100 μm microns, surpassing the capabilities of mere superficial range assessments.

In the subsequent phases of development, only the two most accurate and robust methods were selected to be integrated into Machine Learning algorithms. Support Vector Regression (SVR) algorithms were chosen to enhance the precision of curve prediction, thereby rectifying the curve trends and accurately identifying maxima that correspond to focal points. This integration not only expanded the effective range but also extended it to approximately ± 200 μm.

To further amplify the algorithm's efficacy, a Convolutional Neural Network (CNN) was introduced, leveraging the capabilities of a pre-trained network, MobileNet V2. This CNN implementation, executed prior to constructing the performance curve and SVR algorithm, centered around image classification and labeling.

The incorporation of the pre-trained CNN dramatically heightened both the swiftness and accuracy of autonomous focus determination. While organizing classified images without a corresponding management algorithm may seem inconsequential, this research recognized the necessity of employing closed-loop control strategies. To this end, fuzzy control strategies were implemented, enabling the autonomous management of the focus procedure.

The automated focus process encompasses the coordinated management of the Z motor command, integration with the camera, continuous image acquisition, image processing, directional decision-making (ascending or descending), construction of the performance curve utilizing the aforementioned methods, performance curve predictions facilitated by SVR, image classification by the CNN, as well as the application of fuzzy control for fuzzification and defuzzification. Additionally, specialized fine-focus algorithms were developed to cater to finer intervals, thereby refining the focus accuracy.

This implementation has endowed the autofocus algorithm with the remarkable capability to achieve focus within an expansive range surpassing ± 300 μm.

This comprehensive endeavor spanned a span of five months within the Forsee Biosystems team, with invaluable collaboration opportunities with their engineers. Throughout this period, an extensive collection of datasets was amassed, including over 3000 images and the execution of more than 1000 experiments utilizing the IntraCell machine.

This thesis is organized into five distinct chapters, each contributing to a comprehensive understanding of the subject matter:

1. Foundations of Electrophysiology and Optoporation: The initial chapter serves as a foundational overview of electrophysiology, exploring methodologies for investigating

both intracellular and extracellular activities. Additionally, the concept of Optoporation is introduced and discussed within this context.

2. IntraCell Machine and Imaging Parameters: The second chapter explores the IntraCell machine, highlighting its imaging capabilities and key factors affecting image quality. It offers a concise examination of how these parameters impact imaging conditions, further clarifying their makeup and presenting an extensive overview of various methodologies.

3. Autofocus Algorithms and Development: Chapter three revolves around autofocus algorithms. This chapter not only underscores the necessity of autofocus within the IntraCell context but also outlines the strategies employed to engineer these autofocus algorithms. Furthermore, it intricately details the curve generation phase, the integration of Machine Learning, and the implementation of closed-loop control strategies.

4. Results and Comparative Analysis: The fourth chapter serves as a repository of outcomes derived from the methods devised within this thesis. Each method's results are meticulously documented and subsequently subjected to a comparative analysis, thereby justifying the rationale behind the selection of specific approaches.

5. Reflections and Future Directions: The final chapter provides insightful reflections on the outcomes attained and the objectives successfully realized throughout this endeavor. Additionally, this chapter offers guidance based on the obtained results and contemplates potential avenues for future enhancement and refinement.

In sum, this thesis is a comprehensive journey through the realms of electrophysiology, imaging technologies, autofocus algorithm development, and results analysis. It not only broadens the understanding of these domains but also sets the stage for potential future advancements in the field.

Chapter 1

Electrophysiology basics

1.1 Electrophysiology

Electrophysiology is a field of physiology that examines the electrical activity of living cells, tissues, and organs. It encompasses the measurement and analysis of electrical signals produced by biological systems. Its primary focus is on investigating the electrical characteristics and functions of excitable cells, including neurons, muscle cells, and cardiac cells. Electrophysiological methods enable researchers to comprehend electrical signals' generation, transmission, and regulation within biological systems. These methods entail utilizing specialized tools and equipment to gauge and document electrical signals, such as action potentials, membrane potentials, and ion currents (Tel, 2017).

Electrophysiology finds varied uses in neuroscience, cardiology, clinical diagnosis, prosthesis and BCIs, pharmacology and drug development, audiology, biomedical research, etc. It aids in comprehending neuronal activity, diagnosing cardiac arrhythmia, evaluating heart/brain function, developing prostheses and BCIs, assessing drug effects, diagnosing hearing disorders, and advancing medical research (Tel, 2017).

1.2 Electrophysiology principle

Electrophysiology is a branch of physiology that studies the electrical properties of biological tissues and cells. It involves measuring the electrical signals generated by living organisms, such as neurons, muscle cells, and cardiac cells, to understand their function and behavior. The principle of electrophysiology revolves around the use of specialized techniques and equipment to record and analyze electrical activities (Kouritas et al., 2008).

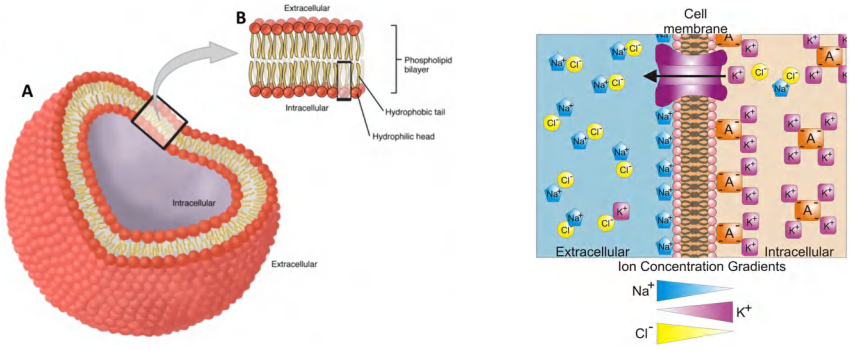
The basic principle of electrophysiology can be summarized as follows[Tab1.1]:

Feature	Description
Excitability	Excitability is the property of certain cells to respond to stimuli by generating electrical signals (action potentials). Neurons, muscle cells, and some endocrine cells are examples of excitable cells.
Membrane potential [Fig1.1b]	In excitable cells there is a voltage difference across the cell membrane known as the membrane potential. This potential is maintained by the unequal distribution of ions (charged particles) across the cell membrane, which creates an electrical gradient.
Ion channels [Fig1.1b]	Ion channels are proteins embedded in the cell membrane that allow the movement of specific ions (e.g., sodium, potassium, calcium, chloride) in and out of the cell. These channels play a crucial role in generating and regulating the membrane potential and are responsible for the electrical excitability of cells.
Action potential	An action potential is a brief, rapid change in the cell's membrane potential that occurs in response to a stimulus that exceeds a certain threshold. When this threshold is reached, voltage-gated ion channels open, leading to a cascade of ion movements and generating an electrical impulse that propagates along the cell membrane.

Table 1.1 Electrophysiology feature (Heckenlively and Arden, 2006)

Electrophysiology plays a crucial role in understanding the nervous system, cardiac function, muscle physiology, and various cellular processes. It has applications in neuroscience, cardiology, pharmacology, and other fields, contributing to advancements in medical research and treatments for various neurological and cardiovascular disorders.

In research laboratories, electrophysiology methods assess ion-channel activity within cell membranes under different electrical conditions [Fig 1.1]. These techniques aid in understanding cellular communication, synaptic transmission, and the mechanisms underlying various physiological processes (Kouritas et al., 2008).



(a) (b)

Figure 1.1 In the left image, A) The membrane completely encloses a normal cell; things and occurrences inside and outside the cell membrane are referred to as intracellular and extracellular, respectively. B) A more detailed image of the cell membrane, which is made up of two phospholipid layers with the heads on the outside and the tails on the inside. In the right image, Ion concentration gradients across the cell membrane. The image shows a cell membrane, the extracellular/intracellular regions, and a single ion channel protein in the membrane. K+ ions (purple square) are highly concentrated inside the cell and at low concentrations outside the cell, with the opposite being true for Na+ (blue pentagon) and Cl- (yellow circle). These gradients are outlined as triangles below the image. The orange square A is an anion, a negatively charged ion that positive ions are attracted to, making an object too big to pass the ion channel (Rocha, 2022). The following images are taken from https://www.photometrics.com/learn/electrophysiology/voltage-imaging

1.3 Fundamentals of electrical and ionic equilibrium in electrophysiology

The study of electrical and ionic equilibrium forms the foundation of electrophysiology, offering valuable insights into the behavior of electrical currents and ion movement across biological membranes as shown in the [Fig1.1b]. This field relies on fundamental laws and equations such as Ohm's Law (1.1), the Nernst Equation (1.2), and the Goldman-Hodgkin-Katz Equation (1.3) (Kouritas et al., 2008). Ohm's Law, Nernst Equation, and Goldman-Hodgkin-Katz Equation are vital tools in electrophysiology, aiding the comprehension of cellular activity. These equations help analyze electrical phenomena by relating voltage, current, and resistance. The Nernst Equation is crucial for determining ion equilibrium potentials like chloride $[Cl^{-}]$, sodium $[Na^{+}]$, and potassium $[k^{+}]$. It calculates the electrical driving force for each ion, influencing nerve signaling, muscle contraction, and cardiac rhythms. The Goldman-Hodgkin-Katz Equation, an extension, considers multiple ion species and permeabilities, explaining resting potentials, action potentials, and drug effects. This knowledge finds applications in medical diagnostics, pharmaceuticals, and fundamental biological research. In fact, IntraCell was purposefully designed to investigate both intracellular and extracellular activity, making these principles absolutely fundamental.

1.3.1 Ohm's Law

Ohm's Law (OL) in electrophysiology states that the current passing through a biological tissue or cell is directly proportional to the voltage applied across it and inversely proportional to its resistance. Mathematically, it can be expressed as:

$$OL: \quad I = \frac{\Delta V}{R} \tag{1.1}$$

V is the voltage across the conductor
I is the current flowing through the conductor
R is the resistance of the conductor

1.3.2 The Nernst Equation

The Nernst Equation (NE) is used to calculate the equilibrium potential of an ion across a biological membrane. It is commonly used in electrochemistry and neurophysiology to determine the membrane voltage at equilibrium for a specific ion. For a single-charged ion, the Nernst Equation can be written as:

$$NE: \quad E = \frac{RT}{zF} ln \frac{[C]_o}{[C]_i} \tag{1.2}$$

E is the potential (Nernst potential) for a given ion
R is the universal gas constant and is equal to 8.314 J/(K *mol)
T is the temperature in Kelvin
Z is the valence of the ionic species
F is the Faraday's constant
$[C]_o$ is the concentration of the ionic species C in the extracellular fluid.
$[C]_i$ is the concentration of the ionic species C in the intracellular fluid.

1.3.3 The Goldman-Hodgkin-Katz Equation

The Goldman-Hodgkin-Katz (GHK) Equation is an extension of the Nernst Equation(1.2) and is used to calculate the resting membrane potential of a cell that is permeable to multiple ions [Fig1.1b]. It takes into account the permeability and concentrations of different ions (Priyadarshini et al., 2022). The GHK Equation can be written as:

$$GHK: \quad V_m = \frac{RT}{F} \frac{p_k[k^+]_o + p_{Na}[Na^+]_o + p_{cl}[Cl^-]_o}{p_k[k^+]_i + p_{Na}[Na^+]_i + p_{cl}[Cl^-]_i} \tag{1.3}$$

o is the extracellular fluid
i is the intracellular fluid
p_K is the membrane permeability for K^+
p_{Na} is the relative membrane permeability of Na^+
p_{Cl} is the relative membrane permeability of Cl^-
$[K^+]$ is the concentration of K^+
$[Na^+]$ is the concentration of Na^+
$[Cl^-]$ is the concentration of Cl^-

1.4 Exploring intracellular and extracellular electrophysiology

Electrophysiology has had a profound impact on our understanding of the nervous system, resulting in significant advancements in fields such as neuroscience, neurology, and psychology. This multidisciplinary field encompasses a wide range of approaches, and two particularly prominent categories are intracellular and extracellular electrophysiology.

1.4.1 Intracellular Electrophysiology

Intracellular electrophysiology is a scientific technique used to study the electrical activity within cells. It involves the insertion of a tiny glass electrode into a cell to measure its membrane potential, ion currents, and other electrical properties. This technique provides valuable insights into the functioning of individual cells, including neurons and muscle cells, and helps researchers understand various physiological processes. By monitoring and manipulating the electrical signals at the cellular level, intracellular electrophysiology contributes to our understanding of complex biological phenomena and can lead to advancements in fields such as neuroscience, cardiology, and pharmacology (Kouritas et al., 2008).

1.4.2 Intracellular recordings: Voltage Clamp and Current Clamp

Voltage Clamp and Current Clamp are intracellular recording methods where the electrode penetrates the cell. Another configuration, known as Patch-Clamp [Fig1.2], offers an alternative approach for these types of recordings. In Patch-Clamp, a glass micro-pipette, serving as

the recording unit, is attached to the cell membrane. This technique causes less cell damage compared to conventional intracellular measurements, enabling more precise and accurate data collection.

Voltage Clamp

Hodgkin and Huxley utilized this technique to investigate ion channel behavior, such as refractory periods, activation, and inactivation time constants. They conducted voltage clamp experiments on the giant axon of a squid (Hodgkin and Huxley, 1952) to determine the functions governing channel opening and closing. By measuring sodium and potassium currents, they deduced the respective conductance, g_{Na} and g_k. Measuring conductance in this experiment was critical because it allowed them to investigate the behavior of ion channels in neurons. By determining the conductance of sodium and potassium ions, they could understand how these ion channels opened and closed in response to changes in voltage. This information helped them deduce activation and inactivation time constants for the channels, which were essential for creating mathematical models of channel kinetics. Additionally, measuring conductance provided insights into refractory periods, which are vital for understanding nerve impulse propagation.

Currently, voltage clamp experiments remain valuable for assessing changes in current across cellular membranes. These experiments involve measuring the potential difference between two electrodes connected to a voltmeter: one inside and the other outside the axon. The measured voltage, V_m, is then compared to a desired reference value (command signal). A third electrode injects a current whenever there's a difference between V_m and the desired value, maintaining the membrane potential at the desired level through a feedback system with high gain. This setup keeps V_m practically constant even as sodium and potassium conductance fluctuate. The injected current, measurable with an ammeter, corresponds to the current flowing through sodium and potassium channels (since V_m is constant, the injected current must equal the ionic species' current).

Current Clamp

The Current Clamp is a complementary technique to the Voltage Clamp, where the current is controlled and the membrane potential is measured. In this method, a series of rectangular current pulses are applied to the cell using an intracellular electrode. By knowing the resistances of both the intracellular and extracellular electrodes, it becomes feasible to assess the membrane resistance and, consequently, its voltage using Ohm's law(1.1).

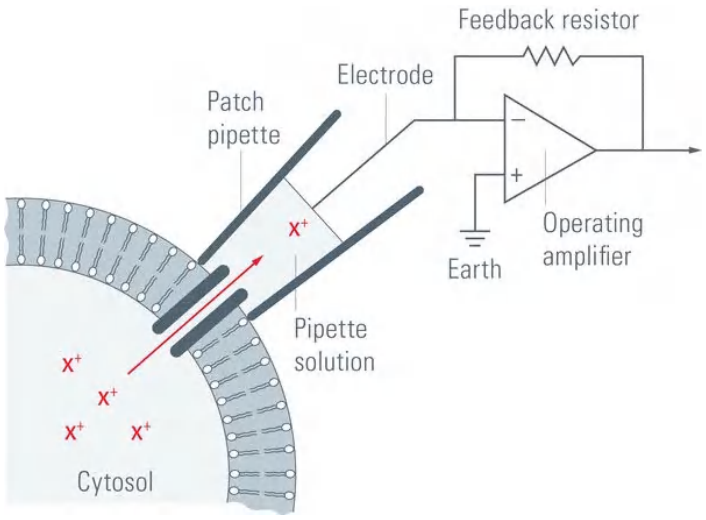


Figure 1.2 In the current-clamp experiment with patch-clamp configuration, a micro-pipette is used to electrically isolate a portion of the cell membrane. The micro-pipette contains a solution with an electrode connected to an amplifier.The following image is taken from https://www.leica-microsystems.com/science-lab/life-science/the-patch-clamp-technique/

Both techniques offer distinct advantages and disadvantages, as summarized in the [Tab1.2], making them well-suited for various types of experiments.

Method	Advantages	Disadvantages
Voltage Clamp	Precise control of membrane potential	Invasive, potentially damaging the cell
	Directly quantifying ion channel conductance and kinetics	Conducted under non-physiological conditions
	High temporal resolution for studying fast electrical events	Limited information about overall cell behavior
Current Clamp	Preserves more natural physiological conditions	Limited information about specific ion channels
	Offers insights into cellular excitability and action potentials	Less precision compared to voltage clamp
	Enables synaptic input simulation to study cell responses	Ensuring stable current injections

Table 1.2 Advantages and Disadvantages of Extracellular Recordings using Voltage and Current Clamp techniques (Inagaki and Nagai, 2016)

1.4.3 Extracellular electrophysiology

Extracellular electrophysiology is a scientific technique used to study the electrical activity of cells by measuring the electrical signals outside the cells. Unlike intracellular electrophysiology, which involves inserting an electrode into a cell, extracellular electrophysiology records the electrical activity from the extracellular space surrounding the cells. This technique is

commonly used to study the activity of populations of cells and provides information about the patterns and synchronization of their electrical signals (Buzsáki et al., 2012).

1.4.4 Extracellular recordings: MEAs and MEMS

Extracellular electrophysiology measurements are conducted using either Micro-Electrode Arrays (MEAs) or Micro ElectroMechanical Systems (MEMS).

MEMS

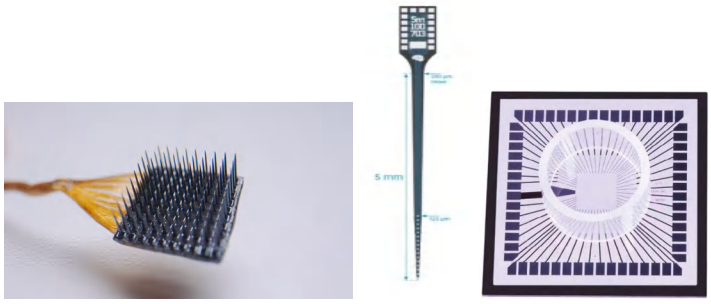
Two main MEMS devices are noteworthy: the Michigan probe and the Utah array [Fig 1.3a]. The former has a higher sensor density and features electrodes with their sensitive part extending along the entire length, not just at the tip. The Utah arrays are three-dimensional matrices of 100 silicon electrodes arranged in a 10x10 configuration, with their sensitive part located only at the tip. These devices are intra-cortical, meaning they penetrate the cerebral cortex, which raises concerns about long-term implant conditions that may cause a strong immune response.

MEAs

MEAs are devices containing hundreds of micro-electrodes used for recording or administering electrical signals. Different types of MEAs are used depending on whether the signals are recorded in vitro [Fig 1.3b] or in vivo [Fig. 1.3c]. In in-vitro applications [Fig 1.3c], these arrays are utilized for cellular cultures or brain tissue slices. The number of electrodes varies depending on the specific application, ranging from models with 60 electrodes to others with 120 or even 256 electrodes. In the last decade, MEAs for in-vitro applications, combined with CMOS circuitry, have emerged, allowing simultaneous recordings from over 4000 electrodes and enabling the discrimination of individual neuronal units (Stringer et al., 2018). These devices also facilitate the electrical stimulation of cells and provide high spatial and temporal resolution for observing current propagation.

Both techniques, as summarized in [Tab1.3], offer distinct advantages and disadvantages, making them well-suited for various types of experiments in the study of neural activity.

MEA technology facilitates the recording of extracellular information but cannot monitor intracellular activities, such as ion channel activity. However, various methods exist to study intracellular processes using MEAs. Notably, Optoporation[Sec1.5] enables access to intracellular information through lasers. The articles (Iachetta et al., 2021, 2022; Melle



(a) (b) (c)

Figure 1.3 Different types of recording devices: (a) Is a Utah array with 100 silicon electrodes. (b) Is an in-vivo MEAs with high-density NeuroNexus electrodes at the tip. (c) Is an in-vitro MEAs for cellular culture recordings. The following images are taken from https://blackrockneurotech.com/our-tech/

Method	Advantages	Disadvantages
MEAs	High spatial resolution, revealing neural network interactions	Poor depth resolution, challenging single-unit isolation
	Suitable for long-term recordings without cell damage	Lower sensitivity to weaker neural signals
MEMS	Integrated microfluidics and sensors, precise control	Invasive, causes tissue damage and immune response
	High-density arrays, detailed neural activity mapping	Complex and costly fabrication
	Miniaturization allows implantation in small organisms	Small electrode size, increased noise, and artifacts

Table 1.3 Advantages and Disadvantages of Extracellular Recordings using MEAs and MEMS (Tanwar et al., 2022)

et al., 2020), comprehensively explain the reasons behind choosing this method and provide detailed insights into how it is applied with MEAs to study activity on both sides of the cell.

1.5 Optoporation

Cell optoporation, also known as optoporation or optical poration, represents a cutting-edge and highly sophisticated technique that revolutionizes the introduction of foreign substances into cells with exceptional precision and control. By utilizing light-based methods, this approach creates temporary pores in the cell membrane, thereby allowing the targeted materials, such as DNA, RNA, proteins, drugs, or nanoparticles, to enter the cells (Spira et al., 2019). The procedure involves several distinct steps, each playing a crucial role in the overall success of the process:

1. Preparation: To initiate the optoporation process, cells of interest are first treated with photosensitizing agents. These specialized agents selectively accumulate within the cell membrane, priming the cells for subsequent light exposure.

2. Light exposure: The key to optoporation lies in the precise and focused application of a laser or intense light source onto the specific cells containing the photosensitizing agents. This controlled light exposure is critical in achieving accurate targeting.

3. Pore formation: Activating photosensitizing agents by the light source triggers the generation of reactive oxygen species (ROS) or other forms of energy. These ROS act as cellular triggers, inducing transient pores in the cell membrane.

4. Substance delivery: The temporary pores, created by the ROS, facilitate the passage of foreign substances through the cell membrane, allowing them to enter the cell's interior with utmost efficiency.

1.5.1 Cell optoporation on MEAs

Cellular optoporation on microelectrode arrays involves utilizing laser-induced optoporation to introduce molecules into cells cultured on microelectrode arrays (MEAs). In this technique, a laser beam precisely targets a selected cell on the MEA, and a brief pulse of light is applied to create a temporary pore in the cell membrane. This pore allows for the introduction of various substances, including drugs and genetic material, into the cell or for intracellular molecule detection.

In our specific application (Iachetta et al., 2021, 2022; Melle et al., 2020), it employs cell optoporation on MEAs for recording action potentials through MEA electrodes[Fig1.4]. Notably, we do not introduce any molecules or drugs into the cells. This distinguishes our approach, which focuses on the precise measurement of intracellular electrophysiological signals using MEAs and optoporation technology. Foresee Biosystems specializes in this unique capability for accurate signal detection within cells. The advantages of this kind of study are presented in the [Tab1.4].

Although cell optoporation on MEAs provides valuable advantages, as previously mentioned in [Tab1.4], it also presents challenges in cell specificity, phototoxicity risks, optogenetic artifacts, declining long-term stability, potential electrical interference, and increased experimental complexity (Davis et al., 2013).

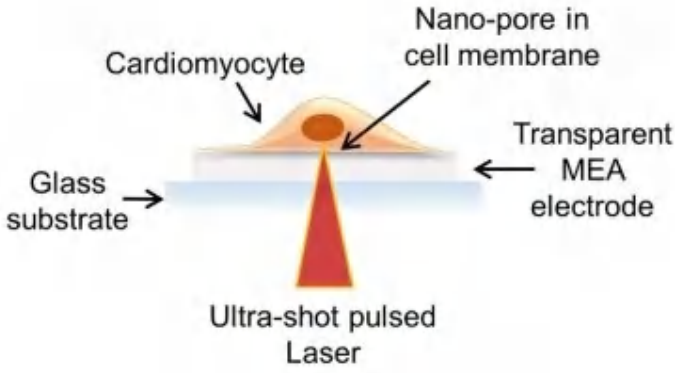


Figure 1.4 A cardiomyocyte cell was positioned on a transparent MEA, and an ultra-short pulsed laser was directed from beneath the sample to target nanopores on the cell membrane. This enables the study of real-time action potential generation by the cells, with the added potential of introducing chemicals like drugs to investigate their effects (Iachetta et al., 2022). The following image is captured from the same cited article.

Advantage	Description
Spatial precision	Laser-induced optoporation enables precise targeting of individual cells on MEAs, delivering chemicals to specific cells while sparing surrounding ones. This spatial accuracy is crucial for studying the impact of certain chemicals on individual cells in a population.
Temporal control	Provides precise temporal control for molecule delivery into cells. This control is valuable for studying cellular responses to specific molecules at precise time points.
Non-invasive	Non-invasive method for molecule delivery into cells, minimizing cell damage or interference with normal function. This non-invasiveness is crucial for preserving the viability and functionality of delicate cells, such as neuronal or cardiac cells commonly studied using MEAs.
Versatility	Allows delivery of a wide range of molecules into cells, including drugs, dyes, genetic material, and more. This versatility allows researchers to explore cellular processes and study the effects of different molecules on cell behavior.
Compatibility with MEA systems	Integration with MEA systems enables researchers to study cellular electrophysiology, manipulate the intracellular environment, and understand the interplay between molecular signaling and electrical activity in cells.

Table 1.4 Advantages of Cell optoporation (Davis et al., 2013)

Chapter 2

Advancing cell optoporation and optical imaging with IntraCell

2.1 IntraCell

The study of cellular processes lies at the core of various scientific disciplines, from cell biology to neuroscience. Advancements in imaging technologies and optoporation techniques have opened new frontiers for understanding cellular behavior and interactions within complex biological systems.

IntraCell [Fig2.1] is an advanced and versatile integrated video microscope designed to revolutionize optical imaging and electrophysiological signal recording, incorporating an advanced 3-axis motorized stage and an optical set-up, comprising a powerful 20X objective lens and a sophisticated black & white camera, ensuring real-time, high-resolution imaging of samples with exceptional clarity.

One of the standout features of IntraCell is its cutting-edge laser technology, which plays a crucial role in enabling targeted cell optoporation. This groundbreaking technique holds tremendous promise in manipulating cellular activity with unparalleled precision. The precise movements of the motorized stages, as well as the seamless control of the laser source, are orchestrated by a sophisticated embedded microcontroller. This microcontroller utilizes efficient serial communication protocols to seamlessly interface and communicate with the computer, ensuring smooth and accurate coordination of all system components. Researchers and scientists can now perform cell optoporation with ease and accuracy, paving the way for exciting discoveries in various fields, including neuroscience, cell biology, and beyond.

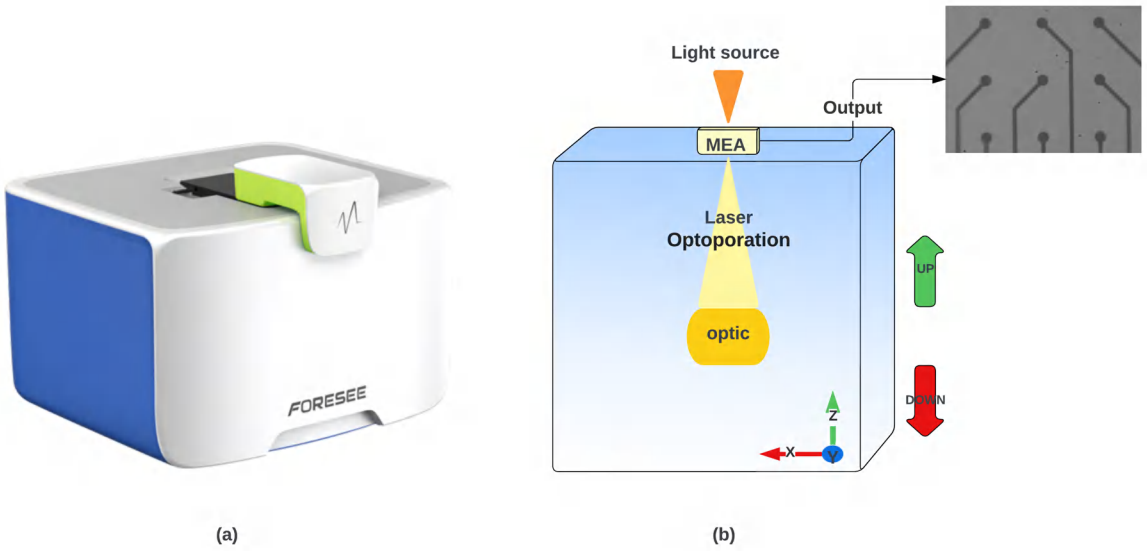


Figure 2.1 Image (a), the final product crafted by Forsee Biosystem. Image (b), is an informative sketch that vividly demonstrates the Z-motor's movements, crucial in facilitating focus processing. The altitude range spans from -3300 to +3300[μm], and it has been divided into 22 steps of 300[μm]each.

2.2 IntraCell imaging

Microscopy is a complex and vital process that magnifies small cells to create precise visual representations, revealing intricate details at a microscopic level. Crucially, images captured while the motor is in motion along the Z-axis carry essential information for autofocus algorithms.

2.2.1 Key aspects of image characteristics

The Intracell captures images at a resolution of 1280x960 pixels and a Pixel Density of 96 DPI (Dots Per Inch) and 24 bits. These images are produced using a combination of RGB channels, but the output is presented in grayscale (Black&White). The Intracell camera offers a range of parameters that allow users to modify imaging conditions, including Brightness, Contrast, Exposure, and Light.

In IntraCell imaging, each parameter plays a crucial and precise role in altering the image and resulting in significant improvements. When manipulated correctly, these parameters can transform an ordinary image into a more informative one. Here's a breakdown of their roles:

Brightness

Brightness refers to the overall intensity or luminance of the image. It determines how light or dark the image appears. Adjusting the brightness can make the image clearer and more visible, especially when dealing with low-contrast or dimly lit samples. Increasing brightness can make the image lighter while decreasing it will make it darker(Plummer et al., 1999).

Contrast

Contrast is the difference in intensity between the brightest and darkest areas of the image. A higher contrast enhances the visibility of details and sharpens the boundaries between different structures in the sample. Adjusting the contrast can help in revealing subtle features and improving the overall clarity of the image (Plummer et al., 1999).

Exposure

Exposure is the amount of light that reaches the optic sensor during image acquisition. Proper exposure is essential for capturing a clear and well-balanced image. Overexposure can result in washed-out and excessively bright areas, while underexposure may lead to dark and indistinct images.

Light

In IntraCell, the light source plays a critical role in illuminating the sample for visualization. Proper lighting is essential, as both insufficient and excessive light can result in poor visibility. The light sources are strategically positioned on top of the MEAs to effectively illuminate the sample, according to the information presented in [Fig2.1].

Optimizing these parameters is vital for obtaining high-quality images in IntraCell. It ensures that the essential details of the sample are visible and enhances the overall image interpretation and analysis. Adjusting brightness, contrast, and exposure appropriately can lead to better results in research, diagnosis, and various other applications in IntraCell.

2.2.2 Adapting imaging parameters

Parameters alterations wield substantial influence over image conditions, as evidenced by the images below [Fig2.2,2.3,2.4,2.5]. Notably, these alterations induce dynamic shifts in brightness, contrast, exposure, and light within the illustration shown in the below figures.

In the following, the potential result differences modifying brightness to its highest [Fig2.2a] and lowest [Fig2.2b] levels are described.

Maximum Brightness

Overexposure: Increasing the brightness to the maximum level can cause overexposure of the image. This means that the brightest areas of the image become too white or washed out, making it difficult to distinguish details in those regions.

Loss of Contrast: High brightness levels can lead to a loss of contrast between different structures or components in the sample. This can result in a lack of distinction between objects, making it harder to observe specific features.

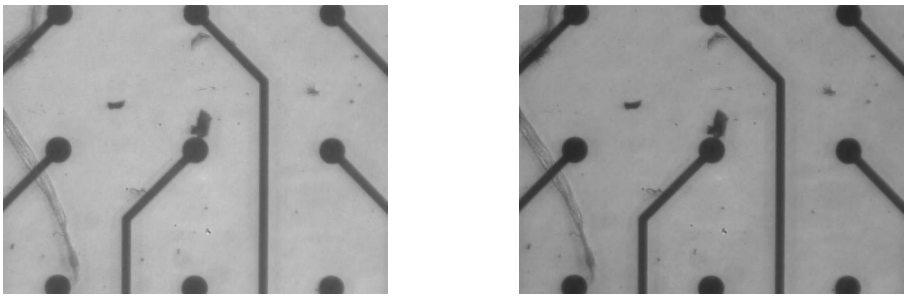
Loss of Information: If the image is overexposed, subtle details and nuances in the sample can be lost due to the saturation of pixel values.

Minimum Brightness

Underexposure: Decreasing the brightness to the minimum level can cause underexposure of the image. This means that the darker areas of the image become too dark, leading to a loss of details in those regions.

Diminished Visibility: When the image is underexposed, it becomes harder to see finer details and structures, especially those with low contrast.

Noise and Grain: decreasing brightness during processing to compensate for underexposure can result in an increase in image noise or graininess, which degrades the overall quality of the image.



(a) Brightness max (b) Brightness min

Figure 2.2 Brightness comparison

The same analysis is repeated for the other three parameters, i.e., contrast [Fig2.3a, 2.3b], exposure [Fig2.4a, 2.4b], and light intensity [Fig2.5a, 2.5b]: .

Maximum Contrast

Enhanced Difference: Increasing the contrast to its maximum level enhances the differentiation between light and dark areas in the image. This leads to a stronger distinction between features and structures, making them stand out more prominently.

Sharper Details: High contrast settings can sharpen the edges of objects, making finer details more apparent and improving the overall clarity of the image.

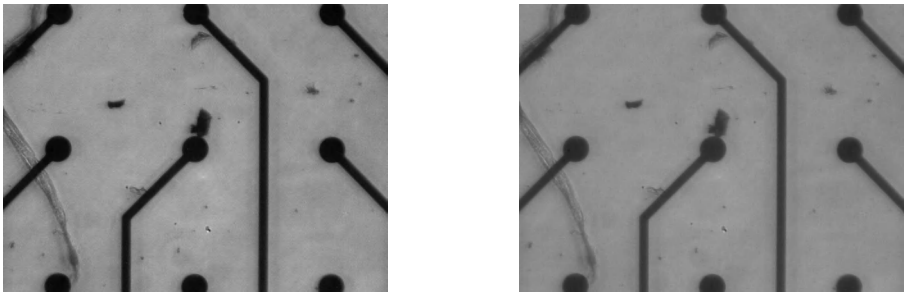
Risk of Artifact: However, pushing the contrast to the maximum might also lead to the amplification of noise and artifacts in the image, particularly in low-contrast regions. This can distort the image and make it less faithful to the actual sample.

Minimum Contrast

Reduced Distinction: Decreasing the contrast to its minimum level diminishes the difference between light and dark areas. As a result, the image may appear more uniform and less distinct in terms of its features and structures.

Loss of Details: Low contrast settings can lead to the suppression of subtle details and finer structures within the image. This can result in a lack of clarity and difficulties in observing specific characteristics.

Flattened Appearance: Images with very low contrast can have a flat and uninteresting appearance, as the visual depth and texture of the sample may be compromised.



(a) Contrast max (b) Contrast min

Figure 2.3 Contrast comparison

Maximum Exposure

Overexposed Highlights: Increasing the exposure to its maximum level can lead to overexposed highlights, where the brightest areas of the image become excessively bright or even white. This results in a loss of details in these overexposed regions.

Loss of Contrast: Overexposure often results in a loss of contrast, as the extreme brightness can cause adjacent structures to blend together, making it difficult to distinguish between them.

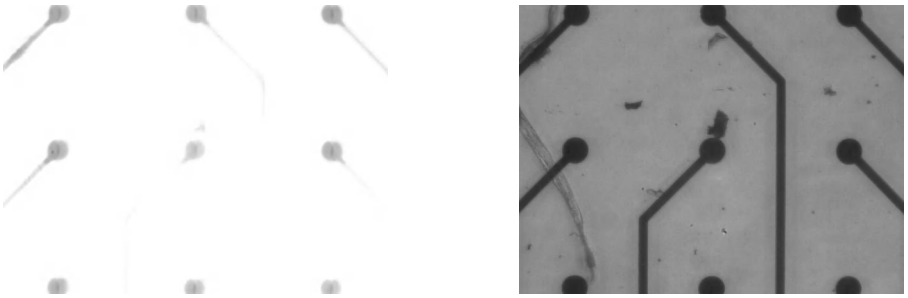
Risk of Saturation: Pixel values might saturate at the brightest points, causing a loss of information. This can lead to a lack of detail and nuances, particularly in areas that are naturally bright or reflective.

Minimum Exposure

Underexposed Shadows: Decreasing the exposure to its minimum level can result in underexposed shadows, where the darkest areas of the image become very dark, making it hard to discern details within them.

Diminished Visibility: Underexposed images lack the necessary illumination to reveal fine structures and features, leading to reduced visibility and clarity.

Increased Noise: When underexposed images are brightened during post-processing, noise, and artifacts can become more apparent, degrading the overall quality of the image.



(a) Exposure max (b) Exposure min

Figure 2.4 Exposure comparison

Maximum Light Intensity

Overexposure: Increasing the light intensity to the maximum level can lead to overexposure in the image. Bright areas may become overly saturated, causing details to be lost in those regions.

Loss of Contrast: High light intensity can lead to reduced contrast between different structures in the sample. This can result in a lack of differentiation between features, making it challenging to distinguish important details.

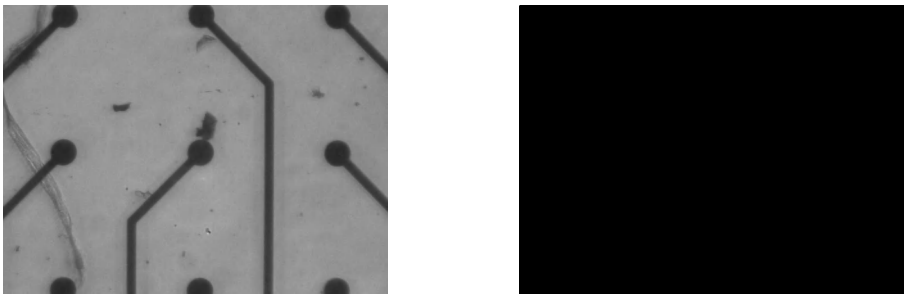
Glare and Reflections: Extremely bright regions can lead to glare, reflections, or even unwanted artifacts in the image, further obscuring details.

Minimum Light Intensity

Underexposure: Decreasing the light intensity to the minimum level can result in underexposure, where dark areas of the image lack sufficient illumination. This can cause a loss of details in those shadowed regions.

Diminished Visibility: Images captured with low light intensity may lack the necessary illumination to reveal finer structures and features, resulting in reduced visibility and clarity.

Increased Noise: Compensating for underexposure during post-processing by increasing brightness can amplify noise and introduce unwanted artifacts, degrading image quality.



(a) Light max (b) Light min

Figure 2.5 Light comparison

Through the IntraCell interface, users can tweak parameters to achieve diverse visual outcomes. As each parameter's optimal value depends on the analysis type, there's no fixed formula. Nevertheless, establishing limits helps avert artifacts. Altered parameters can influence Autofocus algorithms, which hinge on sharp image detection. Chapter 3 delves into comprehensive explanations of pre-processing techniques to counter these challenges.

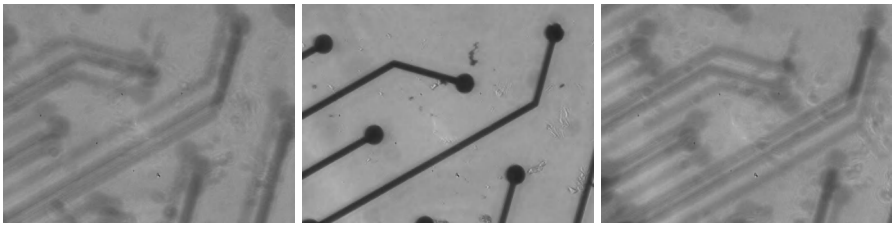
2.2.3 Variable distance imaging

Here, I present a series of image captures taken from various distances. In the accompanying [Fig2.6b], a designated reference point is established along the Z-axis. This reference point holds significance due to the absence of coordination along the Z-axis. To establish a quantitative baseline, an image of exceptional quality is selected as the reference, with an assigned value of zero μm.

It's important to note that this reference point may undergo adjustments based on factors such as variations in the thickness of MEAs, sample thicknesses under analysis, accurate positioning of the MEA, and more.

Within this context, two additional images are captured at distances of both +300 [Fig2.6a] and -300 [Fig2.6c] μm from the initial starting point along the Z-axis. This deliberate

variation carries substantial implications, particularly within IntraCell where the light source [Fig2.1] is positioned from above. Vertical movements influence the amount of recorded light significantly. This impact remains relevant even if the illumination intensity is maintained consistently throughout the analysis.



(a) from +300[μm] (b) from 0[μm] (c) from -300[μm]

Figure 2.6

These observations show how important it is to consider the up-and-down movements and how they affect the way light is captured. This is really crucial when dealing with Autofocus algorithms because it emphasizes the changing nature of the analysis.

2.3 Autofocus

Autofocus in microscopy refers to a technique used to automatically adjust the focus of a microscope's objective lens in order to obtain clear and sharp images of a specimen. Microscopy involves observing objects or models at a tiny scale, often requiring precise focus to capture detailed information.

Traditional microscopes require manual adjustment of the focus knob to achieve the clearest image of a specimen. However, as specimens can have irregular surfaces or dimensions, maintaining optimal focus can be challenging, especially when working with live samples or dynamic processes (Firestone et al., 1991).

In IntraCell, autofocus involves continuous movement along the Z-axis while capturing images, followed by image analysis to identify frames with exceptional quality. [Fig2.7] shows an image taken at different distances, with the motor's initial position at an unspecified distance from the focal point. At the beginning, the entire optical system starts moving and continues until the best position for the focus is reached. [Sec3.2.1] provides a detailed explanation of the Z-motor control method employed.

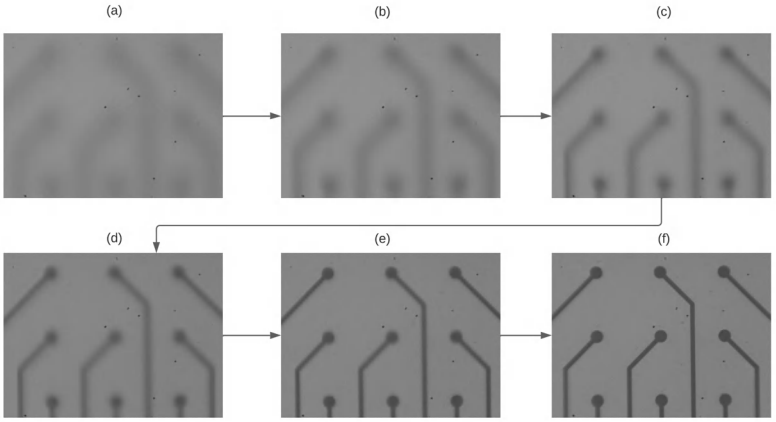


Figure 2.7 Starting at level (a), the motor undergoes successive movements in the correct direction, with each step enhancing the imaging conditions. The motor progresses until it reaches a level (f). Upon reaching level (f), an image with an optimal focal point is obtained through autofocus algorithms. The algorithm then designates this point as the reference zero μm.

2.3.1 State of the art

Autofocus technology automates this process by using various methods to determine the optimal focal plane without manual intervention. Here are several prevalent techniques employed within autofocus systems, as exemplified in the [Tab2.1].

Autofocus Method	**Source Article(s)**
Contrast-based autofocus	Autofocusing in digital holographic phase contrast microscopy on pure phase objects for live cell imaging (Langehanenberg et al., 2008)
Phase-contrast autofocus	Comparison of phase-contrast and fluorescence digital autofocus for scanning microscopy (Price and Gough, 1994)
Intensity-based autofocus	Intensity-based holographic imaging via space-domain kramers–kronig relations (Baek and Park, 2021; Luo et al., 2015)
Depth from focus	On the extended depth of focus algorithms for bright field microscopy (Valdecasas et al., 2001; Young et al., 1993)
Machine learning-based autofocus	Deep learning for single-shot autofocus microscopy (Li et al., 2021; Luo et al., 2021; Pinkard et al., 2019)

Table 2.1 Common Methods Used in Autofocus Systems

2.3.2 Contrast based autofocus

This method relies on measuring the contrast of an image to determine the optimal focus position. As the specimen moves out of focus, the contrast of the image decreases. The autofocus algorithm calculates the focus position by finding the point where the image contrast is maximized (Langehanenberg et al., 2008). There are mathematical formulas used

in contrast-based autofocus to quantify the contrast or sharpness of an image. One common approach is to calculate the gradient of the image intensity, which represents how rapidly the intensity values change across the image. The following formula is often used to calculate the gradient magnitude (G) of an image:

$$G = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{2.1}$$

I is the intensity of the image
$\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ are the partial derivatives of the image intensity with respect to the x and y coordinates, respectively

This formula measures the rate of change of intensity in both horizontal and vertical directions. High gradient magnitude values correspond to regions with rapid intensity changes, such as edges or boundaries in the image. In the realm of contrast-based autofocus, a comparable methodology can be employed to compute the sharpness or contrast parameter for individual images within the Z-axis. This entails aggregating gradient magnitudes throughout the complete image or potentially concentrating on particular areas of significance. Subsequently, the image exhibiting the most elevated contrast metric in this computation would be designated as the optimally focused image.

2.3.3 Phase-contrast autofocus

Phase-contrast microscopy is used to enhance the visibility of transparent specimens. The autofocus algorithm here adjusts the phase ring to optimize the phase difference between the direct and diffracted light, leading to the sharpest image. Phase-contrast autofocus is particularly useful for live cell imaging (Price and Gough, 1994).

Phase-contrast autofocus is a dynamic process that lacks a one-size-fits-all mathematical formula to encapsulate its entirety. Instead, it hinges upon the meticulous examination of intensity or phase data extracted from captured images across diverse focal planes. This examination serves as the basis for deriving a focus metric or score. The underlying [Tab2.2] algorithm governing this metric computation exhibits variability contingent on both the autofocus system's configuration and the unique attributes characterizing the microscope.

Optimization of the phase-contrast autofocus steps can be outlined as follows (Chan et al., 2017):

Step	Description
Image intensity calculation	Capturing images at different focal positions and measuring pixel intensity values. These values are used to compute metrics related to contrast or phase shifts.
Focus metric calculation	Deriving a mathematical focus metric from the intensity data of acquired images. The focus metric quantifies focus quality. The formula can vary; examples include gradient(2.1) or Laplacian calculations to identify intensity changes.
Focusing algorithm	Processing focus metric values from various focal positions using an algorithm. The algorithm determines the optimal focus position, often by identifying the peak of the focus metric curve, indicating the sharpest focus point.
Feedback loop	Utilizing the calculated optimal focus position to adjust the microscope's focus motor. This iterative process involves capturing images, calculating focus metrics, and refining focus until achieving the optimal position.

Table 2.2 Phase-contrast autofocus steps

2.3.4 Intensity-based autofocus

This technique evaluates the intensity of the image pixels to find the sharpest focal point. It calculates a focus metric based on the pixel intensity distribution and then moves the focal plane to the position where this metric is maximized (Baek and Park, 2021).

The approach aligns with that in [Tab2.2], differing solely by employing Laplacian variance(2.4) instead of the gradient in the focusing algorithm. The Laplacian (L) is a mathematical operator commonly used in image processing and computer vision to detect regions of rapid intensity changes, such as edges or corners, within an image. It's often used for enhancing or extracting features that have high-frequency variations (Burt and Adelson, 1987). Mathematically, the Laplacian operator ∇^2 (2.2) can be defined in two dimensions as the sum of the second derivatives of the image intensity with respect to the x and y coordinates:

$$\nabla^2(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.2}$$

$\nabla^2(f)$ Laplacian of the function f, which in this case is the image intensity
$\frac{\partial^2 f}{\partial x^2}$ and $\frac{\partial^2 f}{\partial y^2}$ the second derivative of the intensity with respect to the x and y coordinates, respectively

In discrete form, which is applicable to digital images represented as pixel arrays, the Laplacian operator can be approximated using finite differences. For example, in a 3x3 pixel neighborhood:

$$L(x,y) = I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4 \cdot I(x,y) \tag{2.3}$$

$L(x,y)$ The Laplacian value at pixel (x, y)
$I(x,y)$ The intensity value of the pixel at position (x, y)

The variance of Laplacian can be calculated as follows:

$$Variance = \frac{\sum (\nabla f(x,y))^2}{N} \tag{2.4}$$

$\nabla f(x,y)$ The Laplacian of the pixel intensity at position (x, y)
N The total number of pixels in the region.

2.3.5 Depth from focus

Depth from Focus, also known as Focus Variation, involves capturing a series of images at different focal planes. By analyzing the variation in focus across these images, the algorithm can estimate the three-dimensional structure of the specimen. This method is effective for generating depth maps in samples with complex surface structures (Valdecasas et al., 2001; Young et al., 1993). One common approach to calculating a focus measure is using the concept of image gradients, and a popular focus measure is the Variance of Gradient (VoG). Here's [Tab2.3] a simplified algorithm representation of the process:

Step	**Description**
Image gradient calculation	Compute the gradients of the grayscale image along the x and y axes using derivative filters like Sobel or Prewitt filters. Subsequently, determine the gradient magnitude like (2.1).
Focus Measure (Variance of Gradient)	Calculate the focus measure for each image by computing the variance of the gradient magnitudes within a local neighborhood around each pixel.
Depth Map Construction	Once the focus measures for each image are obtained, construct a depth map using the calculated focus measures. Higher focus measures generally correspond to points closer to the focal plane, while lower measures correspond to points farther away.
Depth Reconstruction	Based on the depth map, further processing can be performed to reconstruct the three-dimensional structure of the specimen. This could involve visualization of depth-resolved images or creating a 3D model from the depth information.

Table 2.3 Focus variation steps

2.3.6 Machine learning-based autofocus

Machine learning approaches utilize neural networks and other algorithms to learn from a dataset of in-focus and out-of-focus images. Once trained, the model can predict the optimal focus position based on the features it has learned. This approach is adaptable and can handle a variety of microscopy scenarios. This technique employs intricate mathematical operations, frequently utilizing neural networks, to ascertain the optimal focal position from input image data. Here's [Tab2.4] a typical overview of how machine learning-based autofocus functions in microscopy (Li et al., 2021; Luo et al., 2021; Pinkard et al., 2019):

Step	Description
Data Collection	A training dataset is prepared by capturing a range of images of the sample at different focus positions. These images should cover a variety of conditions, including different samples, lighting conditions, and potential artifacts.
Feature Extraction	Relevant features are extracted from the collected images. These features could include statistical measures, texture information, intensity profiles, or any other characteristics that help determine the quality of the image.
Labeling	Each image in the dataset is labeled with its corresponding focus position. This step requires manual labeling of the images, indicating whether they are in-focus or out-of-focus.
Model Training	A machine learning model is chosen, such as a convolutional neural network (CNN), support vector regression (SVR), or another suitable architecture. The extracted features and labeled data are then used to train the model to predict the optimal focus position based on the features extracted from new, unseen images.
Validation and Testing	The trained model is validated and tested using separate datasets that were not used during training. This step ensures that the model generalizes well to new, unseen images and provides accurate focus predictions.
Deployment	The trained model is integrated into the microscope's control system. When the microscope captures images, the machine learning-based autofocus system analyzes the images and predicts the optimal focus position. The microscope's motorized stage is then adjusted accordingly to achieve the best focus.

Table 2.4 Machine learning-based autofocus steps

2.3.7 Comparison of autofocus methods

Each of these autofocus methods has its strengths and limitations. The choice of algorithm depends on factors such as the type of microscopy being used, the nature of the specimens, the required speed, and the desired image quality. Advances in technology, such as machine learning and computational imaging, have led to more sophisticated autofocus methods that can adapt to different microscopy scenarios and deliver high-quality images even in challenging conditions. [Tab2.5] presents the characteristics of the aforementioned methods.

Autofocus Algorithm	Principle and Methodology	Advantages	Limitations	Accuracy	Robustness	Efficiency	Time	Depth of Field	Cite
Contrast-based Autofocus	Measure image contrast to maximize sharpness	Simple and quick to implement	Sensitive to noise; may struggle with low contrast samples	Moderate	Moderate	High	Low to Moderate	Shallow	(Langehanenberg et al., 2008)
Phase Contrast Autofocus				Moderate to High	Moderate	Moderate	Moderate	Moderate	(Price and Gough, 1994)
Intensity-based Autofocus	Analyzes pixel intensity distribution	Widely applicable; suitable for various samples	Sensitive to uneven illumination; can be slow with complex samples	Moderate	Moderate	High	Moderate	Shallow	(Baek and Park, 2021; Luo et al., 2015)
Depth from Focus	Uses focus variation to estimate 3D structure			Moderate to High	Moderate to High	Moderate to High	Moderate to High	Deep	(Valdecasas et al., 2001; Young et al., 1993)
Machine Learning-based Autofocus	Utilizing deep learning and CNNs to train on a diverse dataset of images	Adaptable; can handle various scenarios	Requires a well-curated training dataset; initial training effort	High	High	Moderate to High	Moderate to High	Moderate	(Li et al., 2021; Luo et al., 2021; Pinkard et al., 2019)

Table 2.5 Comparison of different autofocus algorithms

Chapter 3

IntraCell autofocus evaluation

3.1 Benefits and significance of autofocus in IntraCell

Intracell technology, created by Foresee Biosystems, markedly enhances the reliability and cost-effectiveness of cardiotoxicity assessments through the meticulous examination of cardiomyocyte cells. These specialized muscle cells exhibit spontaneous generation of electrical impulses and contractions, operating without external nervous system stimuli. This inherent capability underlies the heart's autonomous rhythmic beating, necessitating an intricate analysis of delicate, live cells that faithfully emulate physiological conditions within the body. Regrettably, these cells exhibit restricted viability extrinsic to their natural environment, potentially compromising their integrity.

To counter this challenge, the MEA system from Multi-Channel Systems regulates the temperature, thus strengthening the resilience and vitality of these cells during the analysis process. This strategy proves to be instrumental in supporting the fidelity of cells to in *in vivo* conditions and minimization of potential degradation when subjected to non-optimal circumstances.

Evidently, autofocus within IntraCell plays a crucial role in optimizing laser usage for optoporation. This optimization, in turn, prevents cellular damage, enabling the possibility of conducting long-term measurements. Consequently, the integration of sophisticated algorithms, including autofocus, becomes imperative. By incorporating such advanced algorithms into the analysis workflow, the speed and precision of live cell assessments witness significant enhancements.

Autofocus algorithms play a pivotal role in swiftly and accurately fine-tuning the assay apparatus. This ensures that cells remain consistently in focus, even in the face of potential fluctuations or movements. As a result, this seamless integration substantially streamlines

the analysis process, reducing the time required for data acquisition while simultaneously elevating the accuracy of the results.

The implementation of autofocus in this context offers a range of benefits and holds significant implications for the efficiency and accuracy of live cell analysis within Intracell technology. The benefits are listed as follows [Tab3.1].

Advantages	**Description**
Enhanced Precision	Autofocus algorithms maintain optimal focus on cardiomyocyte cells, ensuring precise and accurate data capture despite cell fragility and dynamics.
Time Efficiency	Autofocus mechanisms swiftly readjust focus, accelerating data acquisition. This enables analyzing more cells in a shorter time, reducing disruptions.
Minimized Variability	Autofocus reduces focus inconsistencies, capturing accurate cell features regardless of location or movement during analysis.
Improved Viability	Swift autofocus minimizes cell exposure to non-optimal conditions, preserving fragile live cell viability during analysis.
User-Independent Operation	Autofocus eliminates manual focus adjustments, promoting standardized and reproducible analyses across users and sessions.
High-Throughput Compatibility	Autofocus facilitates the efficient transition between cells or fields of view, ideal for high-throughput analysis.
Reduced Human Error	Autofocus minimizes operator-related mistakes, enhancing experimental reliability and trustworthiness.

Table 3.1 Benefits of autofocus mechanisms in IntraCell technology

3.2 Software Architecture

An effective software architecture (SA) serves as the blueprint for constructing robust and scalable applications. It defines the fundamental structure, components, interactions, and design principles that guide the development process.

The provided [Fig3.1] illustrates the functioning of the SA governing the autofocus process. Initiated by a user command through the IntraCell graphical user interface (GUI), the PC communicates with IntraCell to activate motor Z, initiating its movement. Sequential imaging occurs with each motor adjustment, and these images are subjected to Python-based algorithms for processing on the PC. The duration of the autofocus process varies between 4 to 15 seconds, contingent upon the initial distance between the motor and the focus point.

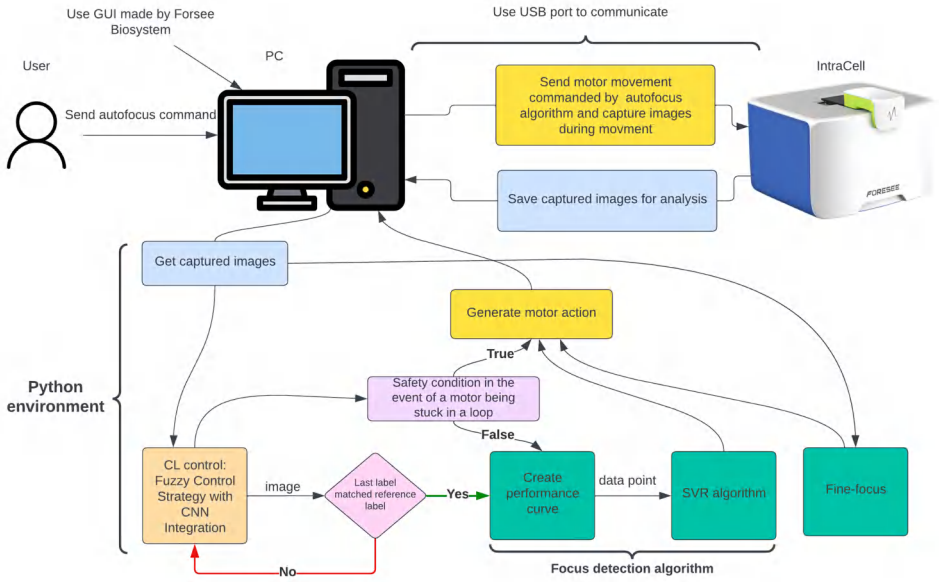


Figure 3.1 Enabling IntraCell interaction: a holistic system-level software architecture for Autofocus

Every element in [Fig3.1] is explained in the following [Sec3.2.1].

3.2.1 Advanced techniques utilized in IntraCell

During my internship at Foresee Biosystems, I contributed to the development of an advanced autofocus algorithm for IntraCell. This cutting-edge algorithm builds upon state-of-the-art techniques, integrating fuzzy control for enhanced performance.

This autofocus algorithm is structured into four distinct phases, as illustrated in [Fig3.2]:

1. Initialization Phase: The automatic autofocus process begins when the user initiates the autofocus command. During this phase, the Z-axis motor, in conjunction with optical and laser components, is shifted upwards by 100 μm and the acquisition of images. In brief, in the initial phase, the correct direction is identified, and motion is improved by implementing a constant speed for the Z-axis motor to reduce motor vibrations. This phase does not have a specific range and can function at various distances. Detailed information about this phase is provided in [Sec3.3].

2. Closed Loop Approach: The second phase employs closed loop control to progressively approach the optimal focus point. Convolutional Neural Networks (CNNs) are used to classify the images acquired during the motor's movement. The captured images are labeled, and their labels are communicated to the Fuzzy control system. The Fuzzy control strategy

reads the labels and compares them with predefined rules. If the labels satisfy any of the rules, the system proceeds to the next phase; otherwise, the initial phase procedure is repeated. This time, however, the correct direction is pursued, as it was calculated during the initial phase. This phase is repeated until the Z-axis motor is within a range of approximately ± 250 μm from the desired focus point. The activity range of this phase is compressed between ± 600 and ± 250 μm. Detailed explanations of Fuzzy control are provided in [Sec 3.4.3] and CNN in [Sec 3.4.4].

3. Precise Focusing: The third phase is dedicated to achieving precise focusing at zero μm. To achieve this result, various methods are employed to extract Relevant Values from each image. These data points are used to construct a comprehensive performance curve [Sec3.5], representing the optical behavior along the Z-axis. Subsequently, Support Vector Regression (SVR) is applied to the performance curves, as detailed in [Sec3.6.1]. The use of SVR becomes necessary when the performance curve exhibits significant oscillations due to disturbances or when the Z-axis motor has not yet reached the desired distance within the ± 250 μm range. In this case, SVR utilizes existing data points to regenerate a more accurate performance curve than the current one. This phase has an activity range between ± 250 and zero μm.

4. Fine Focusing Calibration: In the fourth and final phase, fine adjustments are made to ensure accurate focusing. This phase has an activity range between ± 30 and zero μm. [Section 3.7] provides a detailed explanation of this process.

It's important to note that each of the phases second, third, and fourth operates strictly within the specified range mentioned earlier and does not function outside of these designated ranges. Given the significant variations in image conditions that occur during motion, it's clear that each range demands a tailored approach to achieve optimal results.

The division of this autofocus algorithm into four phases is essential due to the extended length of the IntraCell Z-axis, which spans ± 3300 μm. An error could lead to incorrect motor positioning relative to the intended focus or, worse, reaching the motor's range limits and causing damage.

The automatic focus range of the algorithm is limited to ± 600 μm. It is important to note that the system never attempts to focus beyond this threshold. This limit is derived from the fact that samples placed on the IntraCell typically have minimal thickness, resulting in negligible focus changes between samples. In the rare event of a malfunction or error causing the Z-axis motor to extend beyond ± 600 μm, manual intervention is required to reposition the motor within the desired range.

These phases are sequential, with each one following the completion of the previous one. It's important to note that once the autofocus process is initiated, it cannot be interrupted, and the result is obtained at the end of the fourth phase.

This algorithm is implemented using the Python programming language, which is the language of choice for the company's development efforts.

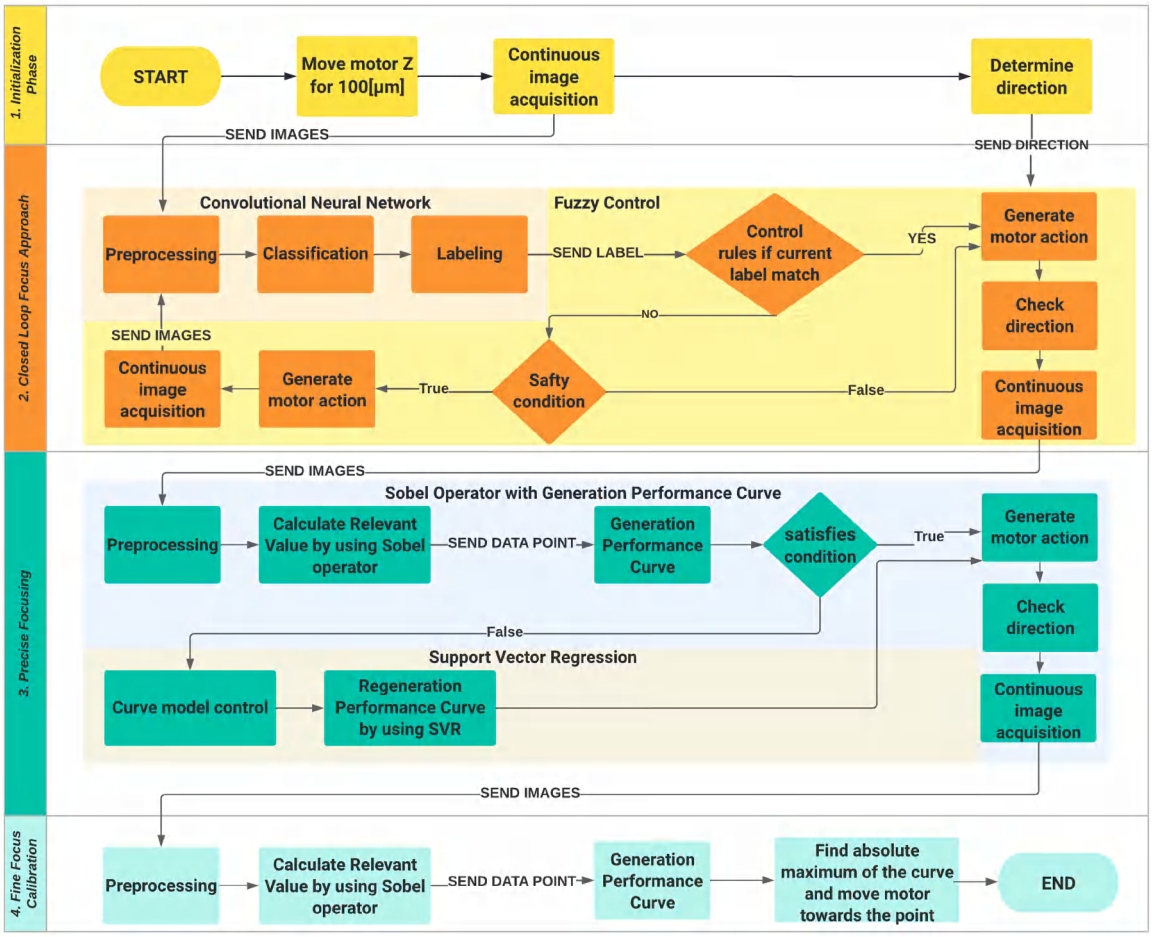


Figure 3.2 Flowchart of the entire Autofocus process

3.3 Initialization phase

In IntraCell, the Z motor is controlled by inputting the desired distance through an interface. This distance is communicated to IntraCell from a PC using a USB portal. Inside IntraCell, a dual-core microcontroller is responsible for handling received messages and sending commands to connected devices.

For the purpose of mitigating optical lens vibration induced by motor speed fluctuations, it is advisable to ensure seamless and uninterrupted movement across a substantial distance. This practice, combined with image registration, contributes to enhanced time optimization.

I've crafted a methodology to ensure a smooth and uninterrupted process, as shown in the following code snippet [Alg1] below encapsulates this process with three distinct functions. The first function focuses on Z-axis motor movement while setting a designated flag to 'False.' The second function continuously operates until the flag is 'False', capturing images at precise intervals during idle periods. The third function coordinates two threads associated with the initial functions. Once the motor reaches the desired distance, it ceases movement and image capture.

For the autofocus algorithm, the initial step involves a +100 μm motor movement, during which approximately 21 images are taken. This information aids in determining motor speed and image capture timing, enabling the calculation of the relative position from the motor's initial movement.

Algorithm 1 Running Z Motor and Image Capture

```
Require: dist: Distance, interval: Capture Interval
  function MoveZ_continuously(dist)
      MoveZ(dist)
      Set stop_flag
      textbftextcolorpurpleReturn dist
  end function

  function CaptureImages(interval)
      Initialize image_list as an empty list while not stop_flag is set do
       end
       capture_images()
      Sleep(interval)

  end function

  function run_for_duration(dist, interval)
      z_motor_thread.start()
      capture_thread.start()
      z_motor_thread.join()
      capture_thread.join()
  end function
```

3.3.1 Determine direction

To attain optical focus with efficiency, precise movement in the correct direction is vital. The algorithm below [Alg2] outlines the process for determining this direction: After image capture in the initial stage [Alg1], image variance is computed individually. Then, using the variances from the initial and final 30% of the data, the correct direction is determined.

In essence, I optimize efficiency by selectively calculating variances for only 30% of the images within a sequence—specifically, at both the beginning and end. For instance, within a sequence of 21 images, I get images 1 through 6, computing variances for each and subsequently averaging these results. I repeat this process for images 15 through 21 as well. This approach allows me to discard the remaining images, significantly boosting my processing speed. This strategic selection from the sequence's outset and conclusion serves as a diagnostic tool. If the initial images exhibit greater variance than those at the sequence's end, it indicates a need to reverse direction. Conversely, if the variance increases in the later images, it signifies that it's moving in the right direction.

This procedure is iterated for each motor movement, effectively preventing any shifts in focus when the motor gets too close to the optical focus point. Typically, images with higher variance exhibit richer frequency content. The variance within an image is intricately tied to the diversity in pixel values. When an image displays substantial variations among its pixel values, it often signifies the presence of swiftly transitioning details, edges, or focused structures. These swift alterations in intensity or color correlate with heightened frequencies in image analysis. In simple words, an image demonstrating higher variance encompasses greater frequency, resulting in more defined edges and, subsequently, a heightened sense of focus.

Algorithm 2 Dedicate Sign

```
Require: variance_list: List of variances
  function DEDICATE_SIGN(variance_list)
    # Calculate the index range for the first and last 30% of the data
    # Extract the first and last 30% of the data
    # Compare the means of the two data sets
    if direction = 'UP'
      if first_data_mean > last_data_mean
            sign → −1
      else
            sign → +1
    else
      if first_data_mean > last_data_mean
            sign → +1
      else
            sign → −1
    return sign
  end function
```

At the end of this phase, the sequence of captured images enters a closed loop state to proceed to the second phase.

3.4 Closed Loop control

Closed-loop (CL) control, often referred to as feedback control, is a method used in control systems. In this approach, the ongoing output or condition of a system is constantly observed and used as input for the system itself. Any disparity between the real output and the desired goal is utilized to create corrective measures, which modify the inputs or characteristics of the system. The primary objective is to diminish the discrepancy and uphold the system's behavior within predefined performance standards. This continuous cycle of sensing, comparing, and adjusting creates a feedback loop that empowers the system to autonomously regulate itself, attaining the intended result even when faced with disruptions or uncertainties (Ogata, 2010).

3.4.1 Key elements of CL Control

CL control components [Fig3.3] form the foundation of a dynamic system's ability to self-regulate and achieve desired outcomes. These integral elements work in concert to continuously monitor, assess, and adjust the system's behavior, ensuring it remains aligned with predefined goals. By seamlessly integrating sensors, controllers, actuators, and feedback loops, closed-loop control components enable precise management, even in the face of uncertainties and disturbances. This interplay between components facilitates the dynamic, real-time adjustments necessary for optimal performance and stability.

The principal components of CL control are (Ogata, 2010):

Controller: This is the brain of the CL system. It processes the error signal (the difference between the desired value and the actual value) and calculates the appropriate control action to be taken. The controller's design and algorithm play a crucial role in how effectively the system responds to changes.

Plant or System: "Plant" refers to the physical system or process being controlled. The plant's behavior is influenced by the control inputs and may exhibit various dynamics and responses.

Sensors: Sensors are responsible for measuring the actual state or output of the plant. These measurements are then fed back to the controller. Accurate and timely sensor data is essential for the controller to make informed decisions.

Actuators: Actuators are devices that carry out the control actions determined by the controller. They adjust the system's inputs, parameters, or outputs to bring the system closer to the desired state.

Reference or Setpoint: This is the desired value or state that the system aims to achieve and maintain. The reference is set by the operator or by system requirements. The error is calculated as the difference between the reference and the actual output.

Feedback Path: The feedback path connects the sensors to the controller, allowing the measured output to be compared with the reference value. This comparison generates the error signal that guides the control actions.

Control Algorithm: The control algorithm determines how the controller processes the error signal and computes the necessary adjustments. Different algorithms are used based on the system's dynamics, requirements, and control objectives.

Feedback Loop: The feedback loop is the central concept of CL control. It represents the continuous cycle of measuring the output, comparing it with the reference value, calculating the error, determining the control action, and applying that action to the plant. This iterative process allows the system to self-correct and maintain the desired performance.

Disturbances and Noise: CL systems often have to deal with external disturbances or noise that can affect the plant's behavior. The control system should be designed to handle such disturbances and still achieve the desired outcome.

The diagram [Fig3.3] illustrates the CL control system in action.

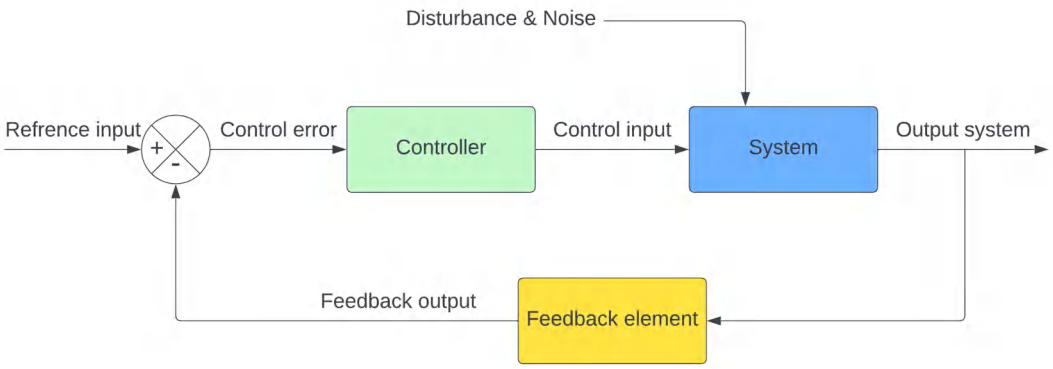


Figure 3.3 Common CL control system block diagram.

3.4.2 Closed loop control within IntraCell

In the context of Closed loop (CL) control within IntraCell, each of the elements illustrated [Fig3.4] corresponds to distinct components in the entirety of IntraCell.

The IntraCell system pertains to the machine developed by Foresee Biosystems, encompassing a B&W camera as the sensor and a stepper motor positioned along the Z-axis as the actuator [Fig2.1], aligned with the camera.

The feedback element incorporates the CNN [Sec3.4.4] implementation.

The Controller and control algorithms within IntraCell adopt Fuzzy control [Sec3.4.3]. This choice is primarily driven by the goal of implementing precise rules while minimizing mathematical computations and linear algebra. The reference or setpoint for CL comprises three types of labels denoting *"Middle"* or *"Close"* or *"Extremely Close"*, representing distances close to optimal firing positions.

Disturbances in the IntraCell context manifest as vibrations due to motor or table movement.

When the autofocus command is initiated, the Z motor ascends by 100 μm. During the movement, the camera captures successive images at a 20 *ms* interval. Typically, a series of 21 images is taken for a 100 μm motion. These images undergo labeling via CNN algorithms, focusing on distance labels. The first and last images in the series establish the start and end points, forming reference labels. The second label, associated with the last image, indicates the Z motor's current position. If this label aligns with a distance-type reference label, the CL process concludes, and autofocus transitions to the subsequent stage, involving performance curve creation [Sec3.5] and potential SVR [Sec 3.6.1] application.

Should the labels not match, the fuzzy control algorithms will engage. The labels are evaluated against predefined rules to find a matching rule. Each rule encompasses two antecedents linked to labels and a consequent directing Z motor action. The identified rule's corresponding consequent value guides the Z motor movement toward the desired point.

The required motor direction is determined by computing the variance of the image list, as elucidated in [Sec3.3.1].

Label types are defined in [Alg4] and motor actions can assume values of 100, 150, and 200 μm. The motor action value is contingent upon the labeling type; evidently, the further the motor is from the target point, the higher the motor action value.

This iterative process persists to locate the nearest focal point before advancing to the subsequent phase of curve creation. Notably, a predefined ceiling of five iterations has been set for CL operations. The primary rationale behind this limitation is to prevent overextending motor movements and encroaching upon the motor's operational limits. This safeguard is in place to counter instances when excessive noise, such as vibrations resulting from the movement of the IntraCell's hosting table, compromises image clarity.

The block diagram [Fig3.4] illustrating the CL control system within IntraCell:

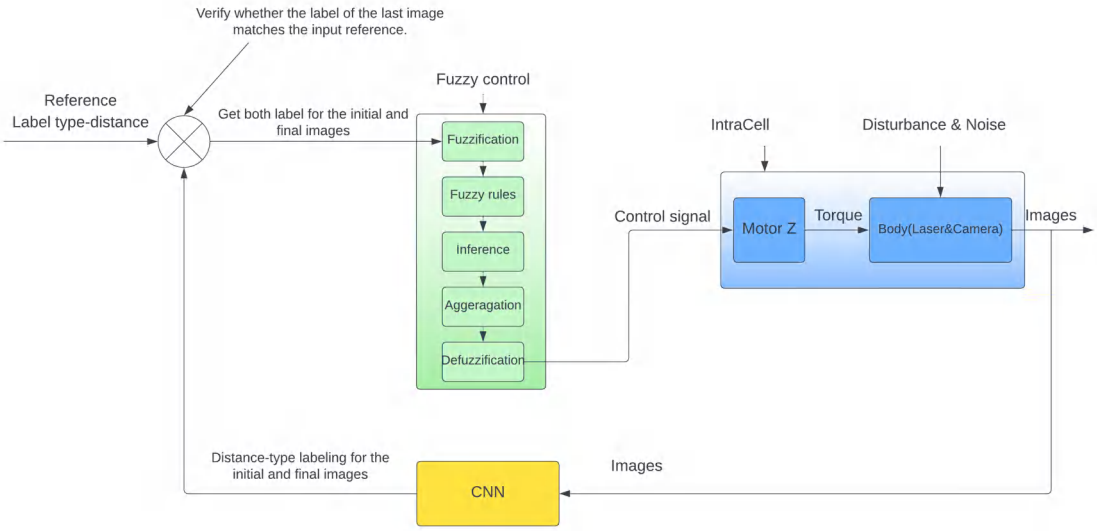


Figure 3.4 CL control system schematic within IntraCell

3.4.3 Fuzzy control

In the realm of control systems, achieving precise and effective control over complex and nonlinear processes can be quite challenging. Traditional control methods often struggle to handle systems with uncertainties, variability, and imprecise mathematical models. This is where fuzzy control emerges as a powerful and versatile approach.

Fuzzy control, often referred to as fuzzy logic control, presents a pragmatic and effective alternative for addressing a wide array of intricate control applications. It offers a practical means of constructing nonlinear controllers through the utilization of heuristic insights. These insights can be derived from various sources, such as an operator who has been directly engaged as a "human-in-the-loop" controller for a given process.

Within the framework of fuzzy control design, the approach involves soliciting the operator to articulate a set of operational rules for guiding the process. Subsequently, these rules are integrated into a fuzzy controller that mimics the human decision-making process. Alternatively, the heuristic knowledge may originate from a control engineer who has undertaken comprehensive tasks including mathematical modeling, analysis, and the development of advanced control algorithms tailored to a specific process (Passino et al., 1998).

Below [Tab3.4.3] is a broad overview of the operational framework employed by a fuzzy control system:

Step	Description
Fuzzification	Inputs from the system are converted into fuzzy values using membership functions. These functions describe how much an input belongs to each linguistic term.
Fuzzy Rules	A set of fuzzy rules is defined based on expert knowledge or learned from data. These rules consist of conditions and conclusions involving linguistic terms.
Inference	The fuzzy rules are applied to the fuzzy inputs using a mechanism known as fuzzy inference. This process determines the degree to which each rule's conclusion is true.
Aggregation	The inferred conclusions from all the rules are combined to create a composite output fuzzy set.
Defuzzification	The composite output fuzzy set is converted back into a crisp value that represents the control action to be taken.

Table 3.2 Fuzzy control steps (Passino et al., 1998)

Fuzzy control boasts numerous advantages over traditional control methods, particularly in scenarios riddled with uncertainty, complexity, and imprecision. It excels at handling uncertain and nonlinear systems, while its linguistic variables and rules offer understandable human interpretation. Fuzzy logic's adaptability, robustness, and utilization of expert knowledge enhance system performance. Its simple formulation, parallel processing potential, and applicability to complex systems make it a practical choice for real-time control. However, the suitability of fuzzy control hinges on individual system characteristics and needs (Passino et al., 1998).

3.4.4 CNN implemation in CL

I utilize CNNs for a specific classification task, wherein the classification occurs when the Z motor is significantly distant from the focal point, typically around the range of $\pm 650\ \mu m$. In such instances, the CNN generates initial labels, which are subsequently subjected to a Fuzzy control [Sec3.4.3] system. This Fuzzy control system aids in determining the precise Z-motor actions that are required to initiate adjustments, aiming to bring the system closer to optimal focus. In other words, the CNN classifies the images by labeling them and then passes these labels to the fuzzy control system to make decisions on how far is the motor Z from reaching the desired focal point. This process generates a distance in microns for adjusting the Z motor.

A CNN is a type of deep neural network designed primarily for processing grid-like data, such as images or videos. CNNs are widely used in computer vision tasks, including image recognition, object detection, and image segmentation. They are designed to automatically learn and extract hierarchical patterns and features from input data by using layers such as

convolutional, pooling, and fully connected layers. CNNs have proven to be highly effective in extracting spatial features from images.

Data structure

In the realm of computer vision tasks, images serve as the foundational input for CNNs. These images, as we commonly know, are translated into a specialized format known as tensors. Within these tensors, the intricate details of each pixel's intensity find their home.

A tensor, a dynamic and versatile multi-dimensional array, assumes the role of a data container across a spectrum of contexts. Its paramount function revolves around its capacity to encapsulate input data, intermediate computations, and eventual outcomes within the architecture of CNNs.

At the core of handling images and extracting essential patterns, tensors stand as the pivotal building blocks. Their significance lies in facilitating the interpretation of visual information and empowering the generation of accurate predictions. Thus, tensors emerge as the cornerstone in unraveling the intricacies of images, enabling us to distill meaningful insights and make informed forecasts with remarkable precision (Wu, 2017).

For grayscale images, a 2D tensor [Fig3.5-left] serves as the foundation, where each individual element signifies the intensity of a pixel. On the other hand, for color images, a more intricate 3D tensor [Fig3.5-right] is employed. The dimensions of this tensor correspond to the image's height, width, and distinct color channels such as red, green, and blue.

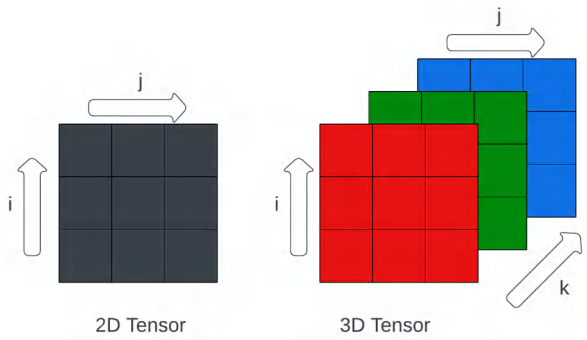


Figure 3.5 The image on the left illustrates a 2D tensor, while the one on the right depicts a 3D tensor. In my scenario, 3D tensors were employed, despite the camera capturing black and white images. This choice was made because the images have dimensions of 1280x960x3, where 1280 signifies the length, 960 signifies the width, and the value 3 corresponds to the RGB channels.

Prediction model

In my scenario, where inputs consist of images with standard resolutions and a relatively simple feature space to extract, I opted to leverage a pre-trained neural network. This choice was motivated by the goal of expediting the process of constructing and training deep learning models for focus detection. By harnessing pre-trained models like MobileNetV2, the endeavor becomes notably more efficient due to the model's prior acquisition of valuable and transferable features from extensive data exposure.

MobileNetV2 emerges as a sophisticated convolutional neural network architecture specifically crafted for streamlined and lightweight image classification and feature extraction. As a natural evolution from the original MobileNet (Howard et al., 2017), it is acclaimed for striking an optimal balance between model accuracy and computational efficiency.

The distinctive attributes and breakthroughs underpinning MobileNetV2 encompass:

1. Inverted Residual Block (specific for MobileNetV2): In MobileNetV2, an Inverted Residual Block is designed to be efficient for mobile and resource-constrained devices. It starts with an optional expansion convolution to reduce computational cost. Then, it applies depthwise separable convolution to process features efficiently. A final pointwise convolution reduces channels. If the input and output channels match, a residual connection is added to retain important information.

The "Inverted" aspect refers to reducing channels in the expansion step, making it lightweight and efficient. This block design helps MobileNetV2 achieve a good balance between accuracy and computational efficiency.

In short, residual blocks learn the difference between input and output, while inverted residual blocks in MobileNetV2 reduce computation through channel reduction and efficient convolutions, making the architecture well-suited for mobile devices.

The distinction between Residual and Inverted Residual blocks is illustrated in the [Fig3.6].

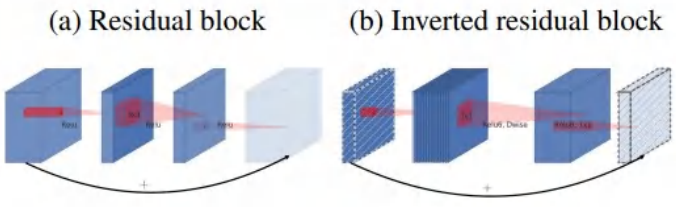


Figure 3.6 The difference between a residual block and an inverted residual. Diagonally hatched layers do not use non-linearities. We use the thickness of each block to indicate its relative number of channels. Note how classical residuals connect the layers with a high number of channels, whereas inverted residuals connect the bottlenecks. Best viewed in color (Sandler et al., 2018). The image was captured from the same article.

2. Linear Bottlenecks: In the MobileNetV2 architecture, a "Linear Bottleneck" refers to the structure of the convolutional layers within the inverted residual blocks. This concept involves keeping the bottleneck layers (typically 1x1 convolutions) linear, meaning they do not have non-linear activation functions like Rectified Linear Unit (ReLU) applied to them. This design choice is different from conventional convolutional architectures, where ReLU activation functions are commonly applied to all layers. ReLU is capable of preserving complete information about the input manifold, but only if the input manifold lies in a low-dimensional subspace of the input space.

The rationale behind Linear Bottlenecks is to allow for better information flow and gradient propagation during training. Non-linear activations like ReLU can saturate or clip gradients, which may hinder training in deep networks. By keeping the bottleneck layers linear, MobileNetV2 aims to preserve the gradient information, leading to more stable and effective training. The diagram depicted in [Figure 3.7] visually illustrates the structure of an Inverted Residual with Linear Bottleneck.

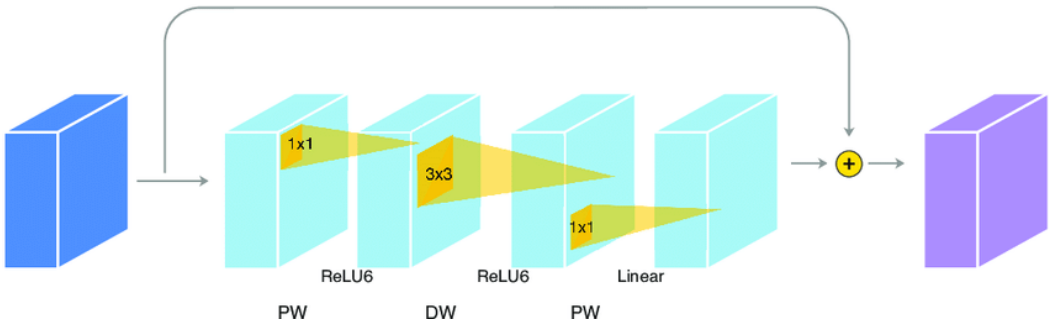


Figure 3.7 PW represents point-wise convolution, and DW represents depth-wise convolution. PW refers to a type of convolution operation where a 1x1 kernel is applied to each channel of the input tensor separately. This operation is often used for dimensionality reduction or expansion. In MobileNetV2, the PW convolution is linear, meaning it does not have a non-linear activation function applied to it. This linear nature is a characteristic of the Linear bottleneck concept. DW is a type of convolutional operation that applies a separate 3x3 kernel to each channel of the input tensor. Unlike traditional convolution, where a single kernel is applied across all channels, depth-wise convolution treats each channel independently. This approach reduces computational complexity while still capturing spatial information. Depth-wise convolution is commonly followed by point-wise convolution to combine information across channels (Wang et al., 2021). ReLU6 is a modification of the ReLU activation function that caps the output at 6, ensuring that even very large positive values do not cause the neuron to saturate. In other words, if the output of the ReLU operation is greater than 6, it's clamped down to 6. This kind of non-linearity can be particularly useful in certain contexts, such as mobile or embedded devices, where computational resources are limited. The following image is taken from https://www.researchgate.net/figure/Inverted-residual-with-linear-bottleneck-PW-represents-point-wise-convolution-DW$_fig2_351817816$

3. Width Multiplier and Resolution Multiplier: An inherent flexibility in MobileNetV2 lie in the provision of two user-adjustable hyperparameters: the width multiplier and the resolution multiplier. The width multiplier facilitates a reduction in channel numbers at each layer, while the resolution multiplier scales down the input image dimensions. These adjustments collectively lead to a substantial reduction in computational demands. I employed

the subsequent [Alg3] to enact modifications on my images prior to integrating them into the CNN framework.

Algorithm 3 Width and Resolution Multiplier

```
Require: image
  function PREPROCESS_IMAGE(image)
      # Convert image from RGB to BGR
      # Resize the image to 224x224
      # Convert image data type to float32
      # Preprocess image using MobileNetV2 preprocessing
      return image
  end function
```

In the code snippet provided above, the data type of the image is converted to float32. Neural networks often work with floating-point data to enable gradient calculations during training.

4. Global Depthwise Pooling: Is a technique used in the MobileNetV2 architecture to reduce the spatial dimensions of feature maps while preserving depthwise information. It's often used as an alternative to global average pooling in certain convolutional neural network architectures, including MobileNetV2.

Global depthwise pooling offers an advantage over traditional global average pooling by preserving channel-wise features while still reducing spatial dimensions. This can be beneficial for tasks where spatial information across channels is important. It's worth noting that global depthwise pooling is one of the components that contribute to the efficiency and effectiveness of the MobileNetV2 architecture, which is designed for mobile and resource-constrained devices like IntraCell.

Layers and Architecture

The MobileNetV2 architecture, distinguished by its input image size of (224, 224) pixels, is commonly denoted as "MobileNetV2 1.0 224." This iteration adopts an alpha value (width multiplier) of 1.0, accompanied by an input image resolution of 224x224 pixels.

Alpha refers to the width multiplier parameter that controls the number of channels (also known as the width) of the model's convolutional filters. It's used to scale the number of filters in each layer, effectively adjusting the computational complexity and model size. More specifically, the alpha parameter is typically set between 0 and 1, where (Sandler et al., 2018):

alpha = 1.0: This corresponds to the baseline architecture with the original number of filters in each layer. It represents the full-sized model and provides a balance between accuracy and computational complexity.

alpha < 1.0: A value less than 1.0 reduces the number of filters in each layer, making the model narrower. This results in reduced computational demands but might lead to a drop in accuracy.

alpha > 1.0: A value greater than 1.0 increases the number of filters in each layer, making the model wider. This can potentially enhance accuracy but comes with increased computational requirements.

Within this setup, the aggregate layer count encompasses a diverse range of elements, including convolutional, pooling, and fully connected layers, typically totaling around 155 layers. This tally comprehensively embraces the intricate interplay of depthwise and pointwise convolutions housed within the innovative inverted residual blocks, alongside both the initial and ultimate layers of the architecture. The neural network diagram below [Fig3.8] illustrates the interconnection of each layer within the architecture. The CNN design is constructed by stacking three fundamental building blocks: 1. Convolution layers, 2. Pooling layers, and 3. Fully Connected (FC) layers.

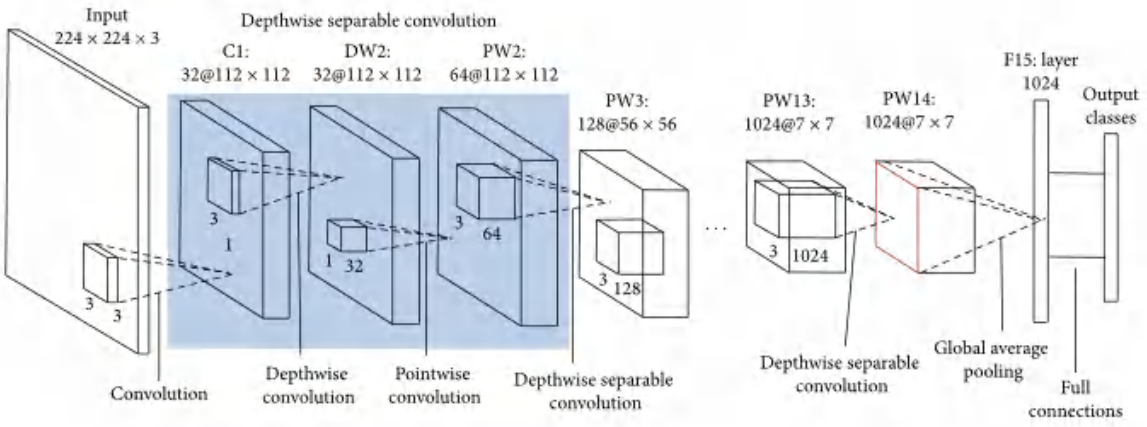


Figure 3.8 MobileNetV2 convolutional neural network architecture. The following image is taken from https://www.section.io/engineering-education/building-a-multiclass-image-classifier-using-mobilenet-v2-and-tensorflow/

Classification

In a CNN, every image is evaluated against a set of 1000 available classes in MobileNetV2. During this evaluation, a "confidence value" is assigned to each image. This value represents the predicted probability or level of certainty that the given image belongs to a specific class. It quantifies the model's conviction in its prediction.

The "confidence value" signifies how strongly the model believes that the image corresponds to a particular label among the available classes. The value falls within the range of 0

to 1, where 0 indicates low certainty (the image might not fit the class well), and 1 indicates high certainty (the image aligns well with the class).

To put it differently, this value encapsulates the likelihood of the model's prediction. The algorithm [Alg4] assigns a confidence interval to each image, offering insight into the degree of trust the model places in its classification outcome. This way, for an image that's sharply in focus, the confidence value tends to be closer to 1, and for an image that's less clear or far from the intended subject, the value may trend toward 0. This mechanism forms the basis for the algorithm's labeling process within specific confidence ranges.

Algorithm 4 Generate Confidence Phrases

```
function GET_CONFIDENCE_PHRASE(confidence_list)
    for i in range( len(confidence_list))
        if 0 ≤ confidence_list[i] < 0.1
            append("Extremely Far")
        elsif 0.1 ≤ confidence_list[i] < 0.2
            append("Far")
        elsif 0.2 ≤ confidence_list[i] < 0.3
            append("Almost Far")
        elsif 0.3 ≤ confidence_list[i] < 0.5
            append("Middle")
        elsif 0.5 ≤ confidence_list[i] < 0.8
            append("Close")
        else
            append("Extremely Close")
    return phrase_list
end function
```

After completing actions in CL, the algorithm transitions to its third phase, where it achieves precise focusing through performance curve analysis.

3.5 Precise Focusing: Approaches to constructing the performance curve

To enhance the third phase of autofocus process, I've developed a method that involves creating a performance curve. This curve serves as a visual representation of imaging conditions, with the motor's relative position on the x-axis and a corresponding value indicating image quality on the y-axis. Constructing this curve requires assigning a Relevant Value to each captured image.

Relevant Value signifies the numerical representation of a point on the performance curve's y-axis. Its name derives from its capacity to reflect a value pertinent to a precise motor position during motion. Functionally, this value denotes the level of sharpness achieved at that particular distance. Naturally, images boasting a higher Relevant Value exhibit superior

focus compared to those with lower values. The latter aspect aids in pinpointing an absolute peak within the performance curve, potentially indicating the optimal focus.

To achieve this, I've integrated a variety of techniques to compute the Relevant Value, including Sobel, Canny edge detection, Noise processing algorithms, Saturation analysis, and Fast Fourier Transform (FFT). Prior to delving into the specifics of each method employed method for constructing a performance curve, it is imperative to amass a suitable dataset and subject it to preprocessing. The subsequent [Sec3.5.1] elaborates on the approach taken to accomplish this.

3.5.1 Dataset & Preprocessing

To ensure efficient use of resources and prevent potential damage to equipment during attempts on the IntraCell, a careful a priori analysis was imperative. I accomplished this by compiling a comprehensive dataset containing over 3000 images. This dataset encompassed a range of image conditions and included distance labels for each image capture.

Before deploying any method on the IntraCell, I meticulously tested each of the aforementioned ones on the dataset. This preliminary testing phase aimed to verify if the method yielded the desired outcomes. If a method demonstrated favorable results during dataset testing, I proceeded to the subsequent stages of experimentation.

The relevance value for each image is determined within a specific area, concentrated at the center of the image, as highlighted in [Fig3.9]. This approach is driven by two primary considerations.

Firstly, this localized analysis accelerates the calculation process, streamlining the determination of the relevance value. Secondly, it aligns with the distinct characteristics of the optical lens aperture. This configuration creates an imaging scenario where the center exhibits exceptional focus while the periphery remains blurred. To circumvent potential errors that might arise from this situation during calculations, I opted to crop the image to the central region.

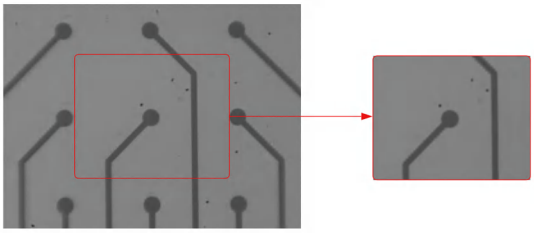


Figure 3.9 Given an image resolution of 1280x960 pixels, I've extracted a square region encompassing 1/4 of the image in each direction.

After cropping the image into smaller pieces, a filtering process becomes essential to eliminate noise and reduce the variability due to other parameters, such as heightened brightness and contrast, as documented in [Sec2.2.2].

The diagram below [Fig3.10] depicts the process: the grayscale image undergoes preprocessing, computing the Light intensity, and then inverting the image using the Bitwise_not operator.

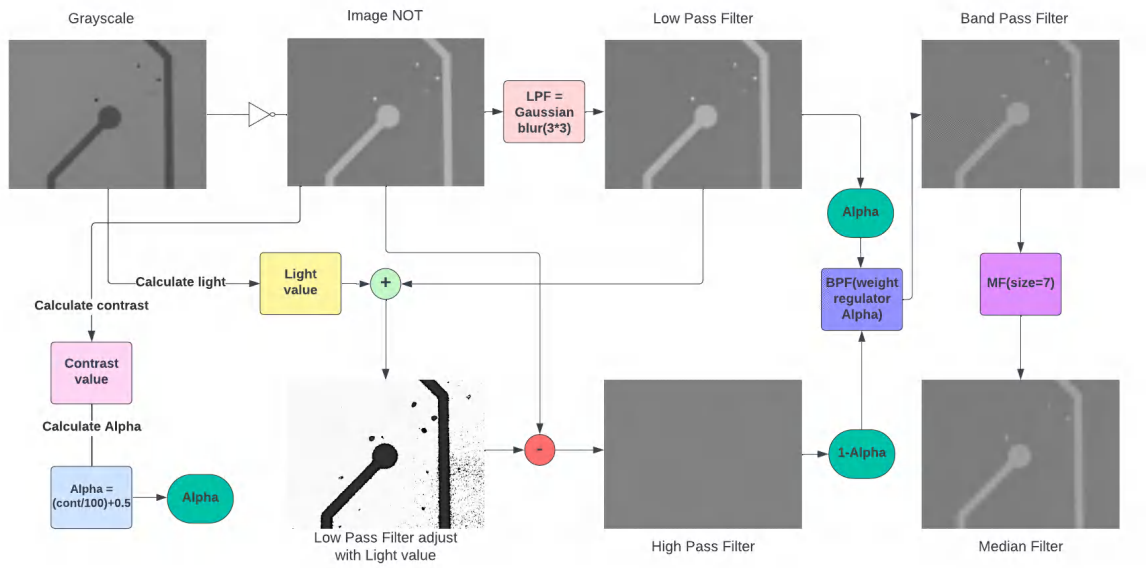


Figure 3.10 Preprocessing image before creating performance curve

Light adjusment

The calculation of light aims to adjust brightness, particularly enhancing borderline filtered pixels. Through the addition of an appropriate value, more pixels can be incorporated into the high-pass filter (HPF), resulting in a more even pixel distribution in the final image. This optimization facilitates clearer image analysis.

The rationale behind this decision lies in the numerical nature of pixel values within an image, spanning from 0, denoting black pixels, to 255, representing white pixels. The arrangement of these dark and light pixels within the image dictates its overall brightness. Should white pixels surpass black pixels in number, the image assumes a brighter aspect due to the prevalence of these higher-value pixels. Conversely, a surplus of black pixels results in a darker visual impression, accompanied by an escalated frequency of these darker pixels (E Woods and C Gonzalez, 2008).

Bitwise NOT

Bitwise NOT is a fundamental operation in computer programming that involves altering the individual bits of a binary number. When applied to a binary value, it transforms each 0 bit into 1 and each 1 bit into 0. In image filtering, applying a bitwise NOT operation to a grayscale image is valuable for specific purposes. It can enhance features and edges by flipping intensity values, making them more visible. This inversion aids in object detection, particularly when objects are darker than the background.

Contrast penalty

The Alpha coefficient regulates contrast within each image, influencing LPF and HPF utilization. Higher contrast leads to an increased Alpha value, favoring LPF. Conversely, reduced contrast stabilizes Alpha at 0.5, balancing LPF and HPF usage, although HPF is constrained.

In the absence of a Band Pass Filter(BPF) input regulator, LPF and HPF yield equivalent results, which are often unsuitable. Elevated contrast enhances detail perception, particularly in higher frequencies, while lower contrast diminishes it(E Woods and C Gonzalez, 2008).

LPF eliminates higher frequencies; HPF removes lower ones. Filter selection hinges on contrast-frequency interplay.

LPF

A Gaussian blur is a type of image-filtering technique commonly used in image processing to reduce noise and detail while preserving the overall structure of an image. It involves convolving the image with a 3x3 matrix generated from a Gaussian distribution. This filter reduces high-frequency noise and fine details in the image while preserving its overall structure. The 3x3 refers to the size of the convolution kernel, and the Gaussian nature of the filter means that pixels closer to the center of the kernel have more influence on the resulting blurred pixel value. This process helps create a smoother version of the image, effectively reducing noise and minor variations in intensity(Gedraite and Hadad, 2011).

The Gaussian blur operation involves convolving an image with a Gaussian kernel. Mathematically, the Gaussian blur can be represented as follows:

Given an input image I and a Gaussian kernel Ga with a certain standard deviation σ, the blurred output image B is obtained by convolving I with Ga:

$$B(x,y) = (I * Ga)(x,y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(x+i, y+j) \cdot Ga(i,j) \tag{3.1}$$

$B(x,y)$ The pixel value at coordinates (x,y) in the blurred output image
$I(x+i, y+j)$ The pixel value at coordinates $(x+i, y+j)$ in the input image
$Ga(i,j)$ The value of the Gaussian kernel at coordinates (i,j)

The Gaussian kernel $Ga(i,j)$ is defined as:

$$Ga(i,j) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{i^2+j^2}{2\sigma^2}} \tag{3.2}$$

σ The standard deviation of the Gaussian distribution, which controls the spread of the kernel and determines the amount of blur

It is known that when using larger kernel sizes and higher standard deviations, the resulting blur becomes more pronounced, whereas opting for smaller values leads to a milder blur effect. The Gaussian blur stands as a foundational technique within image processing, pivotal for diminishing noise and facilitating feature extraction, on the other hand, it has a negative effect on the edges so it must be correctly balanced.

HPF

Combining the output of the LPF with the computed light intensity results in a fresh image distinguished by a more luminous background and contrasting darker trails. Subsequently, performing a Bitwise NOT operation on the output image and then subtracting the resulting image from the previous one yields a distinct picture that highlights the high-frequency elements.

BPF

Two output images are generated: one associated with LPF and another with HPF light adjustments. Combining these images produces a bandpass frequency image. This amalgamation employs the *addWeighted* function from the OpenCV library, facilitating linear blending by predetermined weights. The Alpha value, previously calculated, guides this process. Alpha dictates the LPF image's influence in the blending; ranging between 0 and 1, where 0 implies no impact and 1 signifies maximum influence. LPF weight is set as Alpha,

while HPF weight is set as 1-Alpha. This choice rests on the fact that low-frequency images contain more autofocus-relevant information compared to high-frequency ones. However, utilizing both images processed by the filters is imperative; relying solely on one of the two is insufficient to achieve the desired outcome.

MF

After finalizing the image preprocessing phase, I proceeded to apply a Median Filter (MF) with a kernel 7x7. This strategic selection of the filter type and its dimensions is intended to enhance the overall image quality, simultaneously reducing unwanted noise and irregularities.

The operational principle of the MF involves substituting each pixel's value with the median value derived from the pixel values residing within a designated neighborhood, often referred to as a kernel, centered around that pixel. Opting for the median value is particularly effective due to its inherent resilience to outliers and extreme values, setting it apart from alternative statistical metrics such as the mean.

Following the preprocessing stage, the subsequent step involves executing the aforementioned methods to construct the performance curve.

In the following, all the tested methods and their implementation are described.

3.5.2 Sobel filtering and correction

Sobel is a popular edge detection algorithm used in image processing to highlight edges within an image. The algorithm focuses on identifying regions of rapid intensity changes, which often correspond to object boundaries or other significant features in an image.

The Sobel operator works by convolving the image with a small filter (also called a kernel) that calculates the gradient of the image intensity at each pixel (Castleman, 1996). This gradient information helps determine how quickly the intensity of the image changes in the horizontal and vertical directions. The Sobel operator consists of two separate filters: one for detecting horizontal edges (3.3) and the other for detecting vertical (3.4) edges.

The Sobel filters are typically defined as:

Horizontal Sobel filter:

$$G_x(i,j) = \begin{bmatrix} -1 \cdot I(i-1,j-1) & + & 0 \cdot I(i-1,j) & + & 1 \cdot I(i-1,j+1) \\ -2 \cdot I(i,j-1) & + & 0 \cdot I(i,j) & + & 2 \cdot I(i,j+1) \\ -1 \cdot I(i+1,j-1) & + & 0 \cdot I(i+1,j) & + & 1 \cdot I(i+1,j+1) \end{bmatrix} \tag{3.3}$$

Vertical Sobel filter:

$$G_y(i,j) = \begin{bmatrix} -1\cdot I(i-1,j-1) & - & 2\cdot I(i-1,j) & - & 1\cdot I(i-1,j+1) \\ 0\cdot I(i,j-1) & + & 0\cdot I(i,j) & + & 0\cdot I(i,j+1) \\ 1\cdot I(i+1,j-1) & + & 2\cdot I(i+1,j) & + & 1\cdot I(i+1,j+1) \end{bmatrix} \tag{3.4}$$

G_x and G_y The gradient value computed using the Sobel kernel with respect to x and y direction with dimension 3x3
i and j Represent the row and column indices of the convolution kernel (G_x or G_y)

After the application of Sobel filters, the gradient magnitude (2.1) is computed, representing the rate of change of the function at a specific spatial point. The resulting gradient magnitude is visually depicted in the accompanying [Fig3.11]. However, when aiming to derive a performance curve from a sequence of continuously captured images, the assignment of a Relevant Value to each image becomes imperative.

To address this, I employ a focus measurement strategy by calculating the image average and subsequently refining it using the Light percentage as a dynamic exponent. This correction process for the mean introduces significant modifications to the curve's shape. This approach is vital, as a mere reliance on the mean could result in a linear representation rather than the desired curve.

The choice of using the Light percentage as the exponent stems from its ability to swiftly capture fluctuations in relevance as an image approaches focus. Images captured in proximity to the focus tend to exhibit a notable increase in light intensity.

Upon scrutinizing the resulting performance curve, its peak signifies an image characterized by optimal focus.

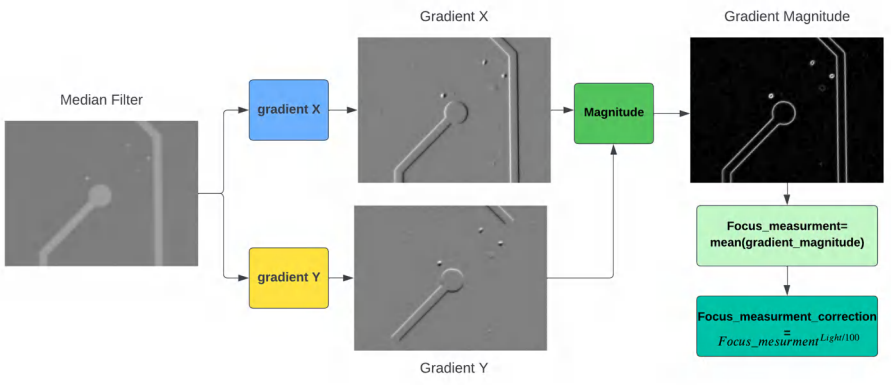


Figure 3.11 Relevant Value calculation by using Sobel

3.5.3 Canny edges detector filtering

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. The Canny edge detector aims to identify the boundaries of objects within an image by highlighting areas of rapid intensity change, which are likely to correspond to object edges (Castleman, 1996).

At the end of the preprocessing phase [Fig3.12], the Canny edge detection technique was employed to compute the significance score. This approach shares similarities with the method described in the preceding section, with the key distinction being the utilization of the Canny edge detection operator in lieu of the Sobel operator.

Similarly, in this context, the determination of the significance score involves evaluating the variance extracted from the processed image. This process is consistently applied across a sequence of images, culminating in the creation of a performance curve. Notably, within this curve, the pinnacle corresponds to the image exhibiting exceptional focus.

A limitation I've encountered with the Canny edge detection algorithm lies in its sensitivity to the selection of thresholds, particularly the initial parameter which dictates the lower threshold value. This lower threshold value plays a crucial role in determining which pixel gradients are suppressed and categorized as non-edges.

The algorithm further employs a second parameter known as the upper threshold value, which serves to identify pixels with gradient magnitudes surpassing this threshold, designating them as strong edges. Additionally, pixels exhibiting gradient intensities between the lower and upper thresholds are marked as weak edges, contingent upon their connection to strong edge pixels.

In my specific scenario, I found that setting the lower threshold to 10 was effective, but the upper threshold displayed variability between 30 and 90. This considerable range for the upper threshold value introduced a degree of instability to the edge detection process, potentially compromising its robustness and overall reliability.

Mathematically, the hysteresis thresholding involves using two threshold values, T_{low} and T_{high}, and categorizing pixels as follows:

If $|G_{\text{mag}}(x,y)| \geq T_{\text{high}}$, pixel is a strong edge.

If $T_{\text{low}} \leq |G_{\text{mag}}(x,y)| < T_{\text{high}}$, pixel is a weak edge.

If $|G_{\text{mag}}(x,y)| < T_{\text{low}}$, pixel is not an edge.

$G_{\text{mag}}(x,y)$ represents the magnitude of a gradient at a specific pixel location (x,y) in an image. To calculate $G_{\text{mag}}(x,y)$ you can use (2.2).

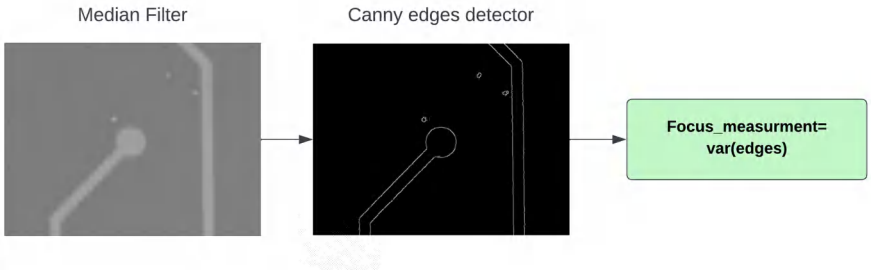


Figure 3.12 Relevant Value calculation by using Canny

3.5.4 Fast Fourier Transform

FFT is a mathematical algorithm used to transform a signal from its original domain (typically the spatial domain) to a frequency domain representation. In the context of image processing, the FFT is particularly useful for analyzing the frequency components present in an image (Castleman, 1996).

In my particular scenario, the FFT technique was employed to derive the essential values necessary for constructing the performance curve. The process unfolds as follows [Fig3.13]:

Following the initial preprocessing of the image output from the MF, the discrete Fourier transform (DFT) of the image is computed. The DFT operation takes the grayscale image as input, converting it into a float32 data type for the ensuing calculation. The output of the DFT operation comprises a complex array characterized by both real and imaginary components. These outcomes are stored within the Fourier variable.

Calculation of DFT for a pixel (u,v) in the frequency domain (3.5):

$$F(u,v) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} G(x,y) \cdot e^{-j2\pi\left(\frac{ux}{w}+\frac{vy}{h}\right)} \tag{3.5}$$

(w,h) The width and height of the image
(u, v) Represent the horizontal and vertical frequency index, respectively
$G(x,y)$ The value of the image pixel at coordinates (x,y)
j The imaginary unit

Subsequently, the DFT's zero frequency component is repositioned to the center of the frequency spectrum. This is achieved using the Shift-FFT function, which adeptly rearranges the quadrants of the complex DFT array. This realignment places the low-frequency components at the core of the representation.

Following this repositioning, the inverse operation (3.7) is performed, returning the zero-frequency component to the corners of the array. This process effectively restores the

DFT array to its original orientation. The next step involves executing a 2D inverse Fourier transform (3.8) on the modified array. This action serves to reconstruct the altered image within the spatial domain.

At this juncture, the dimensions of the reconstructed image are evaluated, and a logarithmic transformation (3.9) is employed to rescale it. This adjustment is commonly undertaken to enhance the visualization of the frequency components. By applying the absolute value function, the magnitude of the complex values is ascertained. A logarithmic function and scaling are then employed.

Finally, the mean value (3.10) of the logarithmic magnitude of the reconstructed image is computed. This resultant value forms the foundation of the pertinent parameter, pivotal in constructing the curve.

FFT shift operation (3.6) rearranges frequency components as follows:

$$F_{\text{shifted}}(u,v) = F(u - \frac{w}{2}, v - \frac{h}{2}) \tag{3.6}$$

Inverse FFT shift operation:

$$F_{\text{ishifted}}(u,v) = F_{\text{shifted}}(u + \frac{w}{2}, v + \frac{h}{2}) \tag{3.7}$$

Inverse Discrete Fourier Transform (IDFT) to reconstruct the image:

$$R(x,y) = \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} F_{\text{ishifted}}(u,v) \cdot e^{j \cdot 2\pi \left(\frac{ux}{w} + \frac{vy}{h}\right)} \tag{3.8}$$

$R(x,y)$ The output of the DFT operation at the coordinates (x, y) in the resulting frequency domain.

The magnitude of the reconstructed image and application logarithmic scaling:

$$M(x,y) = 20 \cdot \log(|R(x,y)|) \tag{3.9}$$

The mean value of the magnitude image:

$$\text{Mean} = \frac{\sum_{x=1}^{w} \sum_{y=1}^{h} M(x,y)}{w \cdot h} \tag{3.10}$$

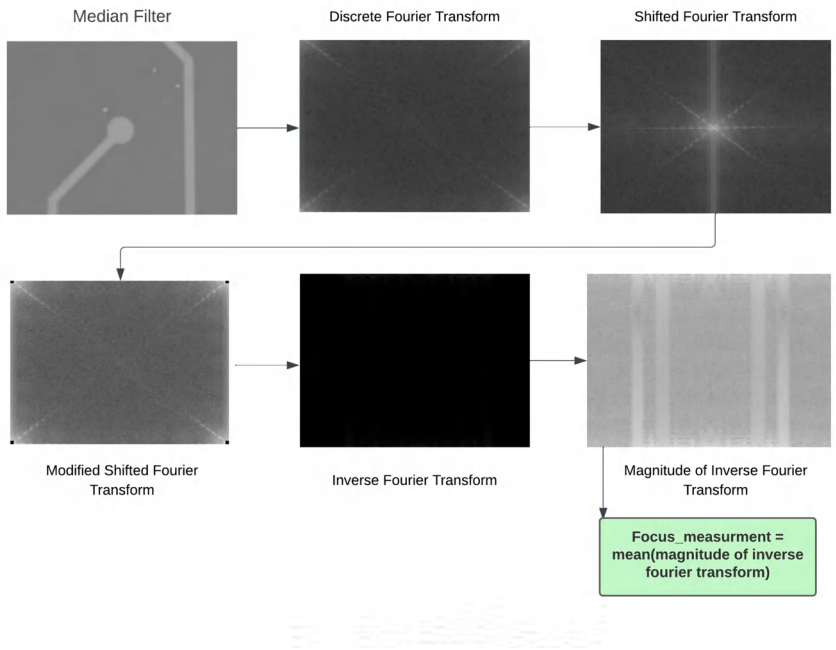


Figure 3.13 Relevant Value calculation by using FFT

3.5.5 Noise processing

The fundamental concept behind this method for extracting relevant information from an image involves introducing controlled noise to the image and then evaluating its Signal Noise Ratio (SNR). The initial inspiration for this approach came from my supervisor, Carolina Scandellari. According to her perspective, an image captured with exceptional focus tends to be more adversely impacted by noise compared to an image that was initially out of focus.

To illustrate this, consider the analogy of two pieces of clothing: a white garment symbolizing an image with precise focus, and a black one representing an image that lacks focus. When these garments accumulate dirt (analogous to noise), both are affected equally. However, the white garment, like the sharply focused image, displays the dirt more prominently.

Extensive preprocessing steps are not undertaken because the focus here is on noise addition. The process depicted in the [Fig3.14] involves applying a Bitwise Not operation to the grayscale image, followed by a Gaussian Blur utilizing a 3x3 kernel. Subsequently, flicker noise with a frequency of 200 Hz and an amplitude of 2 decibels (dB) is introduced.

Flicker noise, or intermittent noise, also referred to as 1/f noise or pink noise, is a distinctive type of noise characterized by its power spectral density, which is inversely correlated with frequency. In the context of a grayscale image, flicker noise manifests

as variations in pixel intensity that are more pronounced at lower frequencies and more consistent at higher frequencies.

Flicker noise (3.11) can be mathematically represented as follows:

$$flicker\_noise[i,j] = A \cdot \sin\left(\frac{2\pi f_i}{M}\right) \tag{3.11}$$

A The amplitude
f The frequency
Image by a 2D array with dimensions (M×N)
$flicker\_noise[i,j]$ The resulting flicker noise 2D array has the same dimensions as image
M, N The number of rows and columns in the image array, respectively
i Ranges from 0 to M-1
j Ranges from 0 to N-1

Let's consider having a grayscale image with varying pixel intensities. The inclusion of flicker noise in this image leads to regions exhibiting gradual, low-frequency alterations in intensity that stand out more prominently than higher-frequency changes. This gives the concept a subtly textured look, with changes in intensity exhibiting a characteristic smoothness (Boyat and Joshi, 2015).

Next, I applied uniform noise (3.12) to the images. This choice was deliberate, as the consistency of this noise type across all images is crucial. Inconsistent noise types can introduce variations that disrupt the autofocus algorithm. For instance, utilizing Salt & Pepper noise with a random distribution creates divergent conditions among sequentially captured images.

The uniform noise value needs to be constrained within the range of 0 to 255, given that it's being applied to a grayscale image. This compression ensures that the subsequent analyses remain manageable. This precaution is particularly important as images are subject to inherent noise during the recording phase, compounded by flicker noise.

To prevent undue darkening or lightening of the images, it's advisable to avoid positioning the uniform noise values at the extreme limits of the grayscale spectrum. Therefore, a choice was made to set the uniform noise amplitude at a quarter of the grayscale range, specifically 1/4 of 255, which amounts to 60. This judicious selection strikes a balance between enhancing the image and keeping the overall brightness within acceptable bounds.

$$\text{Uniform noise}_{M\times N} = \begin{bmatrix} 60 & 60 & \dots & 60 \\ 60 & 60 & \dots & 60 \\ \vdots & \vdots & \ddots & \vdots \\ 60 & 60 & \dots & 60 \end{bmatrix} \tag{3.12}$$

M, N The number of rows and columns in the image, respectively

Moving forward, to calculate the SNR, I isolated the maximum signal from the last processed image. Subsequently, I computed the Root Mean Square (RMS) noise of the image. The SNR (3.15) was then determined using the following formula:

$$\text{max\_signal} = \max(\text{noisy\_image}) \tag{3.13}$$

$$\text{rms\_noise} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{noisy\_image}[i] - \text{mean}(\text{noisy\_image}))^2} \tag{3.14}$$

$$\text{SNR} = \log_{10}\left(\frac{\text{max\_signal}}{\text{rms\_noise}}\right) \tag{3.15}$$

N The total number of pixels in noisy_image $mean(noisy\_image)$ The mean value of the pixel intensities in noisy_image.

Having acquired a relative value for an image, the subsequent step involves analyzing a series of continuously captured images to construct a performance curve. Unlike prior scenarios, in this context, the curve's minimum point signifies excellent focus.

Notably, the addition of noise affects images with an excellent focus more profoundly than those already out of focus. The performance curve corroborates this observation, demonstrating that images with lower SNR values generally correspond to diminished image quality due to a comparatively weaker signal when contrasted with the level of noise.

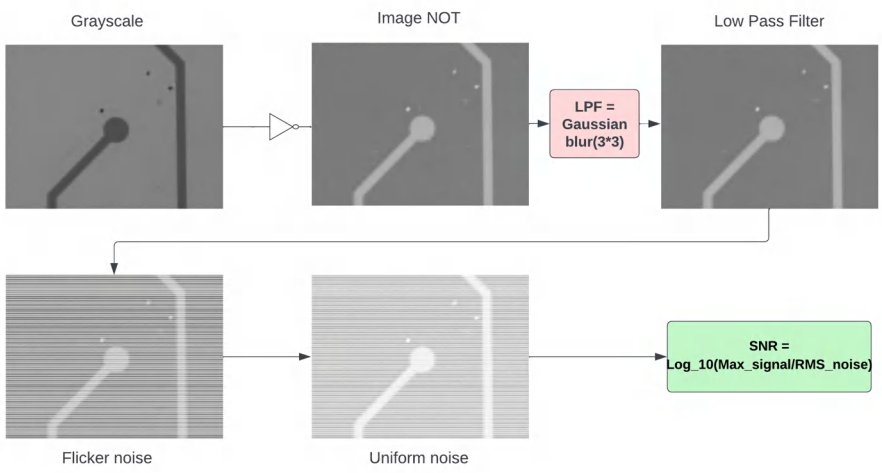


Figure 3.14 Relevan Value calculation by using SNR

3.5.6 Saturation analysis

This approach eliminates the need for extensive preprocessing steps. Instead, it begins by transforming the grayscale image into the HSL (Hue, Saturation, Lightness) color space. HSL, which encapsulates Hue, Saturation, and Lightness components, serves as a comprehensive color model used to define and characterize colors (Garg et al., 2022).

Hue: The hue component signifies the inherent color of a pixel. It's typically denoted as an angle on a color wheel, where different angles correspond to distinct colors. For instance, red might be denoted as 0 degrees, green as 120 degrees, and blue as 240 degrees.

Saturation: Saturation governs a color's intensity and purity. Higher saturation values yield vibrant and vivid colors, whereas lower values lead to subtler or grayscale-like tones.

Lightness: Also referred to as luminance or brightness, lightness controls the overall brightness or darkness of a color. A lightness of 0 corresponds to black, while 100 indicates white. Intermediate values encompass various shades of color.

Given my specific use case, the method requires extracting a Relevant Value. The Saturation component emerges as the most efficacious option due to its reduced reliance on image conditions. Notably, changes in focus significantly influence color intensity, an aspect masterfully controlled by the Saturation channel. Consequently, the subsequent step solely involves the extraction of the Saturation channel, which sets the stage for further analysis.

Following this [Fig3.15], a Gaussian blur is employed to diminish noise and achieve image smoothing. Ultimately, an Equalize Histogram process is implemented. This technique, harnessed within image processing, heightens image contrast by redistributing pixel values.

It proves especially valuable when pixel values are confined to a narrow range, causing the image to appear flat or devoid of intricate details (Garg et al., 2022).

Naturally, the mean value is instrumental in calculating the pertinent figure. Remarkably, the peak of the performance curve epitomizes an image that boasts exceptional focus and clarity.

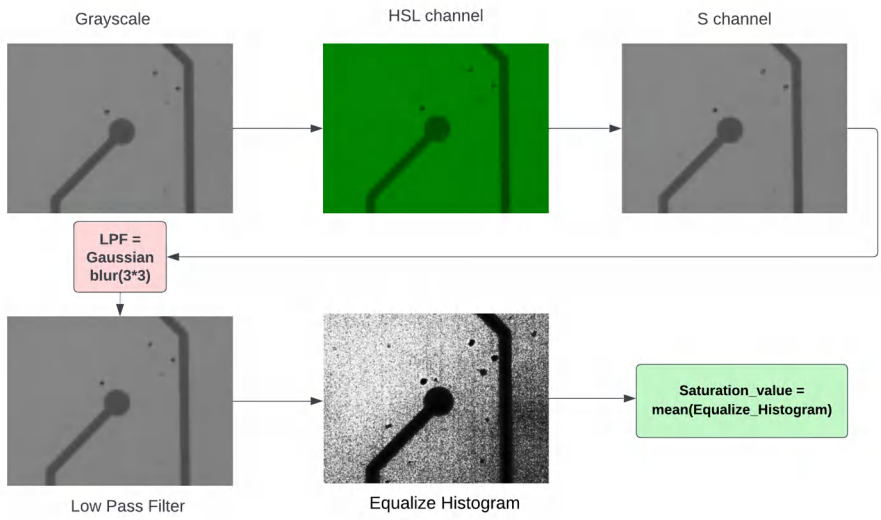


Figure 3.15 Relevant Value calculation by using Saturation analysis

3.6 Optimizing Performance Curve using Support Vector Regression

The techniques outlined in the preceding section provide ample means for constructing the performance curve and identifying a pivotal value, be it a maximum or minimum, as applicable in Noise processing scenarios. These approaches serve the purpose of uncovering an image associated with impeccable focus. Nonetheless, it's important to note that this methodology does possess a constraint inherent to its operation within a particular range. Specifically, this limitation becomes evident when the motor operates within a span of approximately $\pm 250\ \mu m$ around the optimal focus point, or essentially, when it hovers around the pinnacle of focus.

While the methods outlined exhibit commendable proficiency within shallow ranges, the incorporation of Machine Learning (ML) becomes pivotal when dealing with deeper ranges. It's worth noting that the realm of research has previously acknowledged (Li et al., 2021;

Luo et al., 2021; Pinkard et al., 2019) the significance of support vector regression (SVR) and previously mentioned Convolutional Neural Network (CNN) algorithms as notable components of the ML landscape. In my research, I've harnessed the power of these two techniques to forge an algorithm of heightened resilience and precision.

In the field of predictive analysis, SVRs are effectively coupled with performance curves to anticipate trends and forecast the divergence of motor Z from its focal point. SVR, renowned for its remarkable robustness and precision within the midrange of approximately $\pm 250\ \mu m$, consistently delivers reliable predictions.

3.6.1 SVR

SVR is a supervised machine learning technique to handle regression problems. Regression analysis is useful for analyzing the relationship between a dependent variable and one or more predictor variables. SVR formulates an optimization problem to learn a regression function that maps input predictor variables to output observed response values. SVR is useful because it balances model complexity and prediction error, and it has good performance for handling high-dimensional data. SVR is an extension of the Support Vector Machine (SVM) classification algorithm. However, unlike SVM classification, which produces binary output (i.e., a class label), SVR handles a regression problem that allows for real-valued function estimation. SVR applies the basic idea of SVM, i.e., a sparse kernel machine that performs classification using a hyperplane defined by a few support vectors (Mechelli and Vieira, 2019).

In my case, SVR uses a performance curve that's created from the Relevant Values of each image to make predictions. In other words, it examines how the curve behaves, like its slope, at various distances, to make accurate predictions. However, before making predictions, a classification step is performed to accelerate the process.

Classification

Usually, the classification process revolves around the performance curve, which is split into three categories based on the location of its peak: Mid case, Head case, and Tail case.

1. Mid case

A "Mid case" arises when the highest point on the performance curve is identified along the Y-axis, as illustrated in [Fig3.16]. This peak must be centered within the curve's length, achieved by disregarding the initial and final 10% through a specific formula (3.16).

$$l < max_y < L - l \tag{3.16}$$

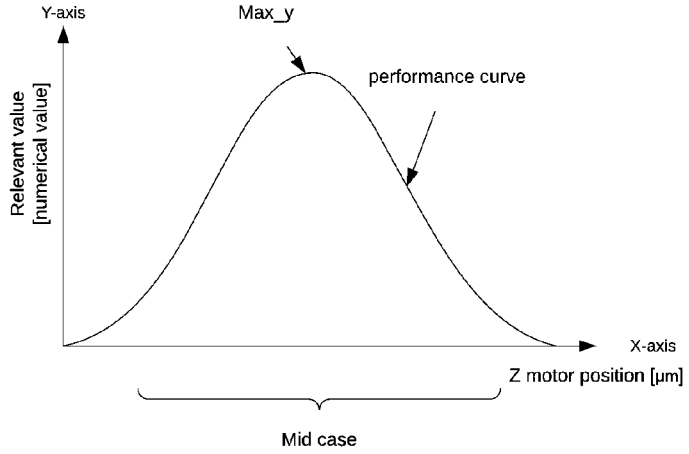


Figure 3.16 Mid case in performance curve

L: The length of the performance curve
max_y: The maximum value in performance curve w.r.t Y-axis
l: 10% of the length of the performance curve

This method checks if a peak exists within this range. If a peak is found, SVR predictions aren't required. Only the X-axis position of the peak is located, and the motor is adjusted accordingly for more analysis.

If no peak is present in this range, the algorithm categorizes the curve into one of the other two classes. This type of analysis is carried out to accelerate the autofocus algorithm and prevent unnecessary predictions.

2. Head case

The "Head case" arises when the highest point is detected in the ***l*** region, which constitutes the initial 10% of the curve's length, as illustrated in [Fig3.17]. At this juncture, SVR employs 20 data points to make predictions that lean toward the "Head" direction.

3. Tail case

Instead, the "Tail case" arises when the highest point is detected in the final 10% of the curve's length, as illustrated in [Fig3.17]. At this juncture, SVR employs 20 data points to make predictions that lean toward the "Tail" direction.

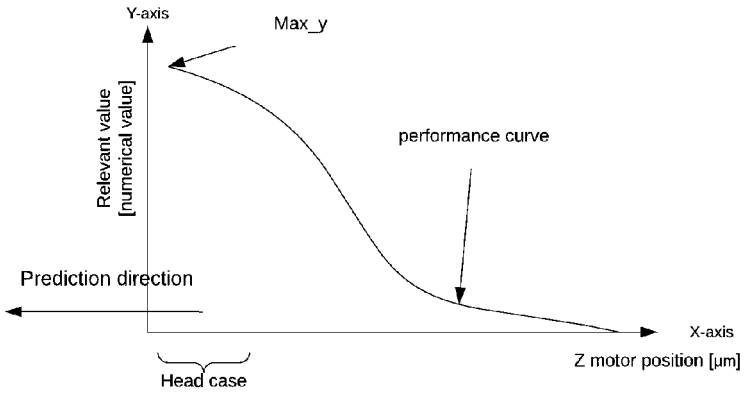


Figure 3.17 Head case in performance curve

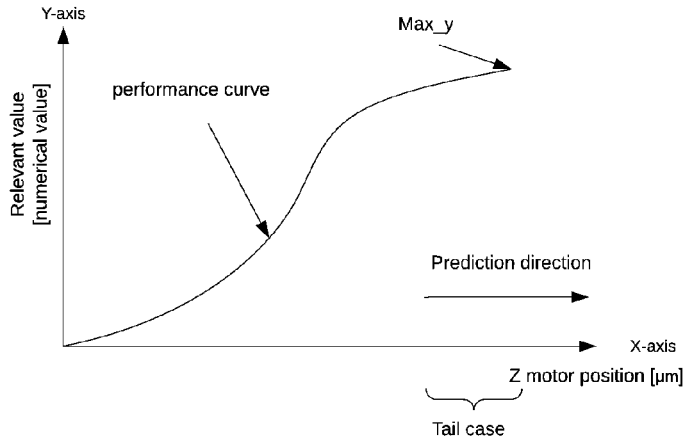


Figure 3.18 Tail case in performance curve

Clearly, for both the "Head case" and "Tail case," it's essential to employ an SVR with a suitable kernel and well-chosen parameters to achieve a satisfactory outcome. Selecting an appropriate kernel and parameters for the SVR model hinges upon the inherent characteristics of the data and the underlying relationship that demands representation.

Kernel

The choice of the kernel in SVR affects how the algorithm fits a hyperplane to the data points and ultimately determines the model's ability to capture complex relationships within the data. The kernel allows SVR to handle non-linear relationships between the input features and the target variable by transforming the data into a space where linear relationships might exist.

The kernel functions in SVR work by implicitly computing the dot product between the transformed data points in the higher-dimensional space, without explicitly performing the transformation. This dot product is used to define the similarities or distances between data points, which are crucial for the SVR algorithm's optimization process.

Commonly used kernel functions in SVR include [Tab3.3].

Kernel	**Description**
Linear Kernel	This kernel performs a linear transformation, meaning it doesn't change the original feature space. It's useful when the data seems to exhibit a linear relationship.
Polynomial Kernel	This kernel applies a polynomial transformation to the data points, allowing SVR to capture polynomial relationships between the features and the target variable.
Radial Basis Function (RBF) Kernel	The RBF kernel is also known as the Gaussian kernel. It maps the data into an infinite-dimensional space and is useful for capturing complex non-linear relationships. It's often a good choice when the underlying relationships in the data are not easily discernible.
Sigmoid Kernel	The sigmoid kernel applies a hyperbolic tangent transformation to the data. It's typically used when the relationship between the features and the target is logistic in nature.

Table 3.3 kernel types (sci, 2007)

I chose the RBF kernel for SVR due to the non-linear nature of the obtained data. The linear kernel is not suitable in this context as the data does not exhibit linear behavior. When considering a sigmoid kernel, demonstrates logarithmic behavior that, at a specific threshold, attains a saturation level. This leads to the predictive curve stabilizing at a particular point.

In deciding between the polynomial and RBF kernels, I made a selection based on a thorough analysis of their respective advantages and disadvantages. This decision-making process, along with its outcomes, has been documented in a [Tab3.4].

Selecting an appropriate degree for the performance curve can be a complex task, especially when dealing with cases where the output does not conform neatly to a well-defined curve. To mitigate the risk of making an incorrect choice in determining the optimal degree for a Polynomial kernel, I opted instead for the RBF kernel. This decision was motivated by the desire to prevent potential shortcomings associated with an ill-fitting polynomial model.

Kernel	**Advantages**	**Disadvantages**
RBF Kernel	Flexibility: The RBF kernel can model complex and non-linear relationships effectively, suitable for various regression problems.	Hyperparameter Tuning: RBF kernel has a hyperparameter γ controlling Gaussian width. Finding optimal γ is challenging, requiring careful tuning as it strongly shapes decision boundaries.
	Universal Approximation: RBF kernels are universal approximators, approximating any function with enough data.	Overfitting: High flexibility can lead to overfitting if not properly regularized. Use cross-validation and regularization.
	Implicit Feature Mapping: RBF kernel implicitly maps data into high-dimensional space, capturing hidden patterns.	
	Adaptive Influence: RBF kernel adjusts support vector influence based on distance from decision boundary, giving more weight to nearby points.	
Polynomial Kernel	Non-Linearity: Polynomial kernel captures polynomial relationships, modeling curved decision boundaries.	Sensitivity to Degree: Polynomial kernel has degree hyperparameter, determining polynomial mapping. Appropriate degree choice is crucial to avoid overfitting/underfitting.
	Lower Dimensionality: Polynomial kernel can map data into lower-dimensional spaces, reducing complexity/memory usage.	Limited Complexity: Polynomial kernels struggle with intricate patterns better captured by RBF kernel.

Table 3.4 Advantages and disadvantages of different kernel functions (sci, 2007)

Hyper-parameters

In SVR, hyperparameters play a crucial role in determining the performance and behavior of the model. Given that selecting incorrect parameters can lead to issues with fitting, as illustrated in the [Fig3.19].

Overfitting [Fig3.19-a] occurs when a model learns the training data too well, capturing not only the underlying patterns but also the noise in the data. This results in a model that performs exceptionally well on the training data but performs poorly on new, unseen data. In an SVR context, an overfitting model will have a complex decision boundary that closely follows every training data point, including outliers and noise.

Adequate fitting [Fig3.19-b], also known as the "right" fitting, occurs when the model generalizes well to new data. It captures the underlying patterns in the data without overemphasizing noise or over-simplifying the relationships. In SVR, a properly fitted model will have a decision boundary that captures the central trend of the data without overreacting to individual data points.

Underfitting [Fig3.19-c] occurs when a model is too simple to capture the underlying patterns in the data. As a result, it performs poorly not only on the training data but also on new data. In SVR, an underfitting model might have a linear decision boundary for data that requires a more complex representation.

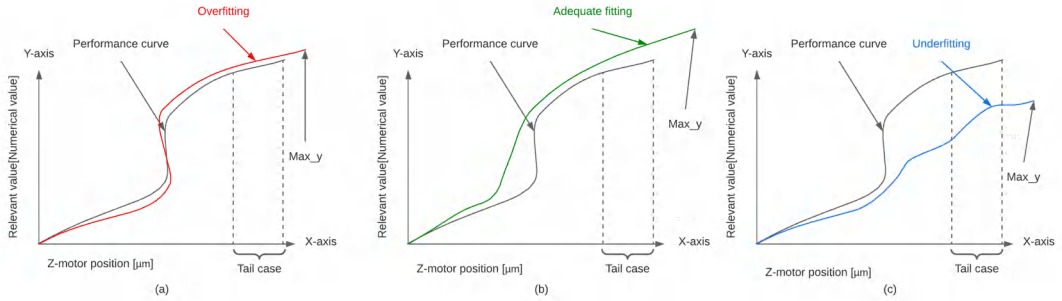


Figure 3.19 Different types of Fitting in the SVR "Tail case"

The trio of hyperparameters associated with SVR employing RBF kernels - encompassing the cost parameter, epsilon, and gamma - engage in tailored interactions that exert discernible impacts on model performance. Ensuring a harmonious equilibrium among these hyperparameters is pivotal in achieving an SVR model that is not only well-generalized but also capable of yielding optimized outcomes.

1. Cost parameter

The regularization parameter balances the trade-off between achieving a low training error and a low testing error. It determines the width of the margin and the tolerance for violations of the margin by the training data points (sci, 2007).

Increase in C: When increasing the value of C, allows the SVR model to have a smaller margin, which means it can closely fit the training data points. This might lead to better training accuracy but could result in overfitting.

Decrease in C: Conversely, when decreasing the value of C, the SVR model aims to achieve a larger margin, even if it means some training data points are not exactly fitted. This can help prevent overfitting and improve the model's ability to generalize to new data. However, too much emphasis on a large margin might lead to underfitting.

I specifically chose to set the parameter C to 20 after conducting cross-validation. Through this process, I systematically evaluated various values, and among these options, 20 consistently emerged as the optimal choice. Opting for a fixed value not only streamlined the cross-validation process but also demonstrated its effectiveness across multiple trials.

2. Epsilon

The parameter epsilon (ε) is the tube width around the regression line within which no penalty is incurred for errors. It defines the margin of tolerance for errors in the training data. The relationship between ε and the regularization parameter C is as follows (sci, 2007):

Large Epsilon (Wide Tube): When ε is set to a larger value, it allows for a wider tube around the regression line. This means that data points can fall within this wider tube without incurring a penalty, even if they are slightly outside the margin. A wider tube implies that the SVR model is more tolerant of errors in the training data. In this case, the influence of C on the model's performance is less pronounced.

Small Epsilon (Narrow Tube): On the other hand, when ε is set to a smaller value, the tube around the regression line narrows. This means that data points need to be closer to the regression line to be within the margin of tolerance. Smaller ε values make the model less tolerant of errors in the training data, which can lead to a stricter fit to the training data.

Here, I employed the same approach to determine the ε value, ultimately selecting 0.5 as the fixed and optimal choice.

3. Gamma

Gamma (γ) controls the "spread" of the RBF kernel. It defines how far the influence of a single training example reaches.

A **small** γ value will result in a wide influence, meaning that the SVR model will consider a broader range of data points for the decision boundary, potentially leading to a smoother and simpler model. On the other hand, a **large** γ value will result in a narrow influence, causing the SVR model to focus more on the data points that are closer to the support vectors, potentially resulting in a more complex and flexible model that captures intricate patterns in the data.

In my implementations, the γ parameter is configured as "scale". When γ is set to "scale", the algorithm dynamically tunes the γ value by considering the inverse of the standard deviation associated with the input features. In essence, this implies that the algorithm calculates γ using the following formula (3.17):

$$\gamma = \frac{1}{n_{\text{features}} \cdot \text{X.var()}} \tag{3.17}$$

n_{features} The number of features in your dataset
$X.var()$ The variance of the dataset's features. It's calculated independently for each feature

In the third phase, following the creation of a performance curve using the method mentioned earlier in [Sec3.5], the algorithm employs the SVR implementation when necessary.

The SVR procedure is activated when the motor operates beyond the $\pm 250\ \mu m$ range, or when the performance curve displays excessive oscillations, making it difficult to pinpoint the absolute maximum. In these scenarios, applying SVR to the existing performance curve and leveraging available data points for predictions can improve the trendline and aid in identifying the maximum point. Beyond this threshold, the system cannot provide satisfactory results. As previously explained, I have successfully addressed this limitation by integrating CNN algorithms and fuzzy control into a CL system, significantly extending the operational range and achieving outstanding outcomes. At the end of the third phase, it is crucial for the motor to be precisely positioned at the optimal focal point. There are occasional instances where the motor is not perfectly aligned with the desired focus point, typically with a deviation of around 10 μm.

In such cases, an additional phase is required to ensure the correct alignment of the motor on the Z-axis. Consequently, I begin the Fine focus phase to verify and adjust the positional accuracy.

3.7 Fine-focus

Introducing a comprehensive control strategy applied in the context of IntraCell, the objective is to achieve optimal image clarity. This entails a series of carefully orchestrated steps that span a wide range of depth variations, extending from deep to shallow focal points. The primary aim of all these maneuvers remains to achieve precise focus. As the process unfolds, it becomes evident that achieving sharp focus isn't just about addressing the larger variations; it's also about ensuring accuracy at finer, micrometer-scale intervals. This becomes especially crucial, as even slight movements can significantly impact optical focus.

Consequently, the procedure culminates in an advanced fine-focus algorithm. This algorithm builds upon the methodologies [Sec3.5] used earlier but tailors them for even greater precision. By employing reduced pitch intervals of 30 μm, the algorithm takes advantage of the motor's already optimized position. This focused approach ensures that the final fine-tuning process retains the clarity attained in earlier stages while zeroing in on the most minute adjustments required for optimal focus.

Chapter 4

Result

4.1 Initialization phase

The upcoming section illustrates the scenario where a first-time user initiates the autofocus command, commencing with the initial phase of capturing a series of images at predefined intervals. This pivotal step serves the purpose of enabling a comparative analysis of the collected images to accurately determine the focus direction.

[Fig. 4.1] presents a 3D visualization from within the IntraCell machine, showcasing the spatial positioning of the motor and camera along the Z-axis. Upon user activation of the motor control, it begins its movement from the initial starting point, ascending a distance of 100 μm.

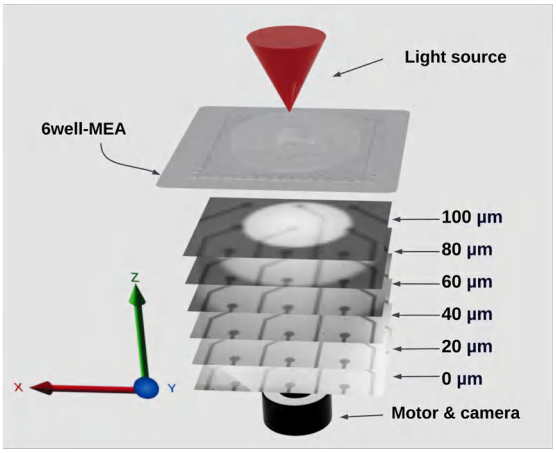


Figure 4.1 Immersive 3D demonstration: navigating autofocus image capture within the IntrCell

[Fig4.2] illustrates the scene captured with the camera positioned at zero μm, while the user initiates the autofocus command. This figure showcases a sequence of 21 images, each taken within a 100 μm range as the camera incrementally moves upward. The images were captured at 20 ms intervals, with 5 μm distance separating each consecutive shot.

Upon observing this sequence, a discernible trend emerges, characterized by a gradual deterioration in image sharpness and an increasing level of blurriness. This blurring effect becomes particularly pronounced during more substantial positional shifts.

However, detecting such changes can prove challenging when shifts occur within smaller intervals. Consequently, the intricate interplay of these factors complicates the task of identifying an image with optimal focus, especially as it approaches the in-focus boundaries.

In the subsequent steps of the autofocus process, the direction for adjustment is determined using the Determine direction function [Sec3.3.1], which, in this case, directs the autofocus process downwards. Subsequently, these same images are processed in the CL phase for further analysis.

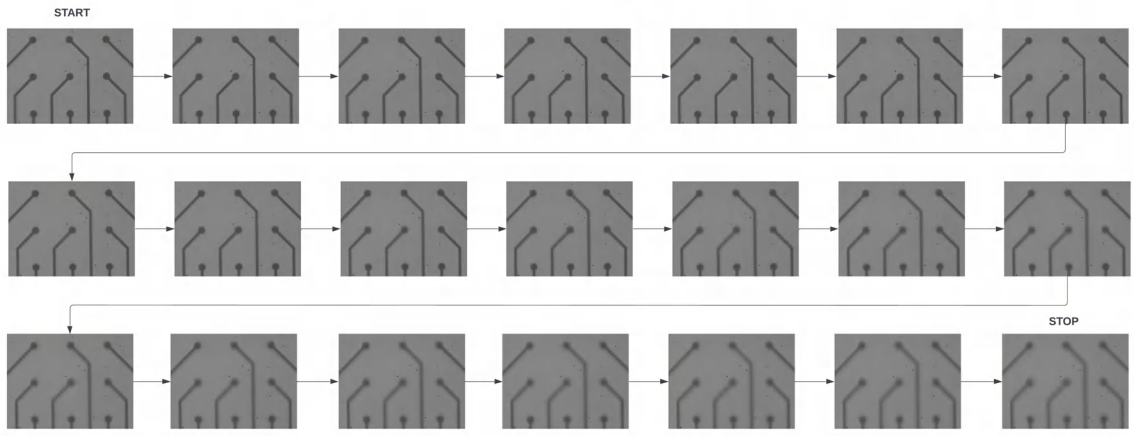


Figure 4.2 Image capture in continuous mode: spanning 0 to 100 μm with 5 μm interval, where 0 μm is the position of optimal focus.

The identical imaging method is employed for the range of -50 to 50 μm. In this scenario [Fig4.3], the motor moves beyond the focal point, leading to a reduction in image quality.

The identical imaging method is employed for the range of -100 to 0 μm. In this scenario [Fig4.4], the motor smoothly glides to the utmost focal point, enhancing the image quality in the process.

The series of images [Fig4.2,4.3,4.4] proceed to the second phase in the CL approach, wherein the images are subjected to classification and labeling using CNN.

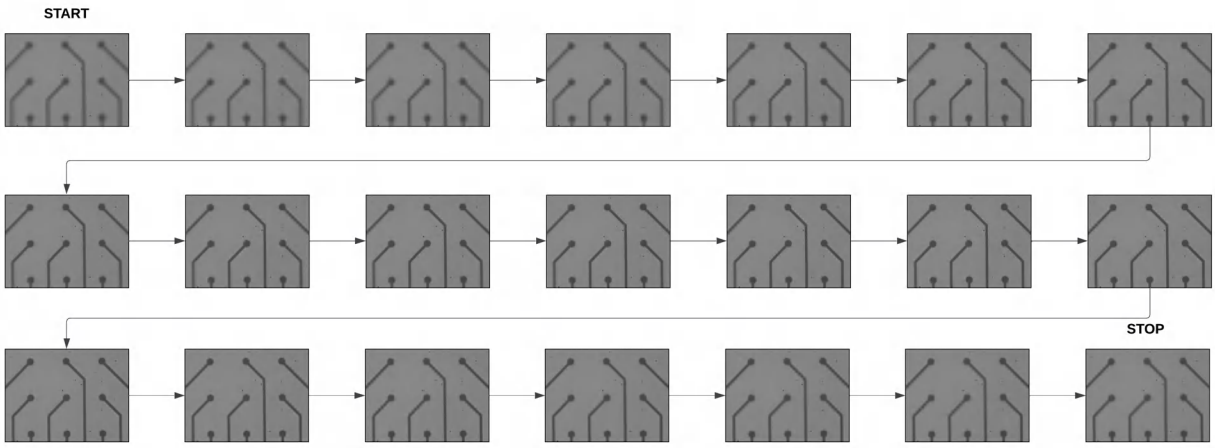


Figure 4.3 Image capture in continuous mode: spanning -50 to +50 μm with 5 μm interval

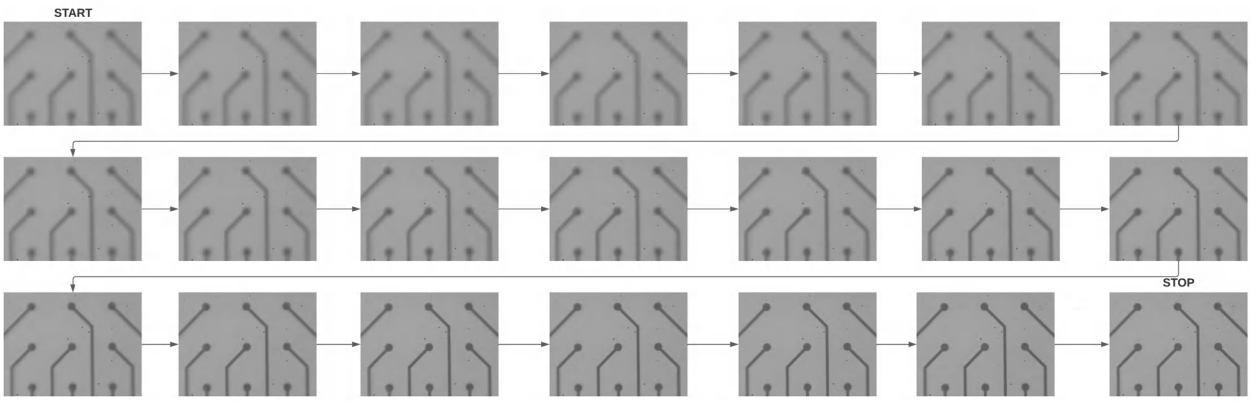


Figure 4.4 Image capture in continuous mode: spanning -100 to 0 μm with 5 μm interval, where 0 μm is the position of optimal focus

4.2 Closed Loop control

The components of the CL approach were previously detailed in Chapter 3. Subsequently, this chapter presents the outcomes, including the results of classification and labeling using CNN, along with a comparison of the Fuzzy control strategies based on the established rules.

4.2.1 CNN

The implementation of CNN significantly enhances the precision of image set classification acquired during distinct movements. The significance of image classification becomes particularly pronounced when algorithms operate across a spectrum ranging from broader to finer nuances.

In [Tab4.1] presents the key attributes and some results highlighted by (Sandler et al., 2018) within the CNN model MobileNetV2.

Characteristic	**Description**
Accuracy	Achieves good accuracy (around 72-75% top-1 accuracy) on benchmark datasets like ImageNet
Inference Speed	Optimized for high-speed inference. Suitable for real-time and resource-constrained applications
Use Cases	Mobile and embedded applications, real-time image classification, object detection, and other tasks with limited resources
Framework Compatibility	Compatible with TensorFlow, PyTorch, and other popular frameworks
Pretrained Models	Pretrained Weights are often available for transfer learning
Model Size	Smaller size compared to many other deep architectures Well-suited for devices with limited storage
Training	Typically trained using techniques like stochastic gradient descent with backpropagation, Often uses variations like learning rate schedules

Table 4.1 Characteristics of Lightweight Deep Learning Model

Illustrated in [Fig4.5] is a sequence of images captured within a range spanning from ± 600 down to ± 350 μm, which have been meticulously categorized using CNN technology as ***"Extremely Far"***.

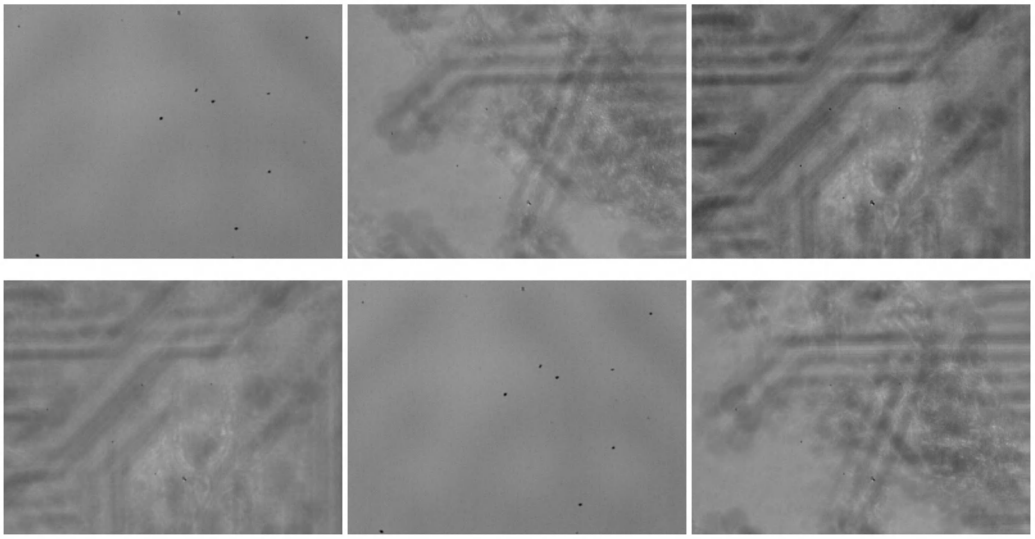


Figure 4.5 Image with label Extremely Far

In [Fig4.6], I encounter a collection of images bearing the ***"Far"*** label, as they were acquired within the interval of ± 350 to ± 250 μm.

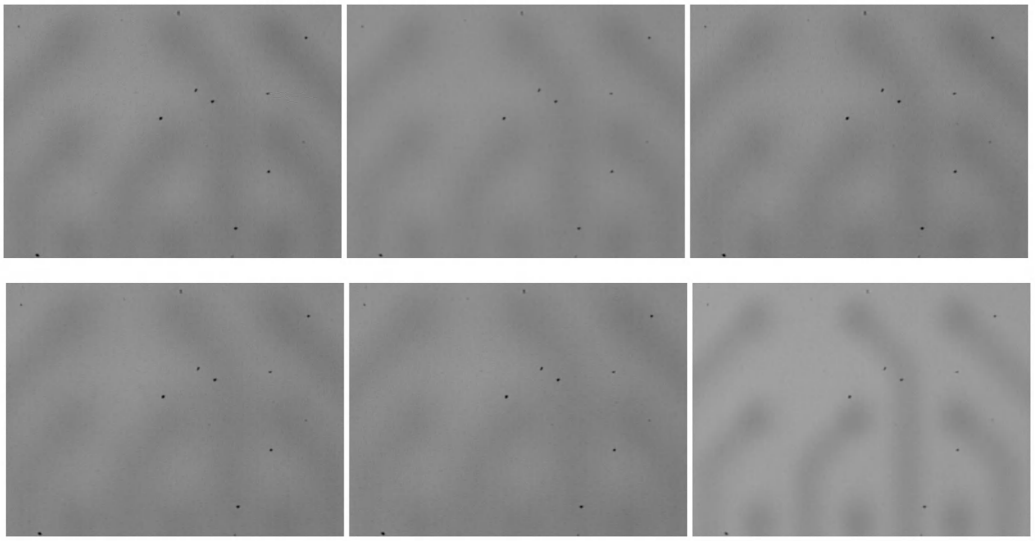


Figure 4.6 Image with label Far

Shifting the focus to [Fig4.7], the images herein designated as ***"Almost Far"*** are situated within the range of ± 250 to ± 100 μm.

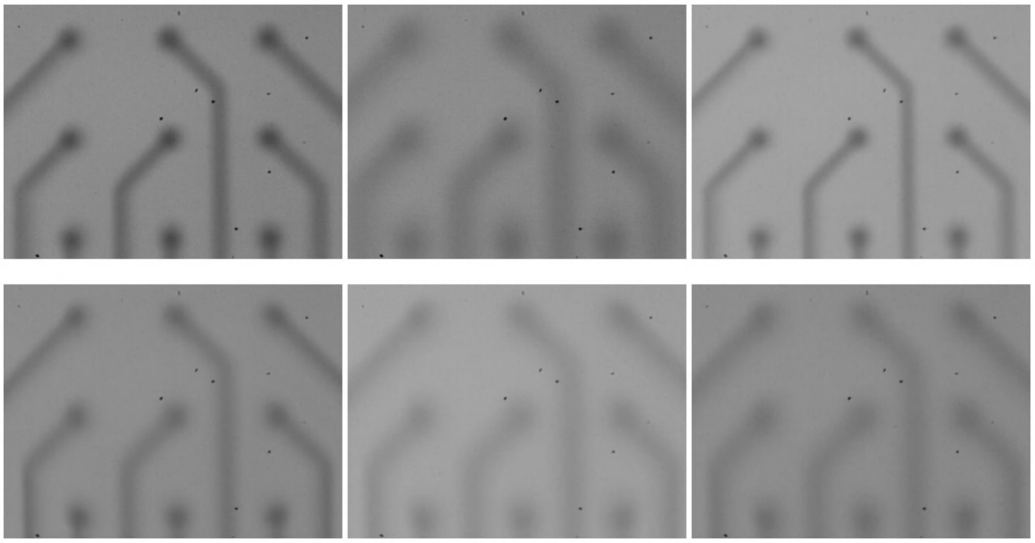


Figure 4.7 Image with label Almost Far

Advancing further, [Fig4.8] portrays a set of images gathered in the range of ± 100 up to ± 40 μm, duly classified as ***"Close"***.

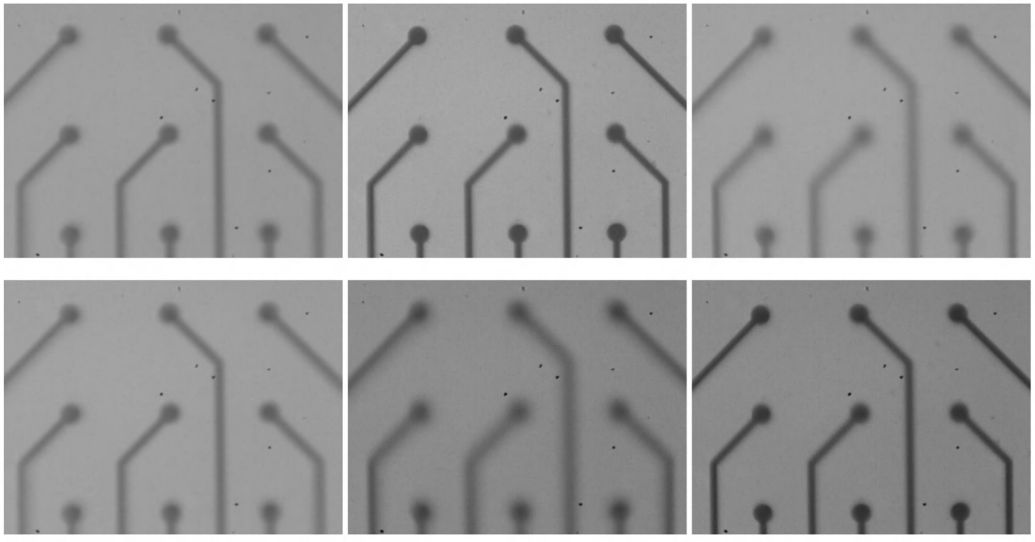


Figure 4.8 Image with label Close

Concluding this series, [Fig4.9] showcases images spanning the interval of ± 40 down to 0 μm, seamlessly affixed with the label ***"Extremely Close"***.

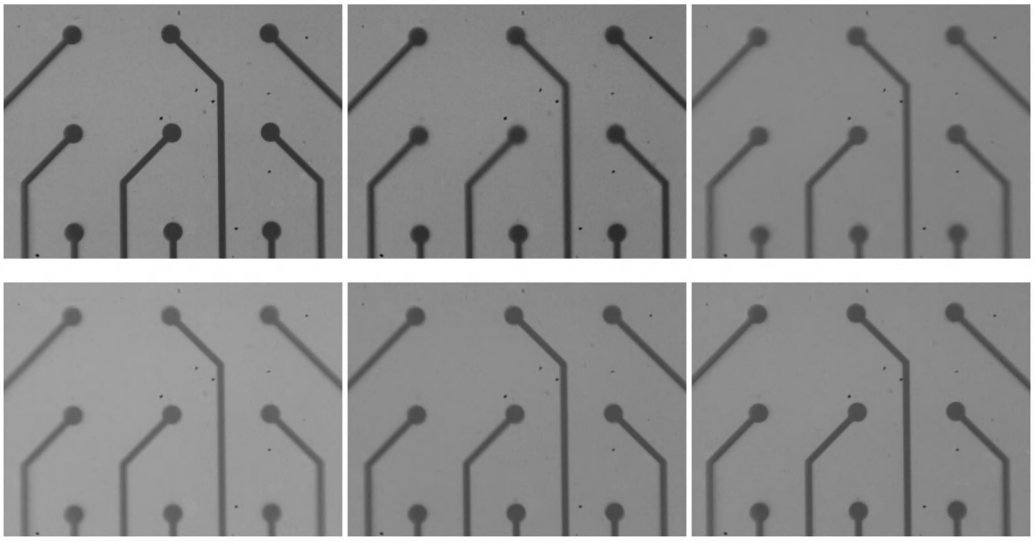


Figure 4.9 Image with label Extremely Close

This meticulous classification process facilitated by CNN empowers accurate contextual understanding and analysis of the depicted scenes.

The image classification process, as envisioned earlier [Sec3.4.4], involves employing a confidence interval for practical implementation.

In the second phase, the series of images [Fig4.2,4.3,4.4] captured during the initial phase undergoes a classification and labeling process within a CNN. These images are categorized as ***"Close"*** and ***"Extremely Close"*** cases, as previously mentioned.

Specifically, images falling within the ± 100 μm range are assigned these distinctive labels. These distance-based labels are subsequently input into a Fuzzy control system to generate motor action.

As illustrated in [Fig4.10], a Gaussian distribution is presented with a height of 1, a mean of 0, and a std of 1. This portrayal effectively showcases the distribution of classification ranges.

The outermost extremity of the curve is denoted as "Extremely Far", while the central segment is marked as "Extremely Close". The handling of nominations follows a consistent approach throughout this analysis.

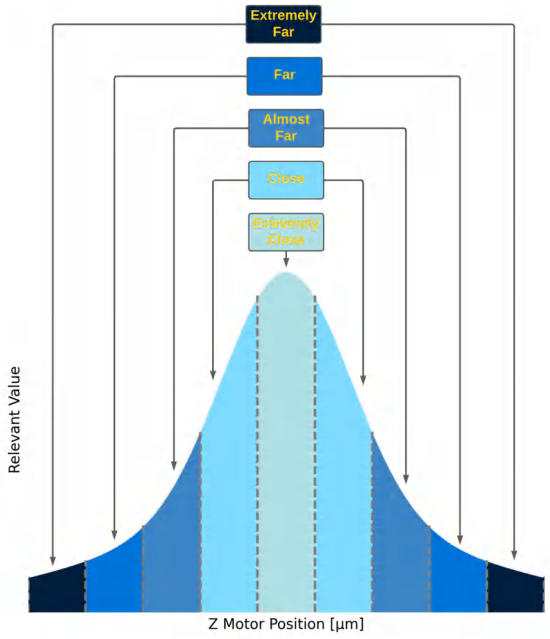


Figure 4.10 Performance Curve Labeling with CNN

Confusion Matrix

The outcomes, obtained from 45 experiments conducted using my dataset comprising over 3000 images, each paired with its corresponding label, have been meticulously summarized in the form of a confusion matrix [Fig4.11]. Along the vertical axis, there are the true labels corresponding to each individual image, while the horizontal axis denotes the predicted labels assigned to those images. The evaluation of the CNN classification through the utilization of the MobileNet V2 model reveals a commendable accuracy of 83% in my particular scenario.

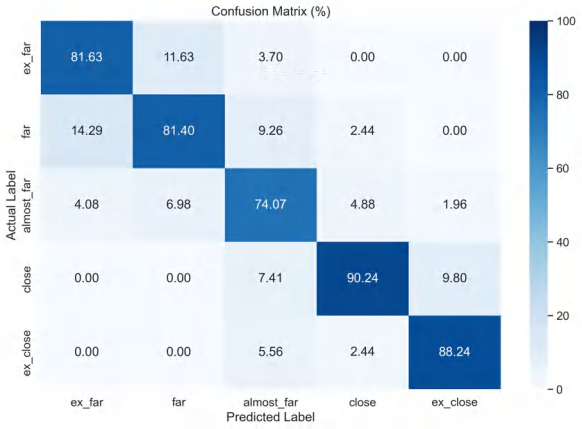


Figure 4.11 Enhancing predictive performance of CNNs through confusion matrix analysis

4.2.2 Fuzzy control

Following the initial phase of image acquisition, a sequence of images undergoes classification and labeling through the utilization of CNNs. Each specific label corresponds to distinct rules within the framework of fuzzy control.

Illustrated in (4.1) is the fundamental structure of the rules as applied in my fuzzy control scenario. Each rule comprises two "Antecedents", intricately linked to the initial label of the first image in the image list and the current label of the last image in the same list. These "Antecedents" essentially signify the starting point and the ongoing position of the motor's operation. Moreover, every rule is associated with a "Consequent", determining the resultant motor action to be executed. Notably, in regions distanced from the focal point, the motor action value demonstrably increases and conversely decreases when closer to the focal point.

$$\text{If distance}[\prime\text{Label1}\prime] \text{ and distance}[\prime\text{Label2}\prime] \text{ are true, then motor\_action}[\prime\text{value}[\mu\text{m}]\prime] \quad (4.1)$$

Specifically, when dealing with previously captured images processed through the CNN, the first series of images [Fig4.2] is assigned two labels: 'Label 1' denoting 'Extremely Close', and 'Label 2' indicating 'Close'. The criteria for label selection are based on the first and last images in the series. To elaborate, within a series of 21 images, only the first image (number 1) and the last image (number 21) are chosen for labeling and subsequent entry into the fuzzy control system. This approach is adopted primarily to expedite processing time, as labeling all images is unnecessary.

These two labels already signify a deterioration in the focus situation, suggesting the need to adjust in the opposite direction from the prior settings. In this scenario, the 'Consequent' value related to motor action remains at zero microns, indicating that the motor is already positioned at an acceptable distance to proceed with the next autofocus phase.

The rationale behind not exclusively relying on CNN and fuzzy control for the entire autofocus process is the limited reliability of this mechanism within fine focus ranges. However, they have exhibited commendable accuracy in broader focus ranges. The decreased accuracy of the CNN within the range of $\pm 250\ \mu m$ can be attributed to the insufficient dataset encompassing diverse image conditions necessary to identify images with optimal focus. This limitation is inherent to the algorithm, as CNN MobileNet V2 was originally designed for mobile applications, which typically have limited storage space for large datasets. A similar constraint is also encountered in the case of IntraCell due to a lack of memory capacity for dataset storage.

The same process is applied to the second set of images [Fig4.3], resulting in both the first label 'label1' and the second label 'label2' being designated as 'Close,' as the distance falls within the $\pm 50\ \mu m$ range. In this instance, the 'Consequent' value consistently remains at zero microns to transition to the subsequent autofocus phase. The primary consideration here is determining the direction, which was initially computed during the first step through the variance explained in [Sec3.3.1]. In this phase, it is merely verified.

For the third set of images [Figs4.4], 'label1' is assigned 'Close,' while 'label2' is designated as 'Extremely Close'. The 'Consequent' value remains at zero microns, and the direction is evident from the labels, indicating that the motor is proceeding in the correct direction.

4.3 Precise Focusing & Preprocessing

In the third phase, my primary objective is to achieve precise focus, as previously mentioned. This phase is initiated once the labeling and fuzzy control decisions have been established in previous phase, this signifies that the second phase has successfully positioned the motor within the designated range at $\pm 250\ \mu m$, thereby triggering the commencement of the third phase. This phase of the autofocus algorithm operates within a constrained activity in specified range. To achieve accurate focusing, I have implemented five distinct methods [Sec3.5]. The overarching concept behind these methods is to extract meaningful data from each image in the series, culminating in a list of data points(Relevant Values). These data points collectively form the Performance curve, a visual representation of the optical system's behavior.

This performance curve encapsulates virtually every change in each frame captured. On the Y-axis, it quantifies the Relevant Value extracted from each image, while on the X-axis, it represents the motor's position in micrometers. Each method has the capability to construct the performance curve, although some may require preprocessing steps to enhance accuracy.

Preprocessing is an essential step for certain methods like Sobel, Canny edge detection, and FFT, aimed at enhancing image quality prior to constructing the performance curve. Within the preprocessing phase, critical aspects include light adjustment and contrast enhancement.

Tests have been conducted to showcase the necessity and robustness of preprocessing for the mentioned methods. It's important to note that preprocessing is exclusively performed beforehand for these methods [Sec3.5.2,3.5.3, 3.5.4].On the other hand, noise processing [Sec3.5.5] and saturation analysis [Sec3.5.6] follow a distinct procedure.

4.3.1 Light adjustment

In [Fig4.12], it illustrates how changes in light intensity affect the performance curve during the autofocus process. When the Z motor moves the camera up and down, a sequence of 21 images is captured. This variation occurs because the light source is positioned at the top of the Z-axis. Even if you don't manually adjust the light during recording, an issue arises. Images taken at lower Z-axis positions receive less light than those taken at higher positions. In other words, moving upward on the Z-axis increases light quantity. Without filters to compensate for this effect, it can lead to errors in determining the maximum point on the performance curve. The details of light adjustment and its integration with different methods have been elucidated in [Sec3.5.1].

This curve was generated using the Sobel method. Among the three methods that require preprocessing, Sobel strikes a favorable balance between reliability and processing time, as indicated in [Tab4.4]. Consequently, it is the preferred choice for constructing the performance curve. Sobel excels in image analysis, outpacing the FFT method in speed and surpassing the Canny method in reliability, especially in scenarios characterized by significant light variations. These curves are the product of the Sobel method and the employed preprocessing technique. They illustrate variations in focus resulting from Z-axis shifts ranging from -50 μm to +50 μm. The optimal focus point should ideally be at zero μm. This range was selected due to the presence of an absolute maximum and near-symmetry around the zero-micron point in the curve. Consequently, grasping the proper operation of the preprocessing integration with Sobel becomes relatively straightforward.

[Fig4.12], indicates this zero point with a black dotted line. When light adjustment is employed, as shown by the red curve, the curve's peak aligns with the zero-micron point. However, without light adjustment, the curves behave differently due to variations in recorded light, making it challenging to identify an absolute focus maximum.

In particular, when the light intensity is set to the maximum value of 255, the curve exhibits markedly distinct behavior compared to other scenarios. Increasing intensity at this level results in overexposure risk, diminished contrast, glare, and reflections. There are detailed explanations of the impact of lighting on images in [Sec2.2.2].

The light values displayed in the [Fig4.12] indicate the light settings at ± 50 μm. However, with light adjustment, these values become normalized to mitigate the impact of light quantity fluctuations.

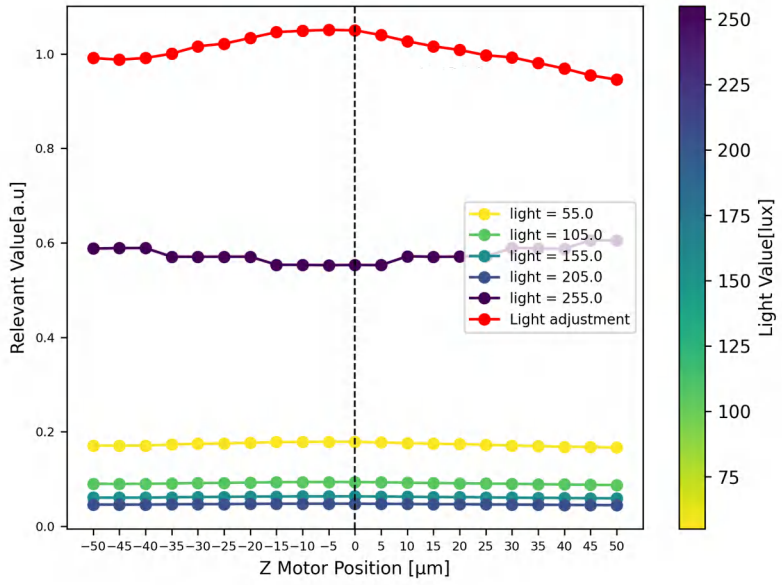


Figure 4.12 Light adjustment test: spanning -50 to 50 μm with 5 μm intervals

4.3.2 Contrast penalty

In [Fig4.13,4.14], it illustrates the dynamic fluctuations of the performance curve in response to changes in the alpha (α) value. The test was conducted repeatedly within the specified range, and the curves were generated using the Sobel method with preprocessing. The utilization of the Sobel method, among the three methods requiring preprocessing, remains consistent as previously described.

The significance of the coefficient α has been explained in [Sec3.10]. This coefficient plays a crucial role in mitigating the impact of contrast variations. Choosing a fixed α value has the potential to alter the algorithm's propensity to accurately locate the peak of the curve.

[Fig4.13] reveals that for alpha values between 0.5 and 1, the curve behaves acceptably in identifying the absolute maximum. Remarkably, even when the alpha value is automatically calculated (represented by the red line), it consistently coincides with zero microns (indicated by the black dotted line).

Subsequently, I conducted the test within the same range under varying image conditions, with the contrast level intentionally set to its minimum. In this scenario [Fig4.14], for constant alpha values above 0.5, the absolute maximum is expected to shift to the right, deviating from the true maximum corresponding to the optimal focus.

However, the red line in[Fig4.14], where alpha was automatically calculated, correctly identifies the absolute maximum, aligning with zero microns. The method for calculating the α value is explained in [Sec3.5.1].

This discrepancy is attributed to higher contrast levels causing an increase in the Alpha value, thereby favoring LPF. Conversely, reduced contrast stabilizes Alpha at 0.5, achieving a balance between the use of LPF and HPF, which indicates a limitation in the use of HPF.

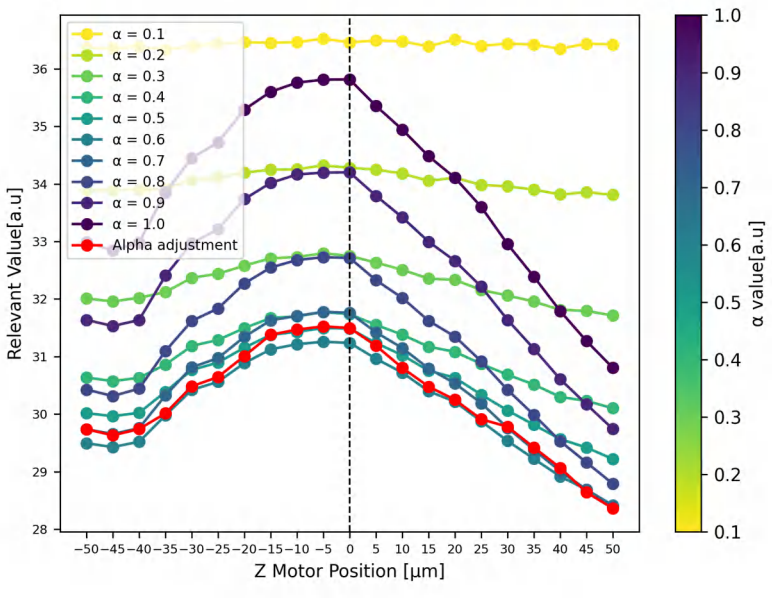


Figure 4.13 Contrast penalty test: adjust α value, image condition normal

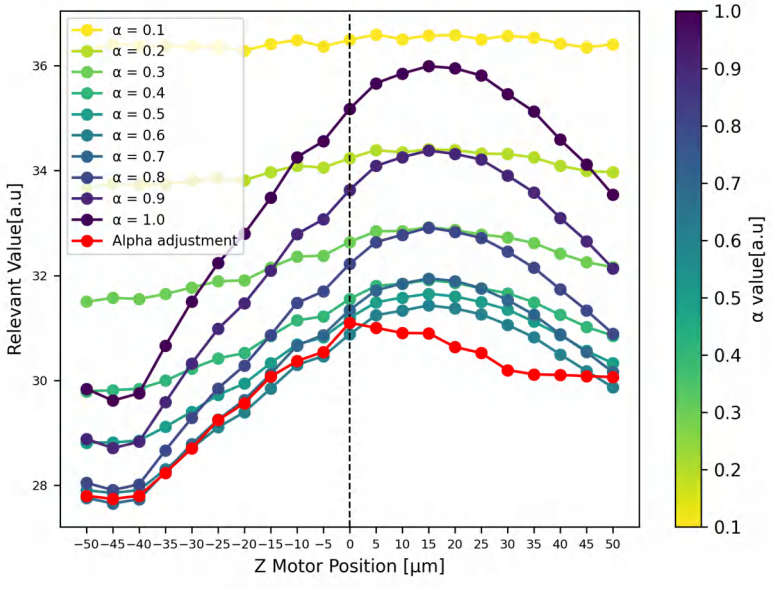


Figure 4.14 Contrast penalty test: adjust α value, image condition with min contrast

4.4 Constructing the performance curve

The performance curve serves as a valuable tool for comprehending the dynamic behavior of the optic's motion. Continuous image capture facilitates real-time analysis streamlining the examination of curve trends and allowing for the immediate generation of commands to adjust the Z motor's position, aligning it accurately with the focal point.

At the conclusion of the second phase, when the CNN generates labels such as 'Close' or 'Extremely Close', the fuzzy control takes these labels into account and makes the decision to halt the loop, proceeding to the third phase. During this phase, the same initial set of images [Fig4.2,4.3,4.4] is utilized since there have been no additional motor movements, and thus, no new series of images is required.

The images presented [Fig4.2,4.3,4.4] were captured under default settings, a deliberate choice made to bolster the algorithm's robustness and diminish reliance on specific parameters. The 'Default setting' in IntraCell refers to the initial values of Brightness level, Contrast level, Exposure level, and Light intensity level when the machine is powered on, with no alterations made. Detailed numerical values for each parameter in default setting can be found in [Tab4.2]. Given that the images form a cohesive series, the numerical values remain consistent across all 21 images.

Figure	Range$_{[\mu m]}$	Best image in$_{[\mu m]}$	Brightness$_{[intensitylevel]}$	Contrast$_{[a.u]}$	Exposure $_{[a.u]}$	Light$_{[lux]}$
4.2	(0,100)	0	133.76	15.21	0.64	124.97 ± 5
4.4	(-50,50)	0	135.56	13.21	0.66	126.71 ± 5
4.3	(-100,0)	0	160.91	11.64	0.76	152.68 ± 5

Table 4.2 Image characteristics across three distinct experiments. "a.u." stands for "arbitrary units", and it indicates that a quantity is measured without a specific unit of measurement. **Brightness** is typically represented in grayscale intensity values. In this case, the unit is often referred to as "grayscale levels" or simply "intensity levels". The values are usually expressed as integers ranging from 0 to 255 in an 8-bit grayscale image, where: 0 represents complete black (no brightness), 255 represents complete white (maximum brightness), and Values in between represent varying levels of gray or brightness.**Contrast** The unit of measurement for contrast, as calculated using the standard deviation of pixel values, is not expressed in a specific unit with a clear physical meaning. Instead, it is a relative measure of the variability in pixel intensities within the image.**Exposure**The unit of measurement for exposure, as calculated based on pixel intensities and histogram analysis, is not expressed in a specific unit with a clear physical meaning. Instead, it's a relative measure of how well the image is exposed, based on the distribution of pixel intensities. **Light** can refer to different aspects, including illuminance and luminous flux. The unit [lux] measures illuminance, which is the amount of light that falls on a surface. It's a measure of the intensity of light per unit area.

At the commencement of developing the autofocus algorithm, a significant challenge arose due to the absence of a definitive reference point for selecting the optimal focus in images. This challenge emerged from the various parameters outlined in [Tab4.2], each capable of introducing distinct effects on the image. These effects, if not accounted for, could potentially hinder the algorithm's ability to accurately identify the correct focus, rendering traditional a priori algorithms inadequate for ensuring robust performance. Consequently, a

more sophisticated approach was required, one that could reliably distinguish images with exceptional focus from the rest.

To overcome this limitation, a pragmatic solution was adopted. Images exhibiting superior focus were subjectively identified through manual observation. The naked eye served as a discerning tool to single out images with optimal focus characteristics. Once identified, the Z-motor was meticulously positioned at this specific juncture, functioning as the zero point. This point was crucial as it represented the pinnacle of optical clarity within the image. Subsequently, data recording commenced, capturing a comprehensive dataset that would be instrumental in constructing a detailed curve for subsequent evaluations and analyses.

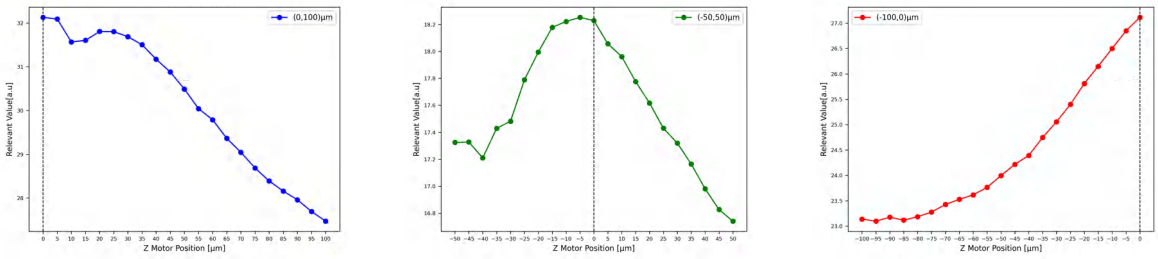
4.4.1 Sobel filtering and correction

Sobel operator and preprocessing techniques underwent testing within three distinct ranges to facilitate the creation of a precision-focused curve. The primary objective of this analysis revolved around generating a curve within a narrow range that centers around the point of focus. To achieve this, I opted for a range spanning from -100 to +100 μm. This range was divided into three intervals, amplifying the granularity of my analysis for enhanced precision. It's important to reiterate that as one moves from the lower to the upper regions, the recorded light intensity increases due to the elevated positioning of the light source. However, the strategic decision to advance in 100 μm increments helps mitigate this influence.

The initial range spans from 0 to 100 μm [Fig4.2]. In this instance, it's imperative for the maximum point of the curve to be positioned at the outset of the curve. In [Fig4.15,a], it illustrates the variation in a Relevant Value by translating it 100 μm from the bottom to the top. The dashed black line denotes the point corresponding to the maximum absolute value, which conveniently aligns with the peak of the curve. Notably, the rapid fluctuations observed within the 5 to 15 μm range stem from disturbances within the system, often attributed to movements of the worktable. To address this issue, I employed SVR algorithms to regenerate a more precise curve, as previously explained in [Sec3.6.1]. The rationale for utilizing SVR in conjunction with the performance curve is to enhance reliability.

In [Fig4.15,b], it replicated the same experiment but extended the range from -50 to 50 μm [Fig4.3]. Notably, the peak of the curve coincides with the zero *mum* point. Similarly, a disturbance is noticeable within the range of -45 to -35 μm. To mitigate these disturbances, I applied the aforementioned method.

[Fig4.15,c] showcases a distinct change in the performance curve, this time within a compressed range spanning from -100 to 0 μm [Fig4.4]. Once again, the peak coincides with the zero μm point.



(a) (b) (c)

Figure 4.15 Generate performance curves by applying the Sobel operator three different intervals between ± 100 μm

Key to this method's success is the precise execution of preprocessing, ensuring that any fluctuations stemming from parameter changes do not affect the integrity of the performance curve. This allows us to pinpoint the absolute maximum, representing the optimal image focus position. Rapid changes within intervals smaller than 15 μm are negligible and can be addressed during subsequent development phases.

The time required to generate the performance curve across a series of 21 images averages 1.54 ± 0.5 seconds

4.4.2 Canny edge detector filtering

The second approach employed the Canny edges detector, and the identical experiments were meticulously replicated. Illustrated in [Fig4.16,a] at the outset of the curve, it finds the maximum point coinciding precisely with the zero μm position. This curve exhibits rapid fluctuations near the maximum focus point, a characteristic inherent to the method's implementation. In this method, I employ variance to calculate the Relevant Value for each image, a metric that proves highly sensitive to changes when compared to the mean used in the Sobel operator method.

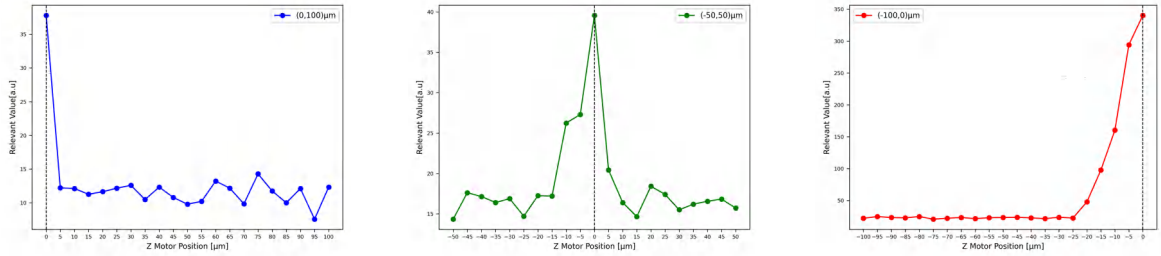
Within the range of 35 to 100 μm, a notable degree of oscillation is observed, primarily due to the use of variance. Nevertheless, since the maximum value detected far surpasses these variations in magnitude, they can be safely disregarded. It's worth noting that such fluctuations may become significant in finer measurement ranges.

In [Fig4.16,b], it depicts a movement spanning from -50 to +50 μm. While the maximum point is accurately identified, the curve's overall trend lacks reliability, a challenge that cannot be resolved through the implementation of SVR. In contrast, [Fig4.16,c] showcases the curve

within a more compressed interval, ranging from -100 to 0 μm, and the maximum value is determined accurately without any unusual oscillations.

Hence, it's important to recognize that this method exhibits greater effectiveness within larger intervals, where image-to-image variations are more substantial, yielding a performance curve of higher quality.

The time required to generate the performance curve across a series of 21 images averages 1.22±0.5 seconds.



(a) (b) (c)

Figure 4.16 Generate performance curves by applying the Canny edges detector three different intervals between ±100 μm

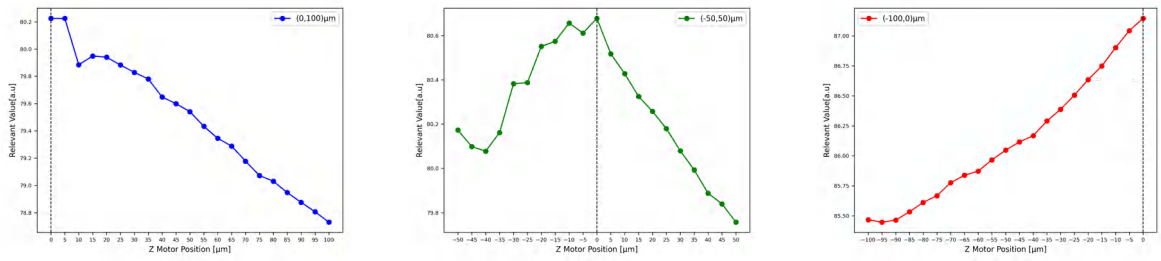
4.4.3 Fast Fourier Transform

The outcomes derived through the FFT approach within the same experiment reveal an interesting pattern. In [Fig4.17,a], it successfully located the maximum point at the correct position, and the curve's behavior using the FFT method resembles that of the Sobel operator. In both cases, I calculated the Relevant Value for each image using the average. It's worth noting a similar drop in performance within the range of 5 to 10 μm, attributed to disturbances.

In [Fig4.17,b], it demonstrates the FFT method's capabilities within the range of ±50 μm. Notably, while the maximum point is correctly identified, the curve exhibits some variability and lacks precise trend representation. This issue can be mitigated with the application of SVR.

In [Fig4.17,c] portrays the curve's behavior within the range of +100 to 0 μm, and once again, it successfully located the maximum point.

The time required to generate the performance curve across a series of 21 images averages 4.80±0.8 seconds



(a) (b) (c)

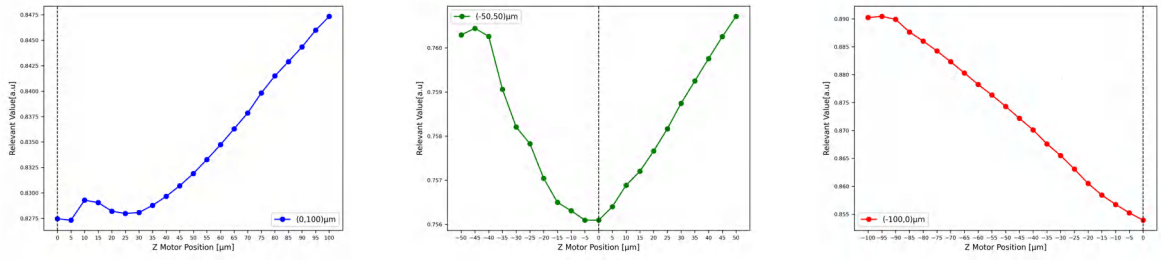
Figure 4.17 Generate performance curves by applying the FFT three different intervals between $\pm 100\ \mu m$

4.4.4 Noise processing

In the Noise Processing approach, the deliberate omission of preprocessing steps is a distinguishing feature aimed at preserving the inherent noise characteristics within images. The resulting performance curve generated by this approach exhibits a noteworthy feature, a discernible **minimum point** that correlates with images of superior quality. This curve's construction relies on the computation of the SNR. A decrease in SNR signifies an increasing dominance of noise relative to the signal within the image. Conversely, an increase in SNR indicates a stronger signal compared to the noise, yielding an image of heightened clarity and excellence.

The fundamental concept underlying this approach is rooted in the observation that images beset by noise, including disruptive elements like flicker noise, as well as images boasting impeccable focus, tend to exhibit a conspicuous level of noise. To validate this concept, identical datasets [Fig4.2,4.3,4.4] underwent meticulous analysis. Illustrated in [Fig4.18,a], it accurately located the minimum at the zero micron point, with a similar fluctuation observed within a compressed range of 5 to 10 μm, attributed to disturbances. Similarly, in [Fig4.18,b], the minimum was correctly identified at the curve's midpoint in a range spanning $\pm 50\ \mu m$, a result [Fig4.18,c] that repeats within another range spanning from -100 to 0 μm and the minimum was found successfully at the end of the curve.

While this method effectively captures the curve's trends within these three intervals, it's important to note that the time required to generate the performance curve across a series of 21 images averages 27.92 ± 2 seconds. This can introduce flicker noise, further contributing to image noise. Consequently, this issue renders the algorithm less robust when compared to alternative methods.



(a) (b) (c)

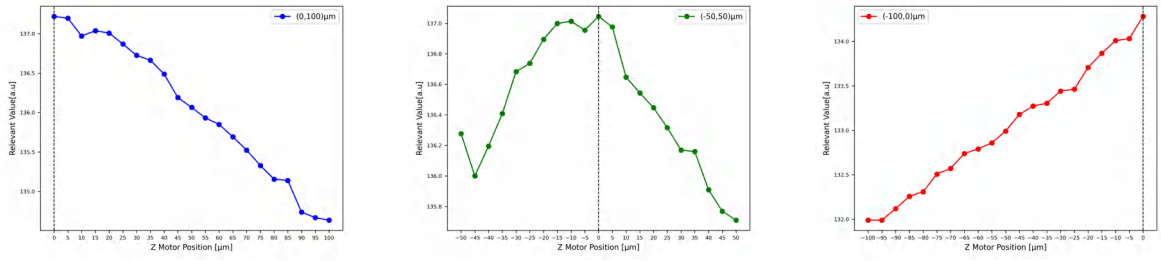
Figure 4.18 Generate performance curves by applying the Noise processing three different intervals between $\pm 100\ \mu m$

4.4.5 Saturation analysis

The Saturation analysis approach revolves around the examination of data exclusively from saturation channels, with subsequent application of filtering and normalization. The significance of this procedural step is underscored by the results obtained. As illustrated in [Fig4.19,a], it successfully identified the maximum at the zero μm point. However, it's important to note that the curve exhibits substantial variability despite the use of the average to calculate the Relevant Value. The same experiments were replicated at different intervals, ranging from $\pm 50\ \mu m$ and from -100 to 0 μm, with the results depicted in [Fig4.19,b] and [Fig4.19,c], respectively. In both cases, the maximum was accurately located.

Within this method, the absolute maximum corresponds to an image with excellent focus. Notably, this method lacks reliability on three sides of the curve, displaying a high number of significant variations. This underscores the challenges associated with selecting the saturation channel to identify images with excellent focus, despite successfully locating them.

The time required to generate the performance curve across a series of 21 images averages 0.39 ± 0.15 seconds.



(a) (b) (c)

Figure 4.19 Generate performance curves by applying the Saturation analysis in three different intervals between $\pm 100\ \mu m$

4.4.6 Comparative analysis of methods

In pursuit of identifying the optimal algorithm characterized by both superior accuracy and unwavering precision among the array of five methods, a systematic approach was adopted. This involved subjecting each method to repetitive experimentation across various parameter ranges and settings. Detailed information regarding the experimental setup for each iteration can be found in [Tab4.3].

Range$_{[\mu m]}$	Condition	Brightness$_{[intensitylevels]}$	Contrast$_{[a.u]}$	Exposure $_{[a.u]}$	Light$_{[lux]}$
(150,250)	default	143.75	9.38	0.72	134.82±5
(150,250)	max brightness	162.32	8.80	0.77	154.10±5
(150,250)	max contrast	139.58	13.76	0.74	130.76±5
(50,150)	default	138.94	16.60	0.67	130.15±5
(50,150)	max brightness	161.32	15.74	0.76	153.20±5
(50,150)	max contrast	136.54	22.09	0.72	127.90±5
(10,110)	default	136.23	15.91	0.65	127.44±5
(0,100)	max brightness	163.20	15.20	0.75	154.93±5
(0,100)	max contrast	139.11	24.08	078	130.63±5
(-10,90)	default	141.33	17.26	0.67	132.55±5
(-10,90)	max brightness	160.34	6.09	0.76	152.28±5
(-10,90)	max contrast	136.72	22.54	0.73	128.16±5
(-30,70)	default	135.78	15.26	0.65	126.98±5
(-30,70)	max brightness	159.12	15.14	0.75	151.01±5
(-30,70)	max contrast	136.49	21.95	0.73	127.91±5
(-50,50)	max brightness	158.89	13.40	0.75	150.71±5
(-50,50)	max contrast	134.84	19.33	0.71	126.16±5
(-100,0)	max brightness	158.32	11.23	0.75	150.03±5
(-100,0)	max contrast	134.62	14.97	0.71	125.84±5
(-150,-50)	default	135.55	9.43	0.76	134.36±5
(-150,-50)	max brightness	160.91	9.12	0.77	152.15±5
(-150,-50)	max contrast	137.46	12.59	0.75	128.60±5
(-250,-150)	default	137.01	7.49	0.66	128.00±5
(-250,-150)	max brightness	159.39	7.59	0.75	151.08±5
(-250,-150)	max contrast	137.46	12.59	0.75	128.60±5

Table 4.3 The image conditions pertain to the 25 conducted experiments, each comprising a dataset composed of 21 distinct images.

Notably, the experiment was conducted in triplicate for every parameter range, yielding three distinct scenarios. These scenarios encompassed:

1. The default configuration of parameters
2. Set to maximum brightness
3. Set to maximum contrast

While parameters such as light and exposure remained relatively stable throughout the experimentation process, it's important to acknowledge that modifications were minimal. This selection arises from the issue of compromised image quality when configuring extreme values for light and exposure. Such settings render the analyses unviable due to the resulting suboptimal image conditions.

Choosing the optimal method

The experimentation involved 25 repetitions for each method. These repetitions were conducted under diverse image conditions, thereby encompassing an extensive range of potential scenarios. This exhaustive approach was designed to encapsulate all feasible scenarios, ensuring a comprehensive evaluation of the methods' performance.

The displayed table [Tab4.4] provides insights gleaned from 125 tests. The findings highlight the superior speed of the saturation analysis method, boasting an impressive execution time of merely 0.39 seconds with a minor standard deviation(STD) of 0.15 seconds. Notably, this expeditious performance can be attributed to its lack of preprocessing coupled with its focus on analyzing a single image channel.

Conversely, the noise processing method emerges as the most time-consuming approach for constructing the performance curve, necessitating 27.92 seconds per iteration with an STD of 2 seconds. This protracted duration is due to the initial step of generating noise to intentionally distort the image, an imperative aspect of this technique.

For the various methods assessed, I conducted a comprehensive analysis that included accuracy, root mean square error, and error rate calculations to provide a more precise evaluation of each method's performance.

To calculate the accuracy (4.2) and error rate (4.3), I employed the following formulas.

The root mean square error (4.4) was calculated by measuring the difference between the current motor's position on the Z-axis "$\hat{d}_i$" and the zero micron reference point "d_0". In simpler terms, this involved comparing the current motor position obtained through a method with the actual target position, where the zero micron position indicates the true focal point. Initially, due to the lack of reliable algorithms for automatically determining the precise motor position corresponding to a focused image, the motor had to be manually positioned at the zero micron distance to identify this disparity.

$$Accuracy = \left(\frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Trials} \right) \times 100 \tag{4.2}$$

$$Error\ Rate = \left(\frac{Number\ of\ Incorrect\ Predictions}{Total\ Number\ of\ Trials} \right) \times 100 \tag{4.3}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{d}_i - d_0)^2} \tag{4.4}$$

$\hat{d}_i$ The current distance of the motor relative to the optimal focus point, specifically the zero micron reference point.
d_0 The optimal focus point, the zero micron point.
n represents the number of trial.

The accuracy credentials of the Sobel filtering and correction method take center stage, boasting an impressive 92% precision rate. Within 25 trials, merely two errors surfaced, underscoring its efficacy. On the other hand, the saturation analysis method yields an accuracy of 72%, denoting seven errors across the same number of trials.

Delving deeper, the Root Mean Square Error (RMSE) serves as a yardstick to gauge the magnitude of errors in instances where higher accuracy is paramount. For the Canny edge detector filtering and noise processing, accuracy levels reached 76%, corresponding to six errors in 25 trials.

In this context, the method exhibiting the least RMSE assumes the mantle of reliability. Consequently, between the aforementioned approaches, canny edge detector filtering prevails as the more dependable choice, surpassing noise processing with an equivalent RMSE of 13.69 μm.

Method	Allocated time$_{[s]}$	Accuracy$_{[\%]}$	RMSE$_{[\mu m]}$	ER $_{[\%]}$
Sobel filtering and correction	1.54±0.5	92	7.91	8.0
Canny edge detector filtering	1.22±0.5	76	13.69	24
Fast Fourier Transform	4.80±0.8	88	11.91	12
Noise processing	27.92±2	76	16.77	24
Saturation analysis	0.39±0.15	72	23.57	28

Table 4.4 Results of applying methods to the dataset. The allocated time signifies the duration required by the algorithm to generate the performance curve for a dataset comprising 21 images. Accuracy and Error Rate (ER) are quantified on a percentage scale. The RMSE is gauged in μm, indicating its magnitude when deviating from the actual peak point.

In a broader perspective, both Sobel filtering and correction, as well as FFT methods, shine as the two most precise techniques, showcasing minimal RMSE values. These distinguished methodologies have been earmarked for further integration into the algorithmic framework for constructing performance curves, synergizing seamlessly with SVR approach.

In [Fig4.20] depicts an Error Heatmap, wherein the Y-axis showcases the different methods employed, while the X-axis corresponds to the range within which each method was tested. The range is compressed between $\pm 250\ \mu m$. This particular heatmap elucidates the error distribution associated with each individual method. Shades of lighter colors within the heatmap signify lower error occurrences, whereas progressively darker hues indicate higher error frequencies.

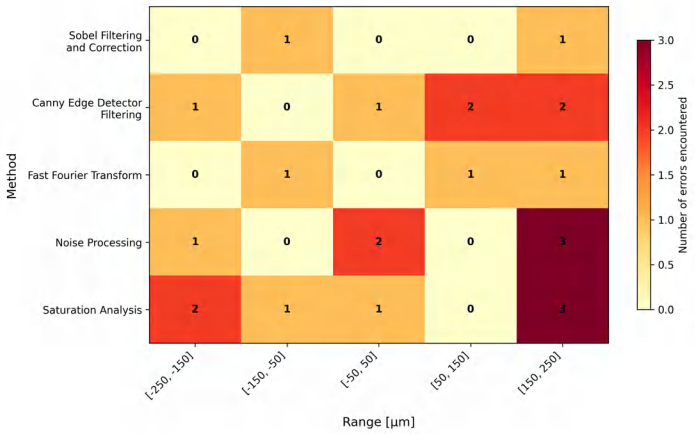


Figure 4.20 Error heatmap for each method within specific range

4.5 Optimizing Performance Curve using Support Vector Machines

In the preceding section, I discussed the outcomes pertaining to the initial, second, and partial results for the third phase. In this section, I delve into the results concerning SVR as it pertains to the completion of the third phase.

This method serves the purpose of regenerating a new performance curve, as previously established in the preceding section. It's crucial to highlight that the application of SVR is not always a requirement. As elucidated in [Sec4.4.1], SVR is selectively employed under particular conditions. These include scenarios where the performance curve exhibits significant oscillations due to disturbances created by motor vibration, or when the motor's output does not align with the desired range of $\pm 250\ \mu m$. Such misalignment to reach desired range may arise during the transition from the second to the third phase, wherein the image classification process via CNN occasionally mislabels data, leading to imprecise motor control actions via fuzzy logic. It is essential to emphasize that the incidence of this

error is relatively low, as demonstrated by the confusion matrix [Fig4.11]. Nevertheless, in the event that such errors do occur, the implementation of SVR proves instrumental in rectifying this anomaly.

If none of the aforementioned scenarios apply, the algorithm proceeds to the next phase without invoking SVR. This approach is undertaken to minimize downtime and optimize efficiency.

4.5.1 SVR

The predictions derived from the SVR algorithm are underpinned by curves generated through a combination of Sobel, and FFT with preprocessing. These facets collectively contribute to a heightened level of accuracy and dependability in the predictions. Within the SVR framework, three distinct curve types emerge:

1. Mid Case: This scenario occurs when the motor transitions between focus points, leading to a peak situated at the midpoint of the curve. Consequently, this configuration entails adjustments to the position without introducing new predictions. As a result, outcomes akin to prior instances are observed [Fig4.15,b,4.17,b].

2. Head Case: In this circumstance, the maximum value is situated at the inception of the curve.

[Tab4.5] displays six datasets characterized by the following properties. The initial three rows in Table [Tab4.5] depict identical performance curve scenarios [Fig4.15,a,4.17,a] within the range of 0 to 100 μm, varying solely in terms of image conditions as illustrated in the table. Conversely, the subsequent rows signify a distinct distance range, spanning from 50 to 150 μm.

index	Range$_{[\mu m]}$	Condition	Brightness$_{[intensity level]}$	Contrast$_{[a.u]}$	Exposure $_{[a.u]}$	Light$_{[lux]}$
a	(0,100)	default	144.57	6.47	0.72	135.61±5
b	(0,100)	max brightness	162.84	6.07	0.77	154.73±5
c	(0,100)	max contrast	141.96	20.11	0.79	133.02±5
d	(50,150)	default	142.10	8.55	0.67	133.12±5
e	(50,150)	max brightness	162.44	7.60	0.75	154.10±5
f	(50,150)	max contrast	139.58	13.76	0.74	154.25±5

Table 4.5 The image conditions pertain to the six conducted experiments, each comprising a dataset of 21 distinct images.

These datasets were the foundation for constructing performance curves employing Sobel and the FFT algorithm, integrated with SVR. The objective of incorporating SVR is to

effectively manage curve segments that exhibit significant noise while predicting potential curve maxima to identify optimal focal points.

In [Fig4.21], validated plots were generated using the Sobel algorithm, alongside predictions derived from SVR. The image indices align with those in the accompanying [Tab4.5]. The blue line signifies the curve produced using the Sobel algorithm, while the red line signifies the regenerated performance curve based on SVR's predictive output.

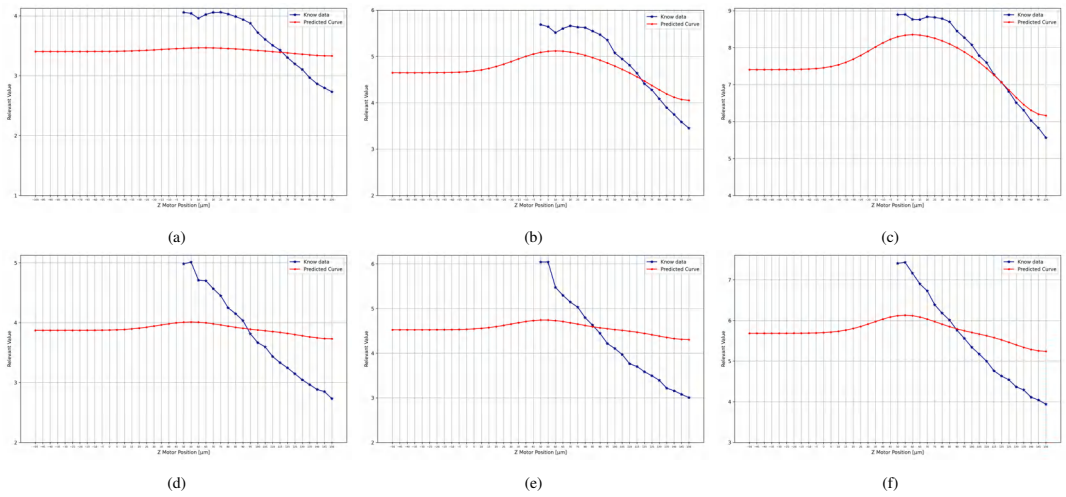


Figure 4.21 Exploring Head case through six experiments: Utilizing the Sobel algorithm and SVR for performance curve generation

In [Fig4.22], validated plots were generated using the FFT algorithm, alongside predictions derived from SVR. The image indices align with those in the accompanying [Tab4.5]. The orange line signifies the curve produced using the FFT algorithm, while the red line signifies the regenerated performance curve based on SVR's predictive output.

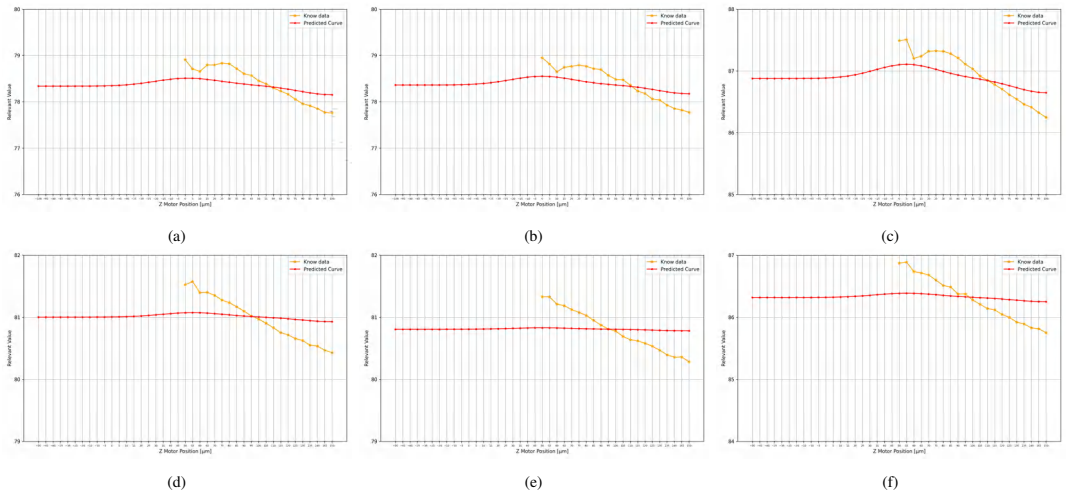


Figure 4.22 Exploring Head case through six experiments: Utilizing the FFT algorithm and SVR for performance curve generation

It is crucial to highlight the significance of the cornering sections characterized by rapid oscillations followed by a return to the previous trend. Clearly, these anomalies stem from various factors, including excessive machine vibrations and the absence of a suitable laboratory table for recording purposes.

3. Tail Case: Conversely, the maximum value finds its position at the terminal end of the curve.

[Tab4.6] displays six datasets characterized by the following properties. The initial three rows in Table [Tab4.6] depict identical performance curve scenarios [Fig4.15,c,4.17,c] within the range of -100 to 0 μm, varying solely in terms of image conditions as illustrated in the table. Conversely, the subsequent rows signify a distinct distance range, spanning from -150 to -50 μm.

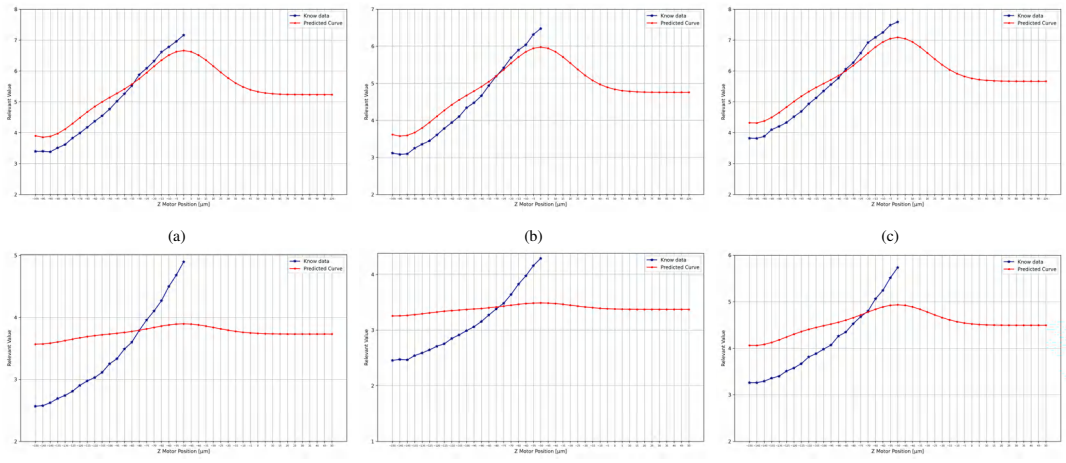
index	Range$_{[\mu m]}$	Condition	Brightness$_{[intensitylevels]}$	Contrast$_{[a.u]}$	Exposure $_{[a.u]}$	Light$_{[lux]}$
a	(-100,0)	default	133.44	8.67	0.75	124.50±5
b	(-100,0)	max brightness	158.42	8.76	0.66	155.20±5
c	(-100,0)	max contrast	136.04	21.48	0.59	121.90±5
d	(-150,-50)	default	136.57	6.13	0.65	127.58±5
e	(-150,-50)	max brightness	158.86	6.12	0.73	150.56±5
f	(-150,-50)	max contrast	139.58	13.76	0.74	130.76±5

Table 4.6 The image conditions pertain to the six conducted experiments, each comprising a dataset composed of 21 distinct images.

These datasets were the foundation for constructing performance curves employing Sobel and the FFT algorithm, integrated with SVR. The objective of incorporating SVR is to effectively manage curve segments that exhibit significant noise while also predicting potential curve maxima to identify optimal focal points.

In [Fig4.23], validated plots were generated using the Sobel algorithm, alongside predictions derived from SVR. The image indices align with those in the accompanying [Tab4.6]. The blue line signifies the curve produced using the Sobel algorithm, while the red line signifies the regenerated performance curve based on SVR's predictive output.

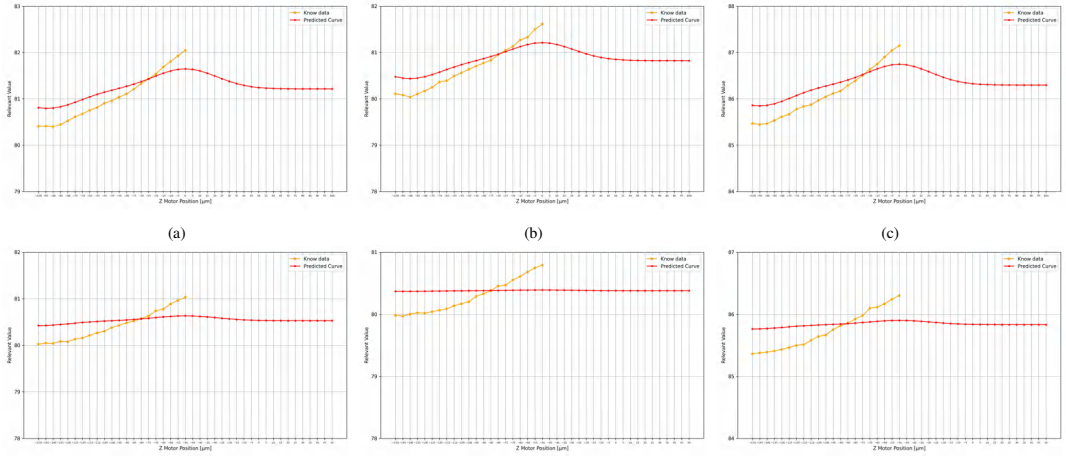
In [Fig4.24], validated plots were generated using the FFT algorithm, alongside predictions derived from SVR. The image indices align with those in the accompanying [Tab4.6]. The blue line signifies the curve produced using the FFT algorithm, while the red line signifies the regenerated performance curve based on SVR's predictive output.



(a) (b) (c)

(d) (e) (f)

Figure 4.23 Exploring Tail case through six experiments: Utilizing the Sobel algorithm and SVR for performance curve generation



(a) (b) (c)

(d) (e) (f)

Figure 4.24 Exploring Tail case through six experiments: Utilizing the FFT algorithm and SVR for performance curve generation

Comparing SVR results between Head and Tail cases: Integrating Sobel and FFT algorithms

The integration of SVR with both Sobel and FFT algorithms yields consistent outcomes in both the Head and Tail cases. Within the compressed range of $\pm 100\ \mu m$, the algorithms consistently identify maximum points with a 95% accuracy rate.

In the broader range of $\pm 200\ \mu m$, this accuracy remains respectable at 80%, with an associated error of $30 \pm 5 \mu m$. For distances under 30 μm, introducing a suitable offset resolves the discrepancies within wider ranges. To achieve even greater precision, Fine-Focus algorithms can be implemented for finer ranges.

Notably, the FFT algorithm's major limitation lies in its time consumption. Utilizing FFT in the autofocus algorithm typically requires around 30±5 seconds to achieve the desired output, whereas the same task using Sobel's algorithm is accomplished in a mere 10±2 seconds.

Given the crucial criterion of speed in addition to precision and accuracy for autofocus algorithms, the Sobel algorithm emerges as the preferred choice for subsequent phases.

4.6 Fine Focus

The final phase is dedicated to ensuring that the last result presented from the three preceding phases accurately reflects an image with correct focus.

In this phase, the algorithm outlined in [Alg2] is employed to recalibrate the direction, shifting the motor Z within a 30 μm range and capturing images at 5 μm intervals, resulting in a set of images constructed from 6 frames. It's worth noting that the Z motor's resolution is limited to 5 μm, prohibiting smaller steps.

Subsequently, only the Sobel method and preprocessing algorithm are applied to generate a new performance curve within this 30 μm range. Utilizing the same criteria for identifying the performance curve's maximum, the Z motor is repositioned accordingly, concluding the autofocus process.

In certain instances, it may be observed that the maximum identified in phase 3 and phase 4 coincides, signifying that no further motor adjustment is necessary.

[Fig4.25] illustrates the conclusion of the third phase, where the final result corresponds to the depiction in [Fig4.25.a]. Nevertheless, precisely gauging the distance to optimal focus remains challenging at this stage. Subsequently, the fourth phase is initiated to refine the previous results within a narrower margin, typically just 10 μm in this case. Notably, the outcome of the 4-step autofocus algorithm is portrayed in [Fig4.25.b], showcasing a remarkable enhancement in central focus compared to [Fig4.25.a].

The apparent blurriness at the image edges can be attributed to the camera lens aperture of the IntraCell, which is an inherent characteristic of the camera. However, this peripheral blurring has no bearing on the autofocus process, as the image is cropped, nor does it impact the Optoporation procedure, as the laser is targeted at the center of the MEA.

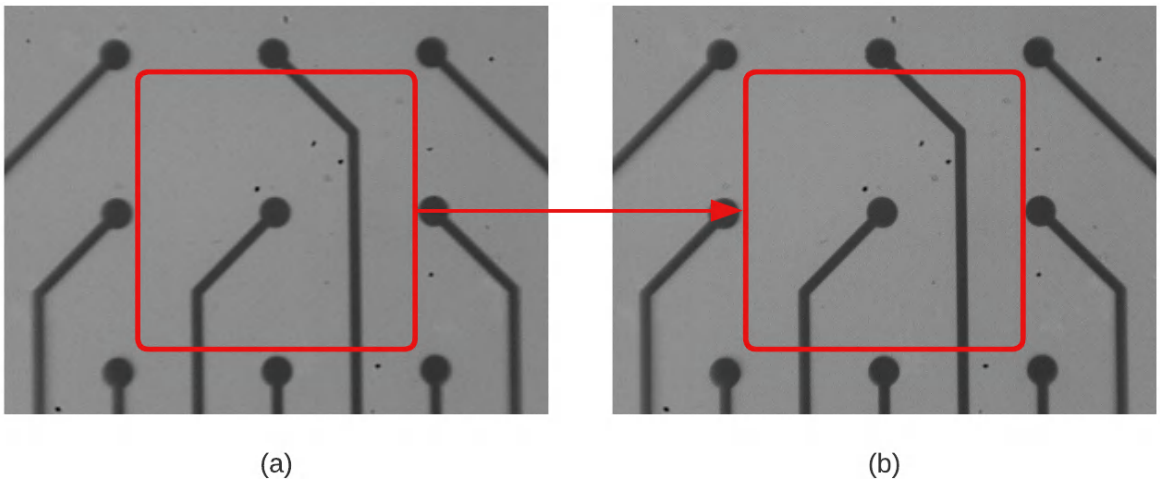


Figure 4.25 Fine focus result: Image (a) corresponds to the output of the third phase, while Image (b) represents the output of the fourth phase. Notably, the central region of Image (b), highlighted with a red frame, exhibits significantly improved focus compared to Image (a). The discernible difference between these two images lies in a 10-micron separation along the Z-axis

4.7 Final result of the full Autofocus procedure

Following the successful implementation of autofocus procedures, a robust algorithm was developed, capable of achieving autofocus over a range exceeding $\pm 300\ \mu m$. The full autofocus procedure encompasses the integration of all the mentioned phases, with the primary objective of achieving focus for the IntraCell camera. The comprehensive autofocus implementation comprises the following phases:

1. The initialization phase as the first step.
2. The second phase involves the use of CNN with Fuzzy control, integrated into the CL controller.
3. The third phase incorporates Preprocessing and the Sobel method, optionally combined with SVR.
4. Finally, the fourth phase is dedicated to fine focusing.

Conducting rigorous Autofocus performance assessment through 45 Iterations across three stages of Real-Time testing. The testing process comprised 45 subsequent iterations, divided into three distinct experiments. During the initial experiment, image parameters were set to default values, and the autofocus was executed over a range of 15 intervals, each situated at varying distances from the focal point. This range spanned from -300 to +300 μm.

In the second experiment, the focus was placed on maximizing image brightness. The same procedure was repeated, evaluating the autofocus performance under these optimized conditions.

For the third experiment, the emphasis shifted to maximum contrast. The experiment was duplicated again to observe the autofocus behavior in this contrasting scenario.

The cumulative results of these tests were graphically represented in [Fig4.26]. The y-axis denoted the starting position of the motor for initiating autofocus in micrometers. The motor was manually positioned at the indicated distance. On the x-axis, time in seconds was depicted, representing the algorithm's autofocus duration within the indicated range.

The colors of each data point on the plot corresponded to the response of the autofocus algorithm. These colors indicated the algorithm's deviation from the desired focus, falling within the range of 0 to 20 μm.

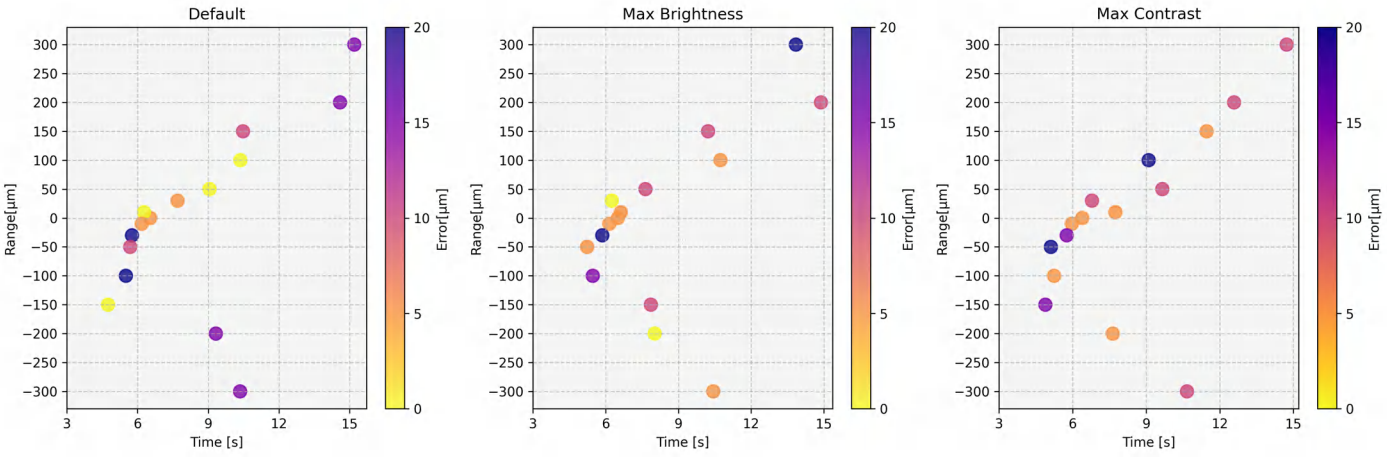


Figure 4.26 The holistic performance of the entire Autofocus process

By encompassing a comprehensive array of scenarios, this testing approach ensured a meticulous evaluation of the autofocus algorithm's performance. Real-world conditions were simulated to facilitate rigorous debugging and optimization.

Furthermore, trials were conducted at even greater distances, spanning up to $\pm 600\ \mu m$, yielding satisfactory outcomes. Nonetheless, it's worth noting that the inclusion of such an extensive range does not pose a risk within the context of this study. What holds greater significance is the autofocus capability within shallower ranges, akin to those scrutinized in the aforementioned tests.

In [Tab4.7] summarizes the comprehensive outcomes of the autofocusing processes. The results obtained underscore the efficacy of the complete autofocus algorithm, boasting an impressive accuracy rate of 85%, coupled with a remarkable resolution of $5\mu m$.

Algorithm	Range	Accuracy	RMSE	ER	Resolution	Execution Time
Integrated Techniques for Autofocus: Fuzzy Control, CNN Classification, and Sobel Operator	$\pm 300 \mu m$	85%	$7.56 \mu m$	15%	$5 \mu m$	$4.5 \sim 16$ sec

Table 4.7 Algorithm evaluation results

These values collectively underscore the algorithm's robustness, rendering it applicable not only to its current context but also suggesting its potential for implementation across diverse machinery to fulfill autofocus requirements.

4.7.1 Video

Watch the approved video [Fig4.27] showcasing the latest autofocus procedure, extensively tested at different ranges. This test utilized IntraCell version 0002 equipped with a dual-core microcontroller and a high-quality monochrome camera. The GUI demonstrated in the video was built by the Foresee Biosystems team. All of these steps were carried out on a 6-well MEA platform.

Figure 4.27 Scan the QR code with your mobile phone to instantly watch the video, or alternatively, you can access it through the following link: https://youtu.be/Sq-RrkoOyj8?si=1mXDsxBrkT-MKrq4

Chapter 5

Conclusion

In summary, my internship at Foresee Biosystems involved a detailed experiment to improve IntraCell's autofocus using various image processing techniques. The research spans a wide spectrum of disciplines including computer vision, machine learning, and closed-loop control strategies. Methods such as Sobel filtering, SVR, CNN, and fuzzy control were explored to refine the autofocus system's performance.

The investigation commences by evaluating the efficacy of the most popular focusing techniques, such as Sobel filtering, Canny edge detection, Fast Fourier Transform, noise processing, and saturation analysis. These methods were rigorously tested across 25 trials, encompassing 125 iterations. Notably, Sobel and FFT methods consistently achieved high accuracy levels of 92% and 88%, respectively, with low root mean square error (RMSE) values. Integration of SVR further enhances precision in performance curves derived from the most successful methods, enabling fine-tuning of focal points. Moreover, the Sobel method demonstrates significantly shorter processing times compared to the FFT method. In practical terms, it's three times faster in extracting Relevant Values from images.

To enhance the accuracy and discrimination of focal points, CNN was introduced, resulting in the categorization of images into meaningful labels such as *"Extremely Far", "Far", "Almost Far", "Close"*, and *"Extremely Close"*. This approach achieved an impressive 83% accuracy across 45 trials on the IntraCell machine.

Furthermore, the incorporation of fuzzy control, utilizing CNN-approved image tags, effectively manages Z-motor movements for autofocus, streamlining motor adjustments.

The culmination of these techniques culminates in a comprehensive autofocus algorithm boasting an impressive 85% accuracy rate, coupled with minimal RMSE values and a remarkable 5µm resolution across a range exceeding $\pm 300\ \mu m$. Rigorous testing underpins

the algorithm's robustness and practicality, positioning it for seamless integration into various robotic systems.

Elevating robotic applications

In the field of robotics, the integration of autofocus emerges as a pivotal advancement, wielding its significance across a diverse array of applications reliant on visual perception and meticulous object recognition. The marriage of robotics and autofocus opens the gateway to a multitude of scenarios where this symbiotic relationship bestows substantial and undeniable advantages upon robotic systems. Herein are a multitude of scenarios where the integration of autofocus confers substantial advantages to robotic systems:

1. Object Inspection and Recognition: The inspection of objects spanning a spectrum of shapes and dimensions stands as a recurring imperative. The challenge of automatically identifying the focal area within images of low contrast holds pivotal importance, especially for camera-equipped surveillance systems. The intricacies intensify due to the reduced signal-to-noise ratio, leading to a pronounced degradation in autofocus performance within such low-contrast images. Consequently, discerning the focal region becomes notably intricate. Addressing this intricacy, they employ a salient object detection methodology to execute autofocus. This approach sheds light on the spotlight cast upon autofocus in the works (Allegro et al., 1996; Mu et al., 2019), both of which deftly harnessed autofocus for their respective reported objectives.

2. Assembly and Manufacturing: Within the complex domain of manufacturing, the task of manipulating components with diverse sizes and configurations emerges as a constant demand. In the manufacturing continuum, image-based focus measurement assumes paramount importance, playing a pivotal role in automated inspection, assembly, and soldering processes. When engaging in tasks such as object inspection or manipulation, the visual system must discern the optimal image position, guaranteeing comprehensive visibility of all product intricacies. It is here that the indispensability of an autofocus system becomes evident. Shedding light on this essential aspect, the work of (Farnes et al., 2021) draws attention, as they adeptly employ autofocus to fulfill the stated objective.

3. Quality Control: In the domain of robotic quality control, detecting defects is paramount. In (Udupa et al., 2000), emphasizes the role of surface topography, especially roughness and form, in engineering part performance. Optics, driven by Opto-mechatronics, takes the lead in measurements. The paper outlines a confocal microscope's principles, integrated with fixtures and software for 3D data. It employs both intensity and auto-focus

techniques to measure 2D surface roughness effectively. This study highlights the fusion of advanced measurement and automation for precise robotic inspections.

4. Pick and Place Operations: In (Tsai et al., 2014), a high-speed pick-and-place methodology for cell-assembly applications is introduced. While factors such as range of motion and precision hold significance, the agility of manipulation systems has been somewhat overlooked in the existing literature. Embarking in the field of high-speed micromanipulation necessitates the swift acquisition of 3-D coordinates for both the intended microobject and the end effector. Addressing this challenge calls for the implementation of autofocus to surmount this obstacle effectively.

5. Navigation and Obstacle Avoidance: Autonomous drones and ground robots rely on visual sensors for navigation. Autofocus helps them adapt swiftly to changing environments, make informed decisions, and avoid obstacles. In (Ghidary et al., 2000), a PTZ camera-powered robotic system identifies humans in a room, localizes them, and guides the robot toward them. The method combines motion, shape, and color analysis to detect human heads and faces, estimating their 3D positions using the depth-from-focus approach. This showcases advancements in human-robot interaction.

6. Surveillance and Security: Video surveillance constitutes an intricate integrated system, boasting formidable preventative capabilities, and finds extensive application in diverse sectors, including the military, customs, law enforcement, firefighting, airports, railways, urban transport, and various other public spaces. This technology stands as a pivotal component within security systems, owing to its ability to offer visualized, precise, timely, and information-rich content. The prominence of video surveillance as a primary tool is attributed to its capacity to provide abundant, intuitive, and precise information. In (Xu et al., 2016), the utilization of algorithms, such as autofocus, is suggested and underscored for enhanced performance.

7. Medical Applications: Continuum robots have gained traction in the industrial and medical fields, driving increased research interest. Conventional designs with bulky base-mounted drive units compromise system mobility. In (Mandolino et al., 2022), the study introduces a novel integrated robotic endoscope featuring lightweight shape memory alloy (SMA) wire actuators within a flexible structure, yielding a compact design. They detail the mechanical setup and present an SMA-driven autofocus camera development for the endoscope. Future work aims to embed autofocus at the robot's tip. This autofocus technology is particularly valuable in medical robotics, ensuring clear visual feedback during surgical procedures and enhancing precision.

At the intersection of these scenarios, the capability of automated focus adjustment emerges as a catalyst. This empowers robots to adeptly navigate changing conditions and objects without human intervention. The resulting improvement in efficiency, accuracy, and overall performance resonates throughout robotic systems, augmenting their effectiveness across diverse applications.

Accomplished objectives

In essence, this represents a significant leap forward in advancing autofocus capabilities within the realm of robotics. The study's multifaceted approach, coupled with the innovative integration of methodologies, underscores its invaluable contributions to this evolving field.

As previously mentioned, the ultimate autofocus algorithm combines all the methods discussed in the state-of-the-art research and implementation of fuzzy control. Consequently, it offers increased robustness and efficiency across various scenarios. In brief, when compared to other reported methods, older enables rapid focusing at both mid-range and deep-range distances. Additionally, it achieves this efficiency without requiring an extensive dataset, as is often the case with machine learning methods employing neural networks. Instead, I employ MobileNet V2, which was designed with a small dataset for memory-constrained devices and still delivers impressive accuracy.

The objectives achieved encompass a comprehensive dataset collection under various imaging conditions, leveraging neural networks for automated focusing, implementation of closed-loop algorithms, optimization of motor movements to mitigate vibrations, efficient thread management for motion time optimization and image acquisition, incorporation of filtering techniques to mitigate parameter-induced effects, and successful implementation of autofocus algorithms on the latest generation of IntraCell. These accomplishments collectively enhance autofocus speed, robustness, and overall efficiency.

Exploring future: advancements & collaborations in autofocus innovation

Looking ahead, a plethora of avenues present themselves for the continuous refinement and expansion of our endeavors:

Cutting-edge Machine Learning Models: An exploration into the realm of advanced machine learning models offers promising potential. Delving into the intricacies of deep neural networks fortified with attention mechanisms, harnessing the power of transformer-based architectures, or even embracing the creativity of Generative Adversarial Networks (GANs) could usher in remarkable strides in both image processing finesse and classification

accuracy. Harmonious Hardware Fusion: The harmony between software and hardware beckons for attention. A harmonious symphony emerges when software engineers and their hardware counterparts collaborate, orchestrating the seamless integration of the autofocus algorithm into the robotic system's physical embodiment. Here, synchronization with motor controls, the orchestration of an ensemble that mitigates mechanical vibrations, and the resolution of any lingering hardware constraints intersect to elevate the algorithm's efficacy.

Synergistic Sensor Synergy: For those scenarios amenable, the integration of diverse sensor data emerges as a path to augmented precision and reliability within the autofocus system. The infusion of depth sensors or infrared sensors, for instance, could illuminate the path toward heightened accuracy, particularly when the optic is very close to focus.

Embarking on the Collaborative Frontier: As horizons expand, contemplation emerges on how autofocus algorithms like the one described in this thesis, the autofocus algorithm, could find their place in the realm of collaborative robotic systems. In such settings, where human-robot interaction thrives, the refined prowess of precise image analysis could potentially act as a linchpin for task accomplishment, igniting new dimensions of cooperation and proficiency.

References

(2007). scikit-learn ,introduction of support vector machines. https://scikit-learn.org/stable/modules/svm.html.

(2017). Teledyne photometric,introduction to electrophysiology. https://www.photometrics.com/learn/electrophysiology/introduction-to-electrophysiology.

Allegro, S., Chanel, C., and Jacot, J. (1996). Autofocus for automated microassembly under a microscope. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 2, pages 677–680. IEEE.

Baek, Y. and Park, Y. (2021). Intensity-based holographic imaging via space-domain kramers–kronig relations. *Nature Photonics*, 15(5):354–360.

Boyat, A. K. and Joshi, B. K. (2015). A review paper: noise models in digital image processing. *arXiv preprint arXiv:1505.03489*.

Burt, P. J. and Adelson, E. H. (1987). The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier.

Buzsáki, G., Anastassiou, C. A., and Koch, C. (2012). The origin of extracellular fields and currents—eeg, ecog, lfp and spikes. *Nature reviews neuroscience*, 13(6):407–420.

Castleman, K. R. (1996). *Digital image processing*. Prentice Hall Press.

Chan, C.-C., Huang, S.-K., and Chen, H. H. (2017). Enhancement of phase detection for autofocus. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 41–45. IEEE.

Davis, A. A., Farrar, M. J., Nishimura, N., Jin, M. M., and Schaffer, C. B. (2013). Optoporation and genetic manipulation of cells using femtosecond laser pulses. *Biophysical journal*, 105(4):862–871.

E Woods, R. and C Gonzalez, R. (2008). Digital image processing.

Farnes, S. A. R., Tsai, D.-M., and Chiu, W.-Y. (2021). Autofocus measurement for electronic components using deep regression. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 11(4):697–707.

Firestone, L., Cook, K., Culp, K., Talsania, N., and Preston Jr, K. (1991). Comparison of autofocus methods for automated microscopy. *Cytometry: The Journal of the International Society for Analytical Cytology*, 12(3):195–206.

Garg, A., Pan, X.-W., and Dung, L.-R. (2022). Licent: Low-light image enhancement using the light channel of hsl. *IEEE Access*, 10:33547–33560.

Gedraite, E. S. and Hadad, M. (2011). Investigation on the effect of a gaussian blur in image filtering and segmentation. In *Proceedings ELMAR-2011*, pages 393–396. IEEE.

Ghidary, S. S., Nakata, Y., Takamori, T., and Hattori, M. (2000). Localization and approaching to the human by mobile home robot. In *Proceedings 9th IEEE International Workshop on Robot and Human Interactive Communication. IEEE RO-MAN 2000 (Cat. No. 00TH8499)*, pages 63–68. IEEE.

Heckenlively, J. R. and Arden, G. B. (2006). *Principles and practice of clinical electrophysiology of vision*. MIT press.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Iachetta, G., Colistra, N., Melle, G., Deleye, L., Tantussi, F., De Angelis, F., and Dipalo, M. (2021). Improving reliability and reducing costs of cardiotoxicity assessments using laser-induced cell poration on microelectrode arrays. *Toxicology and Applied Pharmacology*, 418:115480.

Iachetta, G., Melle, G., Colistra, N., Tantussi, F., De Angelis, F., and Dipalo, M. (2022). Chronic cardiotoxicity assessment by cell optoporation on microelectrode arrays. *bioRxiv*, pages 2022–06.

Inagaki, S. and Nagai, T. (2016). Current progress in genetically encoded voltage indicators for neural activity recording. *Current opinion in chemical biology*, 33:95–100.

Kouritas, V. K., Hatzoglou, C., Foroulis, C. N., and Gourgoulianis, K. I. (2008). Human parietal pleura present electrophysiology variations according to location in pleural cavity. *Interactive CardioVascular and Thoracic Surgery*, 7(4):544–547.

Langehanenberg, P., Kemper, B., Dirksen, D., and Von Bally, G. (2008). Autofocusing in digital holographic phase contrast microscopy on pure phase objects for live cell imaging. *Applied optics*, 47(19):D176–D182.

Li, C., Moatti, A., Zhang, X., Ghashghaei, H. T., and Greenbaum, A. (2021). Deep learning-based autofocus method enhances image quality in light-sheet fluorescence microscopy. *Biomedical Optics Express*, 12(8):5214–5226.

Luo, W., Greenbaum, A., Zhang, Y., and Ozcan, A. (2015). Synthetic aperture-based on-chip microscopy. *Light: Science & Applications*, 4(3):e261–e261.

Luo, Y., Huang, L., Rivenson, Y., and Ozcan, A. (2021). Single-shot autofocusing of microscopy images using deep learning. *ACS Photonics*, 8(2):625–638.

Mandolino, M., Britz, R., Goergen, Y., Rizzello, G., and Motzki, P. (2022). Development of an sma driven articulation and autofocus mechanism for endoscope applications. In *ACTUATOR 2022; International Conference and Exhibition on New Actuator Systems and Applications*, pages 1–4. VDE.

Mechelli, A. and Vieira, S. (2019). *Machine learning: methods and applications to brain disorders*. Academic Press.

Melle, G., Bruno, G., Maccaferri, N., Iachetta, G., Colistra, N., Barbaglia, A., Dipalo, M., and De Angelis, F. (2020). Intracellular recording of human cardiac action potentials on market-available multielectrode array platforms. *Frontiers in Bioengineering and Biotechnology*, 8:66.

Mu, N., Xu, X., and Zhang, X. (2019). Finding autofocus region in low contrast surveillance images using cnn-based saliency algorithm. *Pattern Recognition Letters*, 125:124–132.

Ogata, K. (2010). *Modern control engineering fifth edition*.

Passino, K. M., Yurkovich, S., and Reinfrank, M. (1998). *Fuzzy control*, volume 42. Addison-wesley Reading, MA.

Pinkard, H., Phillips, Z., Babakhani, A., Fletcher, D. A., and Waller, L. (2019). Deep learning for single-shot autofocus microscopy. *Optica*, 6(6):794–797.

Plummer, H., Baird, R., Hammerle, A., Adamczyk, J., et al. (1999). Measurement of automotive catalyst washcoat loading parameters by microscopy techniques. *Microscopy and Microanalysis*, 5(4):267–281.

Price, J. H. and Gough, D. A. (1994). Comparison of phase-contrast and fluorescence digital autofocus for scanning microscopy. *Cytometry: The Journal of the International Society for Analytical Cytology*, 16(4):283–297.

Priyadarshini, D., Ivica, J., Separovic, F., and De Planque, M. R. (2022). Characterisation of cell membrane interaction mechanisms of antimicrobial peptides by electrical bilayer recording. *Biophysical Chemistry*, 281:106721.

Rocha, A. (2022). Cell membrane - cellular biology - adapted for ions gradient and membrane channels.png.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Spira, M. E., Huang, S.-H., Shmoel, N., and Erez, H. (2019). Multisite intracellular recordings by mea. *Advances in Neurobiology*, 22:125–153.

Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C., Carandini, M., and Harris, K. D. (2018). Spontaneous behaviors drive multidimensional, brain-wide neural activity. *BioRxiv*, page 306019.

Tanwar, A., Gandhi, H., Kushwaha, D., and Bhattacharya, J. (2022). A review on micro-electrode array fabrication techniques and their applications. *Materials Today Chemistry*, 26:101153.

Tsai, C.-Y., Wong, C.-C., Yu, C.-J., Liu, C.-C., and Liu, T.-Y. (2014). A hybrid switched reactive-based visual servo control of 5-dof robot manipulators for pick-and-place tasks. *IEEE Systems Journal*, 9(1):119–130.

Udupa, G., Singaperumal, M., Sirohi, R., and Kothiyal, M. (2000). Characterization of surface topography by confocal microscopy: I. principles and the measurement system. *Measurement Science and Technology*, 11(3):305.

Valdecasas, A. G., Marshall, D., Becerra, J. M., and Terrero, J. (2001). On the extended depth of focus algorithms for bright field microscopy. *Micron*, 32(6):559–569.

Wang, L., Weng, L., Xia, M., Liu, J., and Lin, H. (2021). Multi-resolution supervision network with an adaptive weighted loss for desert segmentation. *Remote Sensing*, 13(11):2054.

Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495.

Xu, Z., Hu, C., and Mei, L. (2016). Video structured description technology based intelligence analysis of surveillance videos for public security applications. *Multimedia Tools and Applications*, 75:12155–12172.

Young, I., Zagers, R., Van Vliet, L., Mullikin, J., Boddeke, F., and Netten, H. (1993). Depth-of-focus in microscopy. In *Proceedings of the Scandinavian Conference on Image Analysis*, volume 1, pages 493–493. Citeseer.