# DROPLET MEASUREMENT AND ANALYSIS USING IMAGE PROCESSING FOR INSECTICIDE SPRAY: UNIVERSITY OF FLORIDA

By

RIYA PARAGKUMAR DESAI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2023

To my Mom and Dad for giving me the courage to take up any challenge with a smile.

## ACKNOWLEDGMENTS

Foremost, I would like to acknowledge and give my warmest thanks to my Thesis chair/advisor, Dr. Eisenstadt, who gave me this research opportunity and scholarship. His continuous support and guidance carried me through my thesis study and research.

In addition, I would like to thank my committee members, Dr. McNair and Dr. Wiens for their immense support and insightful comments for the betterment of this dissertation.

Last but not the least, I wish to acknowledge with gratitude, my parents who gave me unconditional support and encouragement to remain patient and pursue my goals.

## TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

DROPLET MEASURMENT AND ANALYSIS USING IMAGE PROCESSING FOR INSECTICIDE SPRAY

By

Riya Paragkumar Desai

May 2023

Chair: William R. Eisenstadt
Major: Electrical and Computer Engineering

Particle droplet size and count of insecticides are essential parameters to determine the amount of insecticide needed to cover an intended target area. Diameter volume spectrum values such as $DV_{0.5}$, $DV_{0.1}$, and $DV_{0.9}$ describe the application an insecticide is suitable for and its susceptibility to spray drift. The current tool i.e., Artium PDI TK1 Droplet Analyzer used to calculate the droplet size, count and diameter volume spectrum parameters in the industry could be time consuming, costly, and tedious. It is an expensive benchtop tool which can only test sample slides indoor, consequently, it does not give instantaneous results. This thesis proposes a user-oriented, Internet of Things (IoT) based cost-effective portable tool to give a real-time droplet size and diameter spectrum measurement using image processing for insecticide spray.

# CHAPTER 1
# INTRODUCTION

## Background

Droplet size and count are primary parameters influencing the efficiency of sprayed insecticides. An optimum droplet size and count confirms that the number of droplets produced, provide sufficient probability of contact and are large enough to hold onto body surface of adult mosquitoes.

Insecticide spray drift is the movement of insecticide droplets through the air at the time of application or soon after, to any site other than the area intended [1]. Large droplet size will result in a coarse spray which will cover less target area, hence attacking a fewer number of mosquito bodies. On the other hand, small droplets have a greater potential to drift as the particles will be airborne for a longer time. Droplet size greater than the optimum value will require multiple sprays due to less area coverage. Hence, calculating the number of droplets and their size is essential to evaluate the amount of spray that reaches out in the testing environment, which in turn determines the spray efficiency.

Understanding these parameters is necessary to prevent potential spray drift. The benefit of the insecticide reduces, if, part of the sprayed liquid does not reach the intended target due to spray drift. Spray drift can also be caused by environmental factor such as wind. If the wind velocity exceeds 10 mph, it is recommended to stop spraying [2].

In order to determine which insecticide is best to prevent potential drift from predetermined target and the amount of insecticide required for a particular target area, the researcher needs to calculate the diameter volume spectrum of an insecticide, through insecticide testing. This diameter spectrum also determines the type of application a particular insecticide is suitable for.

Moreover, researchers who work on spraying technologies, sprayer sellers and companies who produce insecticides want to compare spraying equipment and application parameters, rely on diameter volume spectrum calculation as well [3]. Therefore, determining the characteristics and evaluation of droplet distribution is very important.

This chapter reports information about the Anastasia Mosquito Control District (AMCD), which carries out insecticide testing. Further, it defines droplet diameter volume spectrum and its need in insecticide testing. Later, a section detailing the current technique to determine diameter spectrum followed at the AMCD is described.

## Anathasia Mosquito Control District

The professionals at the Anastasia Mosquito Control District (AMCD) are experts in providing environmental-friendly mosquito control services, community education, and scientific research on mosquito-borne diseases for St. Johns County, St. Augustine, FL [4]. In addition, they have a dedicated field for the testing of insecticides. This testing environment follows the guidelines set by the Environmental protection Agency (EPA). Any kind of insecticide can be tested at the AMCD. A few commercial products they have tested include Fyfanon EW, Aqualuer 20-20 and Naled. The AMCD tests insecticides for diameter volume spectrum.

## Diameter Volume Spectrum

Droplet size is measured in micrometer (μm). As discussed in the previous section,  smaller droplets are prone to drift as they fall more slower than larger droplets. Droplet sizes that are smaller than 150 microns have a tedency to drift readily [5].

The spray cloud produced from a given spray nozzle usually consists of a wide variety of droplet sizes. In research, the diameter of droplets has been used for dropletanalysis. Hence, a parameter is required to find the average droplet diameter for the sprayed insecticide. Diameter Volume Spectrum is one way to describe the droplet diameter range. This spectrum calculations include Volume Median Diameter (VMD)/$DV_{0.5}$, $DV_{0.1}$, and $DV_{0.9}$.

Volume median diameter is a diameter of droplets represented in microns. In a volume spray, half of the volume is contained in droplets of diameter that are smaller than the Volume Median Diameter (VMD), and the other half is contained in droplets of diameter that are bigger than the median. A VMD of 50 microns, for instance, reveals that half of the spray volume is in droplets of size smaller than 50 microns in diameter and the other half is in droplets larger than 50 microns in diameter. A visual representation of volume median diameter is as shown Figure 1-1. The image shows that 50% of the droplets are of diameters lesser than VMD and 50% of the droplets are of diameters more than VMD.

While VMD determines the diameter uniformity of the sprayed insecticide on the field, other diameter spectrum parameters like $DV_{0.1}$ and $DV_{0.9}$ can be used to approximate the amount of drift. $DV_{0.1}$ is a diameter of droplets represented in microns. 10% of the spray's volume is made up of droplet diameters that are smaller than the $DV_{0.1}$ and may include the majority of the fine driftable particles. For instance, if a nozzle had a $DV_{0.1}$ of 50 microns, 10% of the volume of spray it produced would consist droplets with diameters of 50 microns in size or smaller.

$DV_{0.9}$ is a diameter of droplets represented in microns. A value of $DV_{0.9}$ means that 90% of the volume of the spray is contained in droplets of diameter that are 10% smaller (or larger) than this number. If the $DV_{0.9}$ is big (for instance, 100 microns), a significant portion of the spray volume may be occupied by a few droplets with big diameters. Then, there could not be enough droplets to adequately provide spray coverage [5].

The likelihood of drift decreases with increasing $DV_{0.1}$ value. The higher the $DV_{0.9}$ value, fewer droplets will be produced, which will not give sufficient coverage.

The VMD value essentially determines the application, a particular pesticide is suitable for. For example, VMD less than 145 microns is effective for flying insects as it remains in the air stream, VMD between 145-325 microns works for stems and narrow vertical leaves such as grass, and VMD greater than 325 microns is used for flat surfaces. The Table 1-1 displays two different pesticides, fungicides and insecticides. Both these chemical have a range of VMD, each range of VMD depicts a chemical for different applications [6].

**Relative Span**

Relative span (RS) can be mathematically calculated by subtracting the $DV_{0.1}$ value from the $DV_{0.9}$ value and dividing it by the $DV_{0.5}$ as shown in equation 1-1. The smaller this value, the lesser the variation between the diameters of the droplets in the spray spectrum. If two sprays have the same $DV_{0.5}$, for example 50 microns but one has $DV_{0.1}$ of 20 and $DV_{0.9}$ of 100 and the other has a $DV_{0.1}$ of 10 and $DV_{0.9}$ of 80, their relative span will be 1.6 and 1.4 respectively. Based on the RS of the latter spray, it can be inferred that it has a lesser variation among drop diameters, will drift less and have more coverage [5].

$$\text{Relative Span} = (DV_{0.9} - DV_{0.1})/VMD \tag{1-1}$$

**Current Method Used at AMCD to Determine DV**

The AMCD has a field for testing insecticides. The field at the AMCD is 125*122m. The physical diagram of the AMCD mosquito insectide test field is shown in Figure 1-2. This field consists of 9 mosquito trial poles placed equidistantly at 31m apart. It consists of a road on every side of the field. Figure 1-3, shows one of the mosquito poles from the field. This mosquito pole consists of an opening which can fit a spinning impinger as shown in Figure 1-4 B) also known as a spinner. A separate image of the spinning impinger is shown in Figure 1-4 A). Spinning Impinger is devised to collect the airborne particles from a spray cloud at a fixed location. It can be utilized for products like herbicide, insecticide and fungicide testing. It can be switched on/off remotely. It also has the facility to be detected in the dark through its in-built light. It consists of two mount brackets which are compatible with 1 inch slides and 3mm rods [7]. More information about the application of different slides and rods for insecticide testing is explained later in this section. These slides/rods are attached in a vertical orientation on each end of the rotating arm of the spinner.

Once the spinner is fixed on the pole, the brackets on the spinner maybe mounted with rods or slides based on the type of insecticide to be tested. If the insecticide tested is water-based then either water-sensitive cards or Magnesium oxide slides are mounted on the spinners.

Magnesuim Oxide (MgO) slides used at the AMCD are as shown in Figure 1-5. Magnesium Oxide is smoked on a glass slide. By viewing the slide against light, the uniformity of the deposit can be checked. Only the centre part of the slide is coated with MgO, to make it easy to handle these slides. Ideally, MgO slides are used to test droplets of diameter 20-200 μm. These slides will not be very useful to measure large droplets as the droplets would bounce off the surface. Mostly alternative slides are preferred over, MgO slides as they can be easily damaged during the testing process [8].

The water sensitive cards used at the AMCD are as shown in Figure 1-6. This is the Spot-On spray pattern test paper. It is an essential tool to visualize sprayed insecticide. It is highly sensitive to water and whichever part comes in contact with the spray simply turns blue. Along with droplet size these papers help realize the uniformity of the applied insecticide on the field. This paper also helps verify if the insecticide sprayed has covered the tester's intended target [9].

This Spot-On paper can be used to measure droplets of any size. Tests show spray coverage from fine to extremely coarse for different tested insecticides in Figure 1-7 [10].

If the insecticide is oil-based then Teflon-coated rods/slides as shown in Figure 1-8 are used. Teflon coated rods are not reusable. Teflon rods are preferred when the insecticide to be tested is fine and in case the insecticide to be tested is coarse then Teflon slides are used. The Teflon coat is advantegous mainly because it prevents the spray from flowing off the slide/rods.

Once the slides are setup on these spinners, the spinners are switched on so they continuously spin while the insecticide is sprayed. This allows uniform deposition of spray on the mounted slides/rods. The insecticide is sprayed using a spray truck from either road shown in the Figure 1-2. This spray truck consists of a Ultra Low Volume (ULV) spray fogging machine. It is ADAPCO Guardian 190ES ULV sprayer which is equipped with an air-shear nozzle by Adapco. ULV machines distribute very fine aerosol droplets of size ranging from 1–150 μm evenly on large fields. ULV applications require small quantities of active ingredient in relation to the size of the area treated, as the insecticides sprayed are not diluted. This improves killing efficiency. Typically less than 3 ounces per acre [11] is sprayed, which minimizes exposure and risks to people and the environment. Other advantages of ULV spraying include lower risks of injury due to the fog cloud being nearly invisible, low volumes of carrier chemicals, lower application cost and low noise levels. These slides are then taken of the spinner for droplet analysis.

The testing occurs in three rounds with a time interval of 15 minutes between each round. In every round the same insecticide is sprayed onto the field using the ULV fogging machine. Every round the mounted slides/rods on the spinners are replaced. All the samples are then brought indoors and tested for the diameter spectrum using droplet analysis at the AMCD. The analysis at the AMCD is done using Artium PDI TK1 Droplet Spray Analyzer computer as shown in Figure 1-9. This analyzer uses Phase Doppler Interferometry (PDI) for measuring size and velocity of spherical droplets.

The Phase Doppler interferometer instrument otherwise known as the Phase Doppler Particle Analyzer (PDPA), uses standard laser-based diagnostic instrument which evaluates size and velocity of individual spherical particles. It works on the principle of light scattering interferometry. This principle utilizes wavelength of light as a measurement scale. This method does not require frequent calibration, just the factory calibration is enough. The factors which affect the diameter and velocity measurement include laser wavelength, beam intersection angle, transmitter and receiver focal lengths and detector separation. This parameters are not affected with the aging of the instrument [12]. Each sample obtained at the AMCD is placed in the microscope as shown in the Figure 1-10. Each sample tested generates an output PDF document depicting the parameters obtained during droplet analysis at the AMCD. The analysis includes VMD, $DV_{0.1}$, $DV_{0.9}$ and Relative Span as shown in Figure 1-10. This data analysis result also includes a histogram distribution of Droplet count vs Diameter of individual particles tested every sample slide, as shown in Figure 1-11 . This graph also highlights the VMD of the droplet distribution of each sample slide. The probability density function curve shows the probability that a particular droplet diameter lies in given probability range.

## Problem Statement

While the Artium TK1 PDI Droplet Analyzer provides efficient results, this tool has a few limitations as well. This is a bench top tool which can only be used indoors. In general, at the end of every three round insecticide testing, the slides are collected and taken indoors for analysis using this tool. This device cannot give instant or real-time results as soon as the field is sprayed with insecticide. Around 60 slides need to be tested after each insecticide testing, which takes atleast 3-4 hours according to the experts at the AMCD. Collecting the slides from the field and testing indoors requires heavy labour. Even though the tool might produce accurate results but the process could be made more proficient.

## Conclusion

This thesis presents a solution to the above defined problem. This problem is resolved by the development of a portable, real-time droplet imager. The technical details of this tool is explained in the next chapter. This thesis also highlights a prototype on how this tool can be implemented in the AMCD field.

Table 1-1. Different application of Fungicide and Insecticide based on their VMD range [6].

| Application | Droplet Category | Approximate VMD Range (in microns ) |
|---|---|---|
| Fungicide | | |
| foliar protective or curative | Medium(M) | 226-325 |
| Insecticide | | |
| foliar contact or stomach poison | Medium(M) | 226-325 |
| foliar systemic | Coarse(C) | 326-400 |
| soil-applied systemic | Coarse(C) | 326-400 |
| | Very Coarse (VC) | 401-500 |
| | Extremely Coarse (XC) | 500-650 |
| | Ultra Coarse (UC) | >650 |
| Insecticide used at the AMCD | Very Fine (VF) | 2-25 |

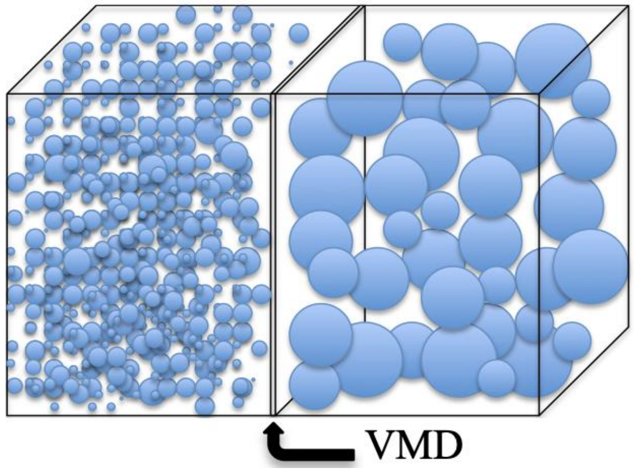



Figure 1-1. Visual representation of Volume Median Diameter

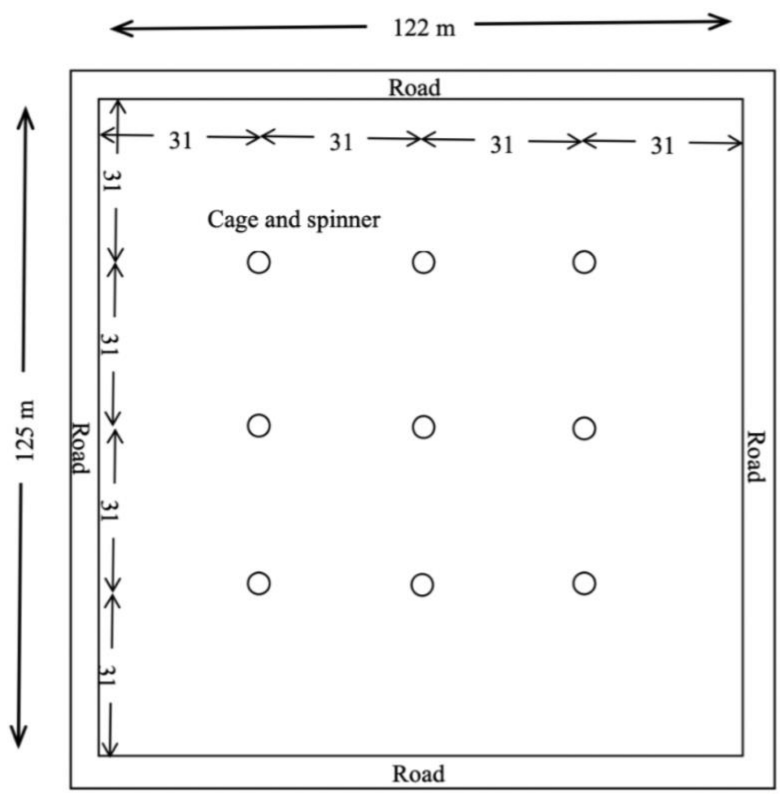



Figure 1-2. The physical diagram of the AMCD mosquito insectide test field.

Figure 1-3. Mosquito pole placed in the insecticide testing field at the AMCD

A 

B

Figure 1-4. Spinning Impinger A) Spinning Impinger used for insecticide testing at AMCD B) Spinning Impinger placed on the mosquito pole in the field at the AMCD

Figur 1-5. Magnesium Oxide coated slide used in the AMCD

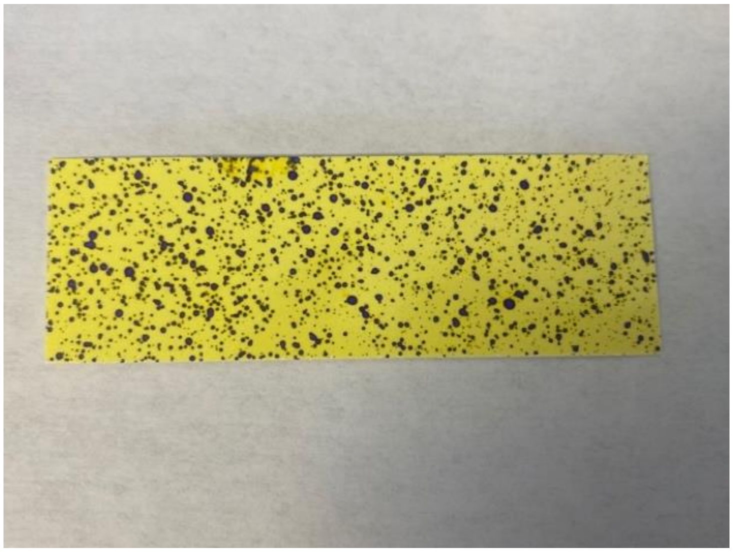


Figure 1-6. Sprayed water-sensitive card sample from the AMCD

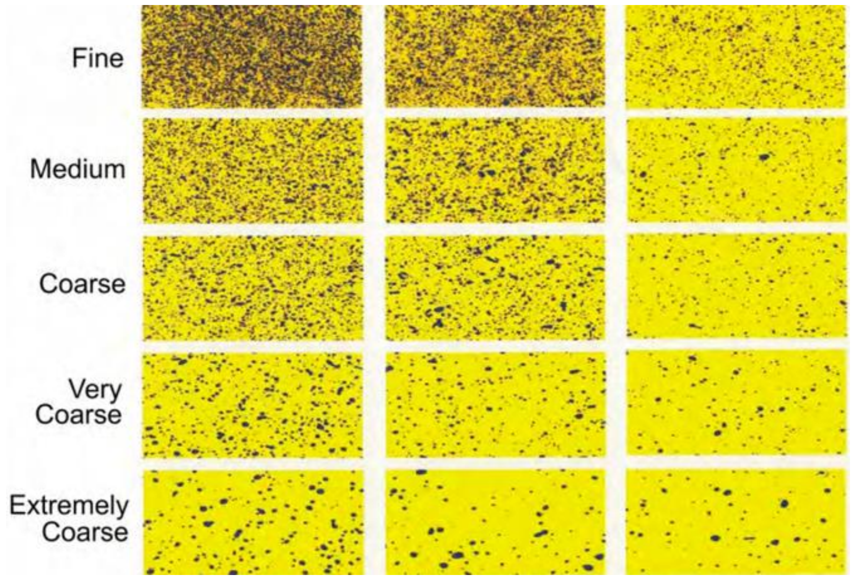



Figure 1-7. Visible spray coverage from fine to extremely coarse on water-sensitive paper [10].

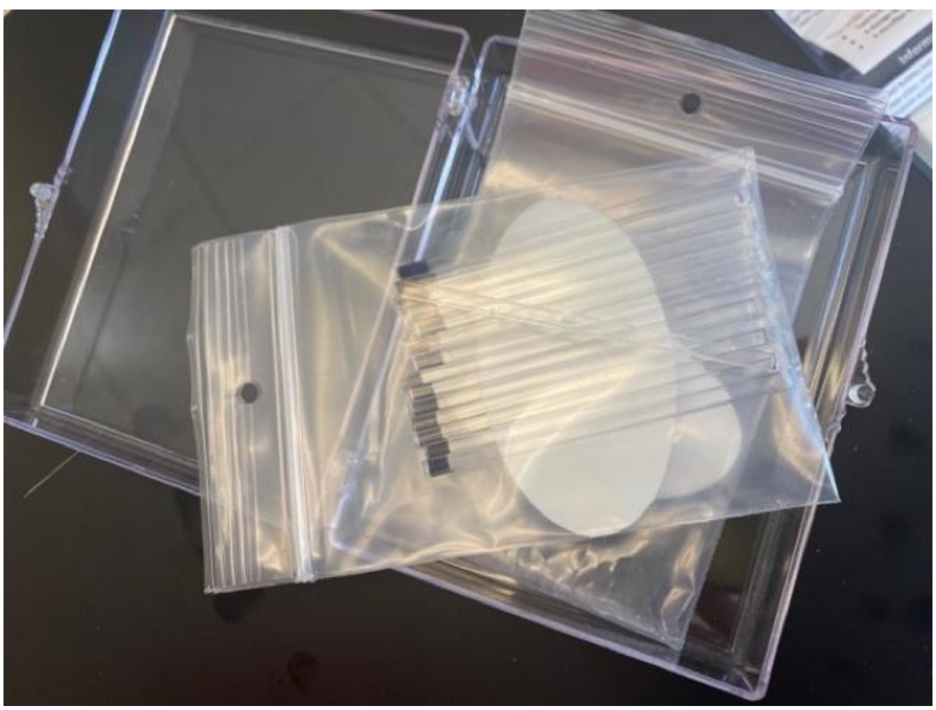


Figure 1-8. Teflon rods

Figure 1-9. Artium PDI TK1 particle analyzer used at the AMCD

# $Dv_{0.5}$ (VMD): 12.25 microns

$Dv_{0.1}$ : 9.17 microns

$Dv_{0.9}$ : 18.39 microns

Relative Span: 0.75

Total Droplets Collected: 181
Droplets over 32 microns: 0
Droplets over 48 microns: 0
Area: 32.667  Drops / mm squared: 5.541

Figure 1-10. VMD, DV0.1, DV0.9, Droplet Count calculated by the  Artium PDI TK1 particle analyzer used at the AMCD.

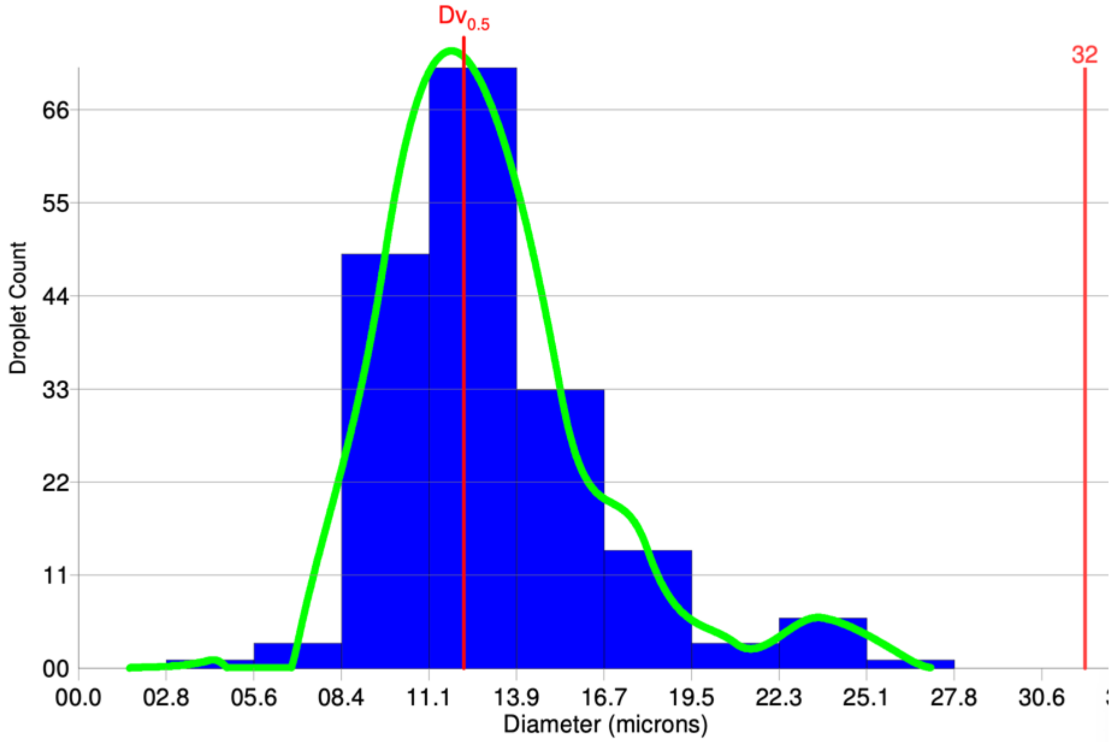



Figure 1-11. Histogram/Probability distribution curve of the number of droplets for the diameter range obtained by Artium PDI TK1 particle analyzer used at the AMCD.

# CHAPTER 2
# DROPLET IMAGER

## Image Processing

An image can be defined as a two-dimensional function, F(x,y), where x and y are spatial coordiates and the value of the function F, at any pair of coordinates (x,y), is called the intensity of that image at that point [13].

An image can otherwise be defined as a two-dimensional array specifically arranged in rows and columns.

Grayscale images are single-channeled images represented by a pixel matrix. Each pixel is is one element of the matrix, the value of the pixel can range from 0-255 depending on the intensity of the gray shade. The values in pixel presentation are uniformly vary from zero (black pixels) to 255 (white pixels). Figure 2-1 shows the a grayscale image and its matrix representation [14].

These images are exclusively made up of shades of gray. Grayscale images should not be confused with binary images. Binary images contain only black and white pixels. In binary images, either a pixel is black (pixel value ‘0’) or it is white (pixel value ‘1’). They have no colours in between. But Grayscale images have a wide range of shades of grey in their pixels.

A BGR image consists of the three channels, blue, green and red. Each of these channels consist of pixels values ranging from 0-255, defining the range of intensity of each of these colours. These are primary colours and when these are mixed secondary colours are formed on an image. In this research the input image channels have been split and only red channel has been selected for thresholding. The red channel obtained is also a grayscale image because it only consists of pixels from 0-255.

## Hardware

This introduction hardware solution for droplet counter consists of three main hardware components, Raspberry Pi (RPI) 4B+, HQ camera, Pimoroni Microscope lens.

### Raspberry Pi 4B+

This model of Raspberry pi (RPI) as shown in Figure 2-2 [15], has Broadcom BCM2711, Quadcore Cortex-A72 (ARM v8) 64-bit SoC @1.5GHz. It runs on 8GB SDRAM (Synchronous Dynamic Random Access Memory). It consists of 2-Micro HDMI ports. It also has a Micro-SD card slot for loading operating system and data storage.

To take images the RPI has a 2-data lane MIPI CSI camera connector. Camera Seial Interface (CSI) is an application of the Mobile Industry Processor Interface (MIPI). A 15-pin ribbon cable interface, connects the camera, to the host processor . In this system, HQ Camera is connected to Raspberry Pi 4B+ using this interface. This interface can be used to connect any Raspberry Pi cameras available in the market like picamera, and picamera V2. The RPI consists of 2, 2.0 USB ports and 2, 3.0 USB ports to connect accessories like Keyboard, and Mouse. The device has a USB-C power connecter for 5V DC supply. This supply can be provided using a power bank with minimum 2.5 A for remote operations.

Raspberry Pi 4B+ was an appropriate microcontroller for this system because it works on 2.4 GHz and 5 GHz IEEE 802.11ac wireless [16]. It can connect to VNC viewer and be controlled remotely. The quick Imager installation provides an operating system that runs essential tools like Visual Studio Code which can program the RPI to run required algorithms. The RPI has a camera port which can serially connect a camera and function with the help of the RPI command line or a python library called

PiCamera. Plenty of libraries can be installed on the RPI for data visualization and analysis, for example, numpy, mathplotllib, scimage.feature, scimage.segmentation, csv. Tools like Git can be installed on the RPI.This allows the work done to be shared, edited and accessed online.

**High Quality (HQ) Camera**

The Raspberry Pi High Quality Camera is the newest camera accessory available for the Raspberry Pi. It offers 12.3 megapixels (MP) resolution which is higher than 8MP [17] offered by picamera V2. HQ camera was chosen for this solution due to its high resolution and its increase in sensitivity by 50% for low-light performance. The hardware set up proposed is to be implemented in the field for industrial purposes, hence the improved sensitivity of the camera produces higher quality images even in low light conditions.

The camera package consists of a circuit board carrying a Sony IMX477 sensor, along with an Flexible Printed Circuit (FPC)  cable as shown in Figure 2-3 for connection to the Raspberry Pi, a milled aluminum lens mount with integrated tripod mount and focus adjustment ring, and a C/CS-mount lens adapter.

It attaches to the RPI using a small socket on the board's upper surface and uses the dedicated CSI port, designed especially for interfacing to cameras.

The camera can be connected to both C and CS mount lenses. Due to this feature, third party  lenses with other form factors can be mounted on this camera module as well. This is a plus, since the application of this solution requires particle detection at a micrometer scale, a third party lens with a greater zoom factor can be mounted instead of only using the lenses made specifically for RPI.

**Pimoroni Lens**

This CAM010 lens is compatible with the C and CS mount on the HQ camera. With a body length of 107 mm and diameter of 39 mm the lens has a adjustable zoom factor of 0.12x to 1.8x [19], focus can be altered by changing the distance to the subject. It comes with two lens cap for the top and base. The Pimoroni lens is as shown in Figure 2-4.

The Raspberry pi HQ camera has a few essential lenses, 6mm wide angle lens, and 16mm telephoto lens. The 6mm lens shown in Figure 2-5, is a wide angle lens and only used for landscape photography. The 16mm lens shown in Figure 2-6 has a relatively wide angle of view [20] compared to the Pimoroni lens with no zoom factor. It cannot magnify to visually capture particles on the water-sensitive cards.

Another lens which was tried during testing was the 300x microscope lens shown in Figure 2-7. This lens is also compatible with Raspberry Pi HQ camera with C mount. However, this lens had a very high zoom factor and was not able to capture an optimum area of the slide for testing purpose.

Hence with appropriate miminum focus distance and zoom factor, the Pimoroni lens proved to be efficient for droplet analysis.

**Hardware Diagram**

Figure 2-8 shows the hardware setup of the proposed solution. The RPI is connected to HQ camera, which in turn is connected to the Pimoroni lens. This setup takes an image of the water-sensitive card, from a distance of 6.5 inches. It can capture a 1*1 inch of the image. These calculations were made by mounting the camera and lens setup onto a microscope stand as shown in Figure 2-10. The physical size of the image captured by the lens proved to be an important input parameter to develop the algorithm. This setup runs an image processing algorithm programmed on to the RPI. This hardware setup can be controlled and monitored remotely setup using VNC Viewer. This arrangement can take multiple images, store them, calculate particle count, diameter, diameter spectrum and relative span for individual slides instantly with just a click.

## Software

The software solution is built with python using OpenCV and PiCamera libraries.

### Picamera

Initial camera setup and testing was done with the help of the RPI command line. A prolonged preview was required to setup camera focus and distance to the sample slide, for this purpose ‘raspistill -t 0’ command was used, which gave a continuous preview of the HQ camera and Pimoroni lens setup.

Once the camera was setup, an algorithm was developed to pre-process the sample water-sensitive card slides and calculate the drople count and diameter. The algorithm was developed with the help of a few test images obtained from the AMCD. One of the images is shown in Figure 2-9. These images were captured by the Atrium PDI TK1 Analyzer used at the AMCD. The different stages of developing the algorithm will be explained later in this chapter. Once the algorithm was built, it had to be tested on images taken from the hardware setup, to check if this system could function in real-time.

PiCamera is a python library. This library can be used to control the functions of the HQ camera through the CSI. The camera can simply be activated with initializing PiCamera(). Three basic functions would allow capturing images and saving them to the RPI.

The algorithm simply starts an image preview using camera.start_preview(). A delay is added in the preview so the user can adjust the sample slide for testing. This delay can be altered for hardware setup in the AMCD field and might not required once calibrated to work on the field. The camera.capture() simply captures the image and save to the path provided. The preview is stopped using camera.stop_preview() [21].

In order to get an idea of how this setup can be done in the AMCD field, a microscope stand was added to the setup as shown in Figure 2-10. Experiments with the stand allowed measuring the distance required between the lens and the sample slide in the field for appropriate focus. Another parameter calibrated using this setup was the physical size of the slide being captured while running PiCamera commands. These parameters calculated were explained in the previous section.

**Thresholding**

The most challenging part of developing this algorithm was preprocessing the image. Pre-processing an image is essential to remove noise and a clear image gives efficient analytical results. The best way to remove noise from an image is to perform thresholding. Thresholding is a type of image segmentation used for separating foreground i.e., the required object of the image from the background i.e., the unwanted part of the image. In this research the foreground is the particles to be detected.

During thresholding, pixels values are compared in relation to the threshold value given as input to the function. If the pixel value is lower than the threshold value then then the pixel value is set to ‘0’, otherwise, it is set to a maximum value. The maximum value generally is set to 255. The logic of thresholding is as shown in equation 2-1

$$\begin{array}{l} \text{If } f(x, y) < T \\ \quad \text{then } f(x, y) = 0 \\ \text{else} \end{array} \tag{2-1}$$

$f(x, y) = 255$

where
$f(x, y)$ = Coordinate Pixel Value
T = Threshold Value [18].

The initial approach for thresholding the image was to apply a thresholding function on a grayscale. A similar threshold output is obtained when a threshold function is applied to individual image channels also.

The image shown in Figure 2-9 is a BGR image and the background is yellow, the colour yellow comes from a combination of red and green. Hence in this algorithm, the red channel is chosen for thresholding. The OpenCV function for thresholding used in this algorithm is as shown in equation 2-2. Here the input image is the red channel in the image, threshold value T=155, maximum value is 255, the image output obtained will be an inverted binary, where the foreground will be white and the background will be black.

```
Img_channel= img[ :,:, 2]
ret,binary=cv2.threshold (img_channel, threshold, 255,                    (2-2)
cv2.THRESH_BINARY_INV)
```

The threshold value of 155 was selected using trial and error. A few other threshold values were tested on the red channel as shown in Figure 2-11,12,13 where sample slides were preprocessed with threshold values 100,120,140 respectively.

For this particular application, whether the image is converted to grayscale or channels are split before threshold value is applied, the output binary image remains the same, the parameter which alters the thresholded output image matters, is the threshold value ‘T’ used.

Figure 2-14, shows the different stages of image analysis done on sample images obtained from the AMCD. A simple threshold function was sufficient to produce a clean image for bigger droplet size analysis (VMD value to be added).

**Connected Component Analysis**

The most common way to detect the number of particles and contour them is using the cv2.connectComponentsWithStats as shown in equation 2-3.

```
output = cv2.connectedComponentsWithStats(binary, 4, cv2.CV_32S)
(numLabels, labels, stats, centroids) = output
for i in range(1, len(stats)-1):
x = stats[i, cv2.CC_STAT_LEFT]                                   (2-3)
y = stats[i, cv2.CC_STAT_TOP]
w = stats[i, cv2.CC_STAT_WIDTH]
h = stats[i, cv2.CC_STAT_HEIGHT]
```

This function takes the input thresholded binary image shown in Figure 2-14. Stats is an output of this function which consists coordinates of the bounding box of the particles detected, which is basically the box coordinates of the detected particles. The bounding box values are extracted from stats as shown in equation 2-3 and is used to contour the detected particles. Image contouring is process of identifying structural outlines of objects in an image which in turn can help us identify shape of the object [18]. Another parameter obtained in the output is the centroid/center coordinates of each particle detected on the image. This is used to mark the center of each detected particle as shown in equation 2-4.

```
for c in centroids:
x = int(c[0])                                                     (2-4)
y = int(c[1])
cv2.circle(img, (x,y), radius=1, color=(0, 0, 255), thickness=-1)
```

To show clear particle detection cv2.rectangle() has been used to mark a rectangle around the particles detected with the help of input values obtained from ‘stats‘ on the sample slide image as shown in equation 2-5. The contoured output is as shown in Figure 2-13

```
cv2.rectangle(img, (x,y), (x+w, y+h), color=(0,0,255), thickness=1)
```
(2-5)

**Diameter and Area Calculations**

The physical_width and physical_height of each particle in μm is calculated as shown in equation 2-6

```
physical_width = w / pixels_per_metric_width
physical_height = h / pixels_per_metric_height
radius =  getParticleRadius(physical_width, physical_height)
area = getParticleArea(radius)
```
(2-6)

The physical_width and physical_height are used to calculate the radius of each particle. The area is calculated using the calculated radius. The functions to calculate radius and area created are as shown in equation 2-7.

```
def getParticleArea(radius):
   return np.pi * radius**2

def getParticleRadius(width, height):
   if width == 0:
      return height/2
   elif height == 0:
      return width/2
   else:
      return (height+width)/4
```
(2-7)

During real-time image testing, where images were taken using the RPI setup, the water-senstive cards were sprayed with water. The particle size of water is relatively smaller.Hence thresholding algorithm was tweaked to accommodate for any error due to smaller particle size. The issue with smaller particles is that they overlap making it difficult to segment during basic thresholding, reducing the overall particle count.

Hence, along with the simple thresholding shown in equation 2-2, a few more steps are added to the algorithm for efficient droplet count measurement .

Euclidean distance transform was applied to the find the distance between each pixel in the foreground to its nearest background pixel. Peak_local_max identifies the maximum values from this matrix D obtained from the distance transform and returns an output boolean image. The maximum values output obtained have atleast 5 pixels of a distance between them. This minimum distance value has been set by trial and error. It can be changed based of the spray used for testing. This minimum distance avoids oversegmentation of the image. This value has been set so that particles overlapping each other can be identified as separate particles. Increasing the minimum distance would consider overlapping particles to be one.

```
D = ndimage.distance_transform_edt(thresh)
localMax = peak_local_max(D, indices=False,
min_distance=min_distance,labels=thresh)
markers = ndimage.label(localMax, structure=np.ones((3, 3)))[0]
labels = watershed(-D, markers, mask=thresh)
print("[INFO] {} unique segments found".format(len(np.unique(labels)) - 1))
```

(2-8)

ndimage.label [19] considers Boolean value 1 as features and creates markers for the same as potrayed in equation 2-8. The watershed algorithm [22] [23] [24] considers the local minima values (-D), which are also the background values and makes labels for contouring. The watershed algorithm uses the binary image output as an input mask so that the pixel values that are 0 (black) remain 0.  The label region is allocated memory and drawn on a mask with pixel value 255 (white) as shown in equation 2-9.

```
for label in np.unique(labels):
    # if the label is zero, we are examining the 'background'
    # so simply ignore it
    if label == 0:
        continue                                                    (2-9)
    # otherwise, allocate memory for the label region and draw
    # it on the mask
    mask = np.zeros(thresh.shape, dtype="uint8")
    mask[labels == label] = 255
```

The identified labels are contoured using the cv2.contour function [25]. This also identifies the droplet count of each image.

```
cnts = cv2.findContours(mask.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)                              (2-10)
    c = max(cnts, key=cv2.contourArea)
```

The enclosing method for smaller particles used is circle as shown in equation 2-11.

```
if enclosing_method == "circle":
        # get and draw circle enclosing the object
        ((x, y), radius) = cv2.minEnclosingCircle(c)
        cv2.circle(image, (int(x), int(y)), int(radius), (0, 0, 255), 1)     (2-11)
        physical_width = 2*radius / pixels_per_metric_width
        physical_height = 2*radius / pixels_per_metric_height
        radius = (physical_width + physical_height) / 4
```

The background mentions about the importance of the Diameter Volume Spectrum values. Their calculation is shown in equation 2-12. Here dv01, dv05,dv9 represent $DV_{0.1}$, $DV_{0.5}$, $DV_{0.9}$ respectively.

```
def get_vmd(radiuses):
   diameters = list(map(lambda n: n+n, radiuses))
   n_particles = len(diameters)                                     (2-12)
   sorted_diameters = np.sort(diameters)
```

```
dv01 = sorted_diameters[int(0.1*n_particles)]
dv05 = sorted_diameters[int(0.5*n_particles)]
dv09 = sorted_diameters[int(0.9*n_particles)]
print("DV01: ", dv01)
print("DV05: ", dv05)
print("DV09: ", dv09)
```

The final parameter to be calculated for droplet analysis is the Relative span as shown in equation 2-13.

relative_span = (dv09-dv01)/dv05 (2-13)

**Scaling**

Thresholding can separate forground from background, clean up the image, and aid contouring for particle detection. However, all parameters evaluated are still in pixels. As droplet analysis is best done in microns another essential step for this research is scaling. Scaling is an essential step to detect the particles and calculating particle size from an image. The crucial information required before calibrating the algorithm for any image is its actual/physical image size and resolution.

The algorithm used to calculate particle size was developed using a few test AMCD images. One of the image is as shown in Figure 2.9, it is a water-sensitive card sprayed with insecticides in the AMCD field. The obtained images had a physical image size of 3*1 inch with a resolution of 1252*447 pixels.

The algorithm allows the user to input the physical image size. The physical image size is generally measured in inches, however, output particle size is required to be in $\mu$m. The algorithm developed also consists of a function created to convert the physical image size from inches to micrometers as shown in equation 2-14. One inch is 25400 $\mu$m.

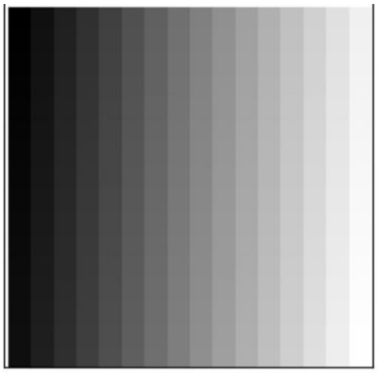
```
def inch_to_mm(x):
        return x*25400
```
(2-14)

The prameters physical_height_of_view and physical_width_of_view are the height and width of the image respectively obtained from converting their values from inches to μm. These are obtained using equation 2-14.

Image shape in pixels was determined using shape() function in OpenCV as shown in equation 2-15, so this value need not be manually entered into the algorithm for every testing.

The number of pixels per metric (μm) were mathematically derived by

```
pixels_per_metric_height=image.shape[0]/physical_height_of_view
pixels_per_metric_width=image.shape[1]/physical_width_of_view
```
(2-15)



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 32 | 48 | 64 | 80 | 96 | 112 | 128 | 144 | 160 | 176 | 192 | 208 | 224 | 240 |
| 1 | 17 | 33 | 49 | 65 | 81 | 97 | 113 | 129 | 145 | 161 | 177 | 193 | 209 | 225 | 241 |
| 2 | 18 | 34 | 50 | 66 | 82 | 98 | 114 | 130 | 146 | 162 | 178 | 194 | 210 | 226 | 242 |
| 3 | 19 | 35 | 51 | 67 | 83 | 99 | 115 | 131 | 147 | 163 | 179 | 195 | 211 | 227 | 243 |
| 4 | 20 | 36 | 52 | 68 | 84 | 100 | 116 | 132 | 148 | 164 | 180 | 196 | 212 | 228 | 244 |
| 5 | 21 | 37 | 53 | 69 | 85 | 101 | 117 | 133 | 149 | 165 | 181 | 197 | 213 | 229 | 245 |
| 6 | 22 | 38 | 54 | 70 | 86 | 102 | 118 | 134 | 150 | 166 | 182 | 198 | 214 | 230 | 246 |
| 7 | 23 | 39 | 55 | 71 | 87 | 103 | 119 | 135 | 151 | 167 | 183 | 199 | 215 | 231 | 247 |
| 8 | 24 | 40 | 56 | 72 | 88 | 104 | 120 | 136 | 152 | 168 | 184 | 200 | 216 | 232 | 248 |
| 9 | 25 | 41 | 57 | 73 | 89 | 105 | 121 | 137 | 153 | 169 | 185 | 201 | 217 | 233 | 249 |
| 10 | 26 | 42 | 58 | 74 | 90 | 106 | 122 | 138 | 154 | 170 | 186 | 202 | 218 | 234 | 250 |
| 11 | 27 | 43 | 59 | 75 | 91 | 107 | 123 | 139 | 155 | 171 | 187 | 203 | 219 | 235 | 251 |
| 12 | 28 | 44 | 60 | 76 | 92 | 108 | 124 | 140 | 156 | 172 | 188 | 204 | 220 | 236 | 252 |
| 13 | 29 | 45 | 61 | 77 | 93 | 109 | 125 | 141 | 157 | 173 | 189 | 205 | 221 | 237 | 253 |
| 14 | 30 | 46 | 62 | 78 | 94 | 110 | 126 | 142 | 158 | 174 | 190 | 206 | 222 | 238 | 254 |
| 15 | 31 | 47 | 63 | 79 | 95 | 111 | 127 | 143 | 159 | 175 | 191 | 207 | 223 | 239 | 255 |

Figure 2-1. Grayscale image and its matrix representation [].

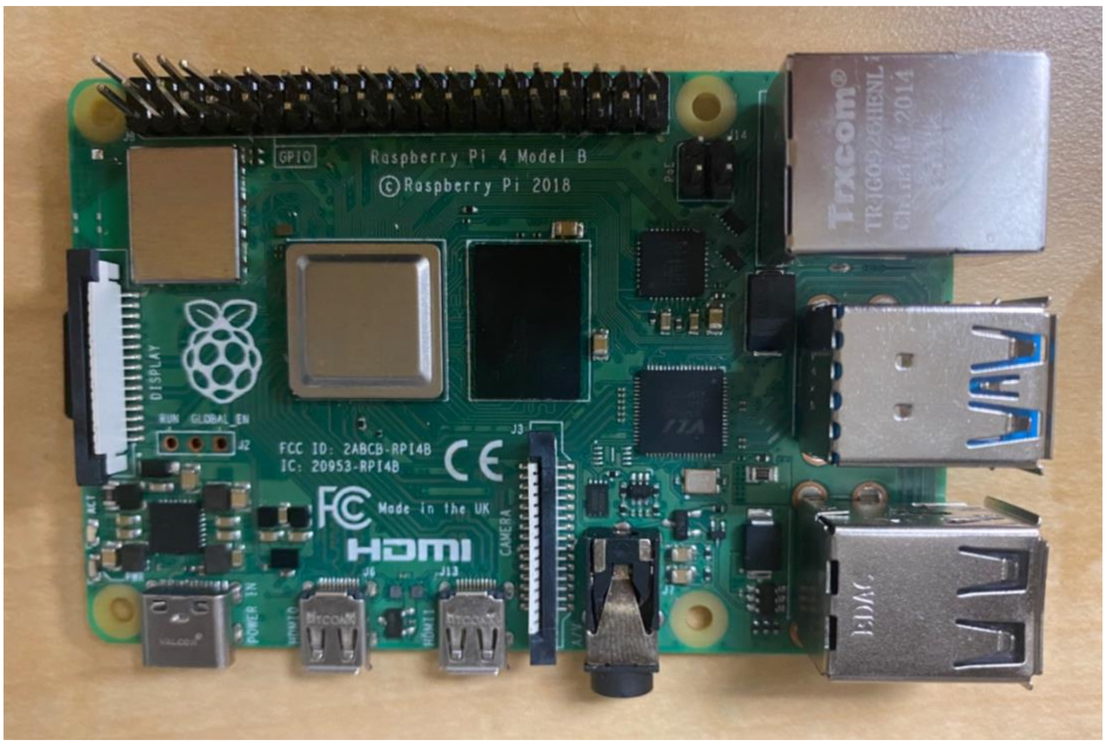



Figure 2-2. Raspeberry Pi 4B+

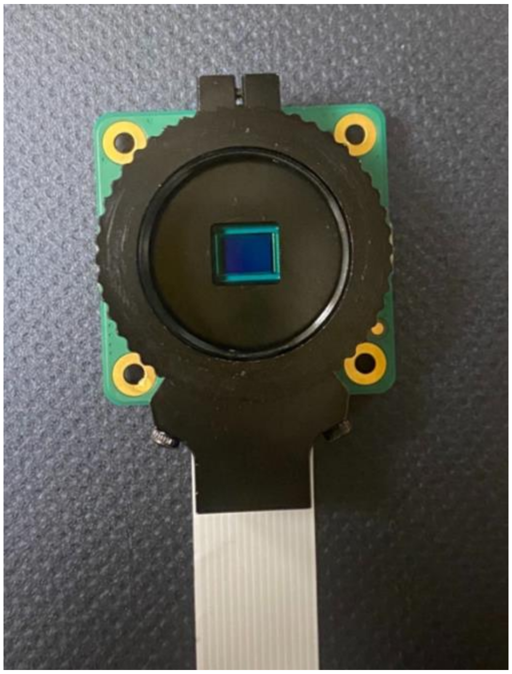


Figure 2-3. Sony IMX477 sensor, along with an Flexible Printed Circuit (FPC)  cable

Figure 2-4. Pimoroni Microscope Lens


Figure 2-5. 6mm Wide Angle Raspberry Pi Lens


Figure 2-6. 10mm Telephoto Lens

Figure 2-7. 300x Microscope lens compatible with HQ Camera

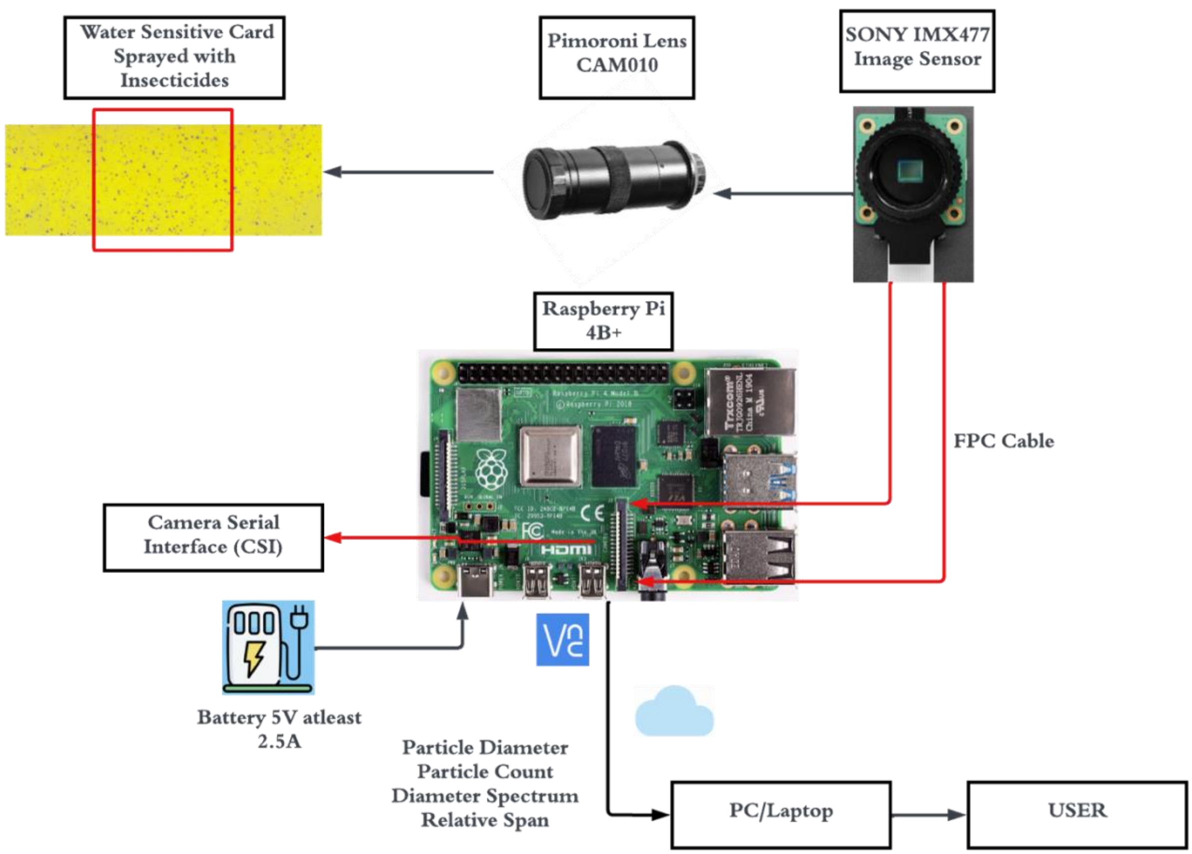



Figure 2-8. Hardware Diagram for proposed solution

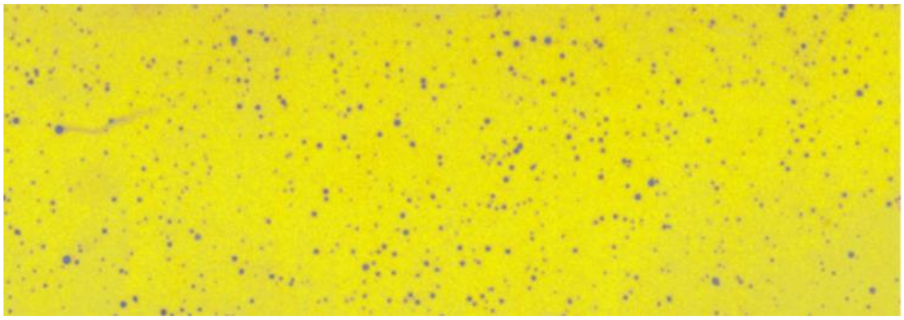


Figure 2-9. 3*1 inch Image Water Sensitive Card sprayed with Insecticides in the AMCD Field

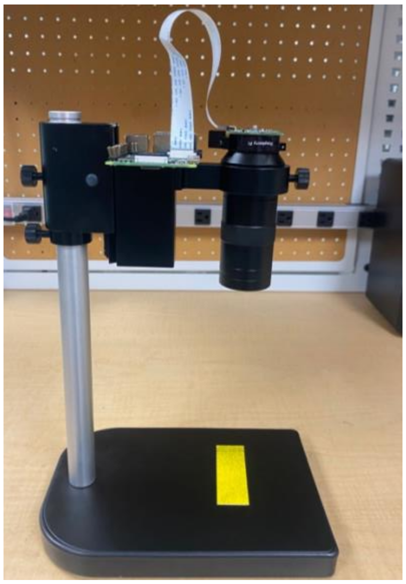


Figure 2-10. Raspberry Pi and Pimoroni lens microscope setup

Figure 2-11. Binary Image Output with T=100

Figure 2-12. Binary Image Output with T=120

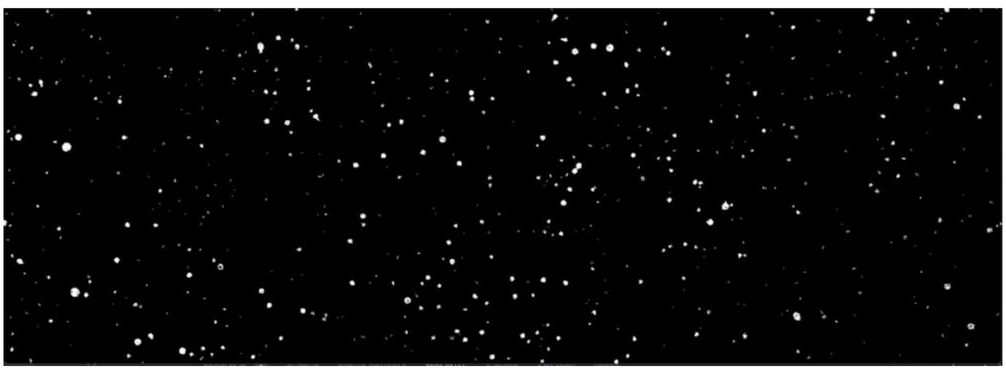


Figure 2-13. Binary Image Output with T=140

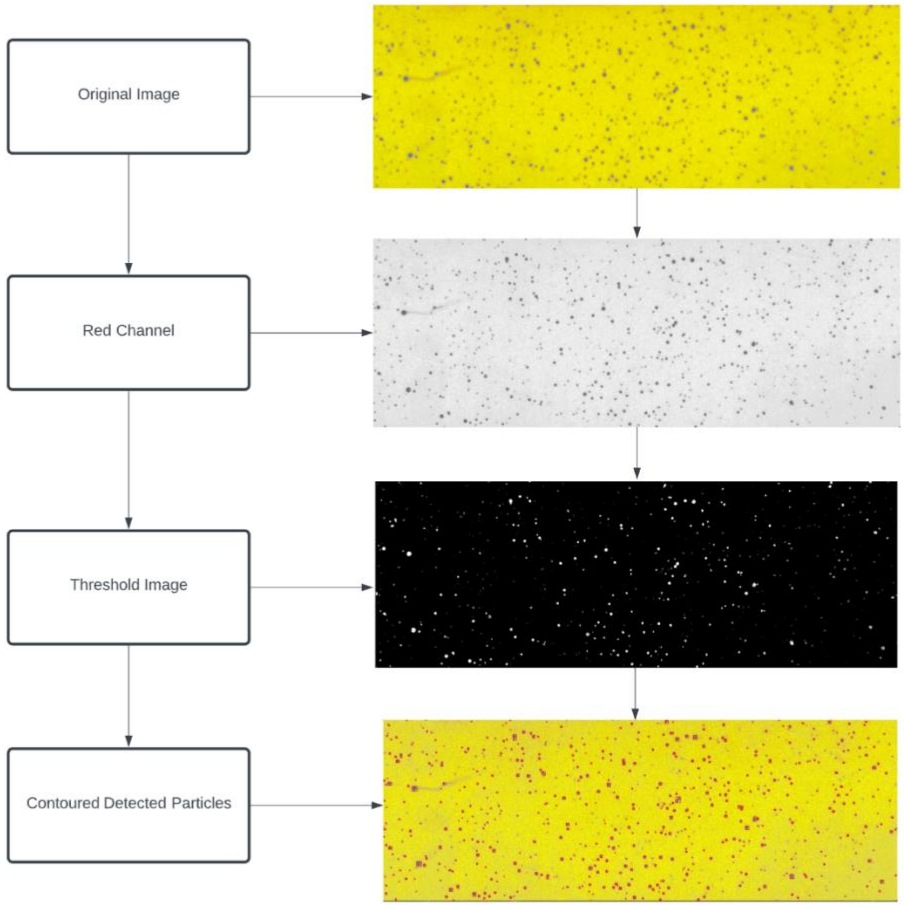



Figure 2-14. Step by step image analysis shown for image of size 3*1 obtained from the AMCD.

# CHAPTER 3
# PROTOTYPE

A typical mosquito pole sketch with a spinner is as shown in Figure 3-1. As explained in Chapter 2, this is the current framework for insecticide testing at the AMCD. The hardware setup proposed in Chapter 2 could be deployed on each mosquito pole at the AMCD for real-time droplet analysis.

Figure 3-2 gives an outline of how this can be accomplished.The new framework will consist of the same mosquito pole as a base. The opening where the spinner is usually attached can be split into two openings by attaching an extender. The opening positioned away from the mosquito pole will be used to place the spinner and the opening towards the mosquito pole can be used place an additional framework on which the RPI camera and lens can be attached. This proposed framework is shown below the mosquito pole in the Figure 3-2. It consists of an adjustable sliding rod, which could be motor operated. Over this the RPI camera and lens can be attached on a slider which also works on a motor that moves up and down the adjustable sliding rod. Ideally the RPI lens will be placed at a distance of 6.5 inches from the slides mounted on the spinner. Once the insecticide is sprayed, the spinners are switched off and they return to their idle state. The RPI can be programmed to take the image of the first slide and move horizontally over the sliding rod, take an image of the second slide and produce results, which will be saved for analysis. This prototype could give real-time droplet analysis with minimal funds. It can be operated remotely. The instant analysis save hours of labour for testing slides.

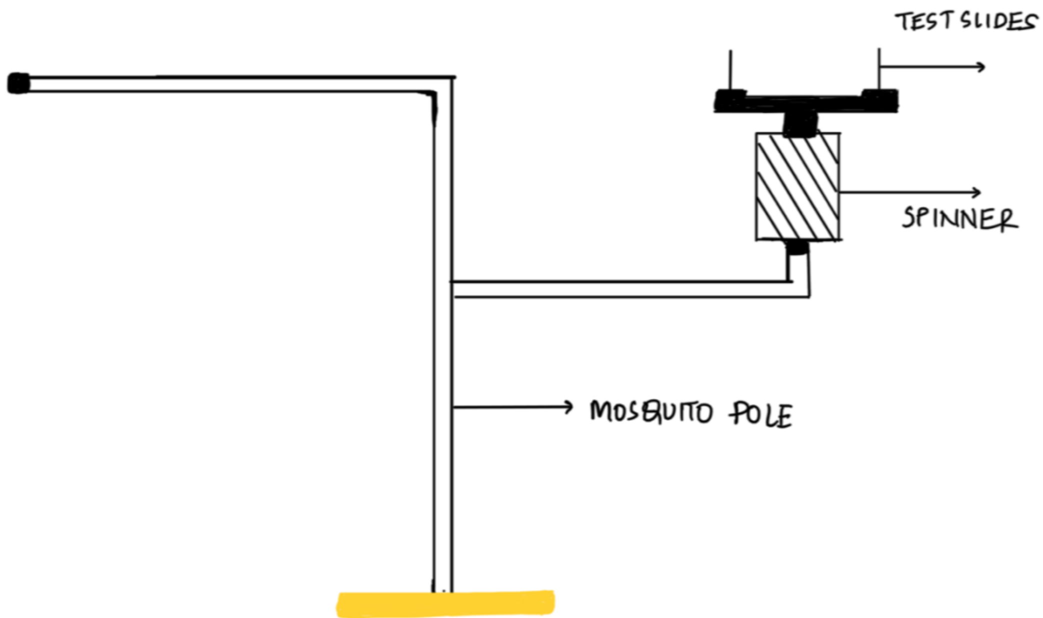



Figure 3-1. Sketch of Mosquito Pole with mounted spinner

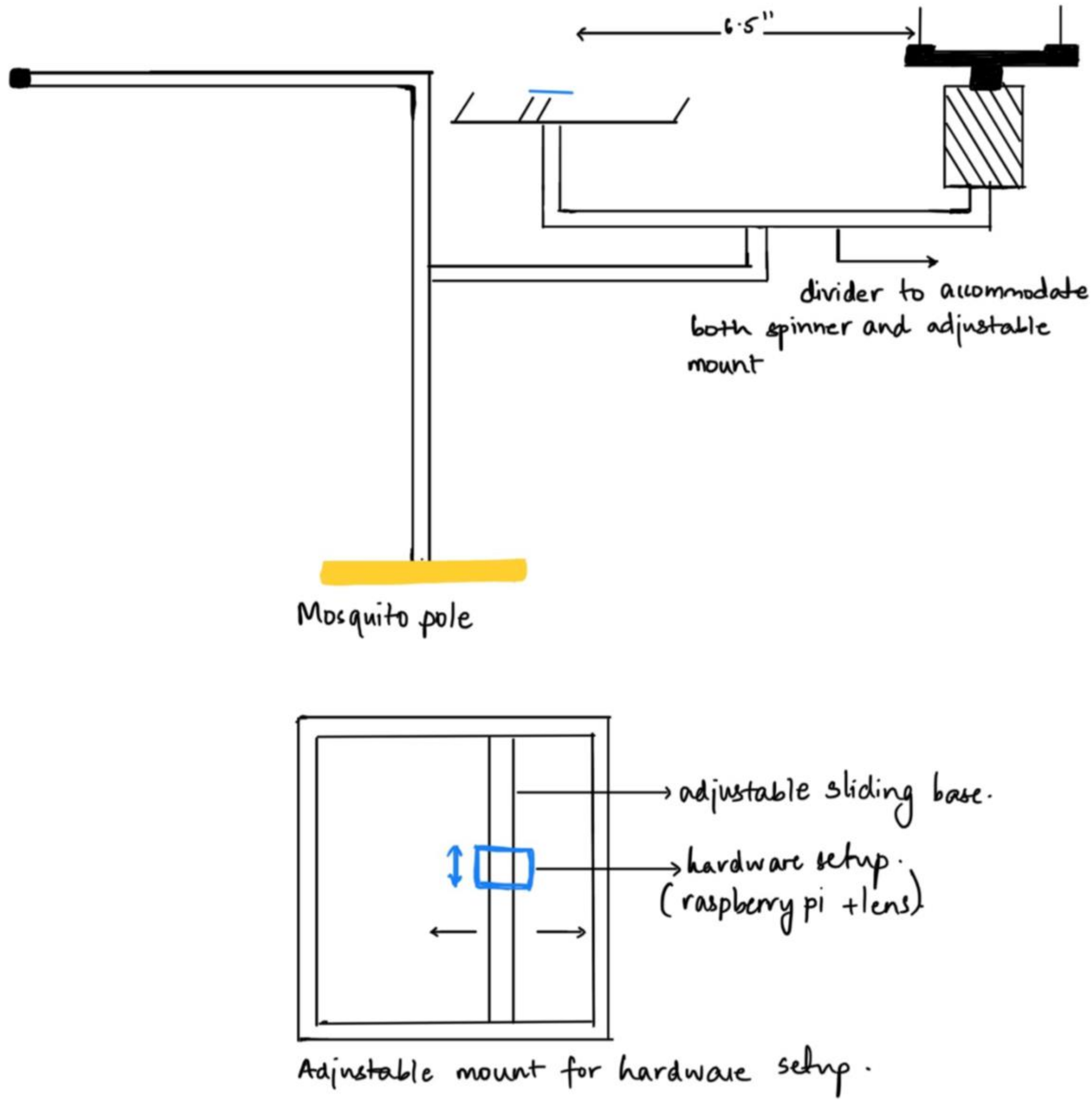



Figure 3-2. Mosquito pole with an extender for spinner and adjustable mount for hardware setup

# CHAPTER 4
# RESULT AND CONCLUSION

## Results: AMCD Images

The first set of images used to develop the algorithm were obtained from the AMCD. The input images are as shown in Figure 4-1, 4-4,4-7, 4-10 ,4-13. The Figures 4-2, 4-5, 4-8, 4-11, 4-14 show the output images after the running the algorithm. The input images go through the same image processing algorithm as shown in Figure 2-13. The output images clearly show the detected droplets considered for measurement analysis.

The algorithm gives an output csv file, with image names and their corresponding droplet count, diameter volume spectrum ($DV_{0.1}$, $DV_{0.5}$, $DV_{0.9}$), relative span as shown in Figure 4-16.

One of the main features that this tool resembles with the droplet analyzer used at the AMCD is the output histogram analysis of diameter volume spectrum. The histogram outputs corresponding to the input images are as shown in Figure 4-3, 4-6, 4-9, 4-12, 4-15.

The histogram has been plotted with the droplet diameter in μm on the x-axis and number of particles on the y-axis. This histogram analysis is limited to each sample slide just like the droplet analyzer at the AMCD. The histogram divides the number of particles on the sample slide into bins on the basis of their diameters. The histogram obtained not only gives information about the number of particles but also gives a thorough analysis of the $DV_{0.1}$, $DV_{0.5}$, $DV_{0.9}$ values. Three vertical lines drawn with the help of the python function axvline denote $DV_{0.1}$, $DV_{0.5}$, $DV_{0.9}$ from the left to the right of the histogram. The values mentioned above the lines correspond to the $DV_{0.1}$, $DV_{0.5}$, $DV_{0.9}$ values respectively.

A brief explaination of an example analysis experts perform using output values, at the AMCD is given below.

Let us compare the $DV_{0.1}$ values of images A25 and A46 which are respectively 91 and 0.02 microns approximately. A $DV_{0.1}$ value shows that the spray consists of 10% by volume droplets smaller than the $DV_{0.1}$. In this case, A25 has droplets 10% by volume smaller than 91 microns and which still has overall bigger droplet size compared to A46, which has droplets 10% by volume smaller than 0.02 microns. Hence, the insecticide tested on sample slide A46 has more driftable properties compared to the insecticide tested on A25.

Now let us compare their $DV_{0.9}$ values of images A25 and A46 which are respectively 673 and 315 approximately. A $DV_{0.9}$ value denotes that the spray contains 10% by volume droplets larger than the $DV_{0.9}$ value. A larger $DV_{0.9}$ value proves that spray is less airborne and 10% by volume of spray might settle down before reaching the intended target.

The VMD/ $DV_{0.5}$ values of A25 and A46 are 289 and 62. This denotes they are suitable for VMD less than 145 microns is effective for flying insects as it remains in the air stream, VMD between 145-325 microns works for stems and narrow vertical leaves such as grass,

The relative span values of A25 and A46 are 2 and 4 respectively. The smaller this value, the lesser the variation between the size of the droplets in the spray spectrum. respectively. Based on the RS of the latter spray, it can be inferred that it has a lesser variation among drop sizes, will drift less and have more coverage.

### Results: Real-Time Testing

In this test, three water-sensitive card were sprayed with water consecutively. Images were taken using the microscope stand setup shown in Figure 2-9. An image analysis pipeline created using PiCamera ran the algorithm on these images one by one. The input images are as shown in Figure 4-17, 4-20, 4-23. The results are obtained instantly and stored in the order they were analysed in. Outputs procured are just as the previous section. The contoured output images are as shown in Figure 4-18, 4-21, 4-24.

This test signifies that the solution can capture multiple images without the need of any calibration and provide number of particles, diameter volume spectrum values and relative span for them. This data is stored systematically as a csv file, which can be viewed instantly or later for further analysis.

### Testing The Reliability Of The Droplet Imager

To test the reliability of this Droplet Analysis and Measurement device a single water sensitive card was sprayed with water. An image of the same slide was taken thrice using the PiCamera pipeline as shown in Figure 4-27,4-30, 4-33 . Just like in the previous section the algorithm was applied to the images in a loop.The output images are as shown in Figure 4-28, 4-31, 4-34. The results for number of particles, diameter volume spectrum values and relative span were similar as shown in Figure 4-36 as a CSV output. The csv output clearly shows the similar results for the number of particles, diameter volume spectrum and relative span for all three input slides. This shows that the results obtained from the droplet analyzer are reliable.

## Conclusion

The droplet analyzer created during this research is a portable device that capture images in the field and give instant droplet size and count measurements. The immediate results can save a lot time. The system outputs a histogram analysis to give a detailed understanding of diameter volume spectrum values. These results along with the relative span are stored in a CSV file for expert analysis at the AMCD. This tool developed can output results similar to the droplet analyzer used at the AMCD at a much lower cost.

## Future Scope

This droplet analyzer can test water-sensitive cards, however, the Artium droplet analyzer can test Magnesium Oxide and Teflon slides/rods. This research can be extended to test sample slides other water-sensitive cards.

This research works in the field, however, only a basic prototype has been prepared to setup this system in the field at the AMCD. Multiple hardware setup tests can be carried out to come up with an efficient, cost-effective and environmental-friendly mechanical design to deploy this system permanently in the field.

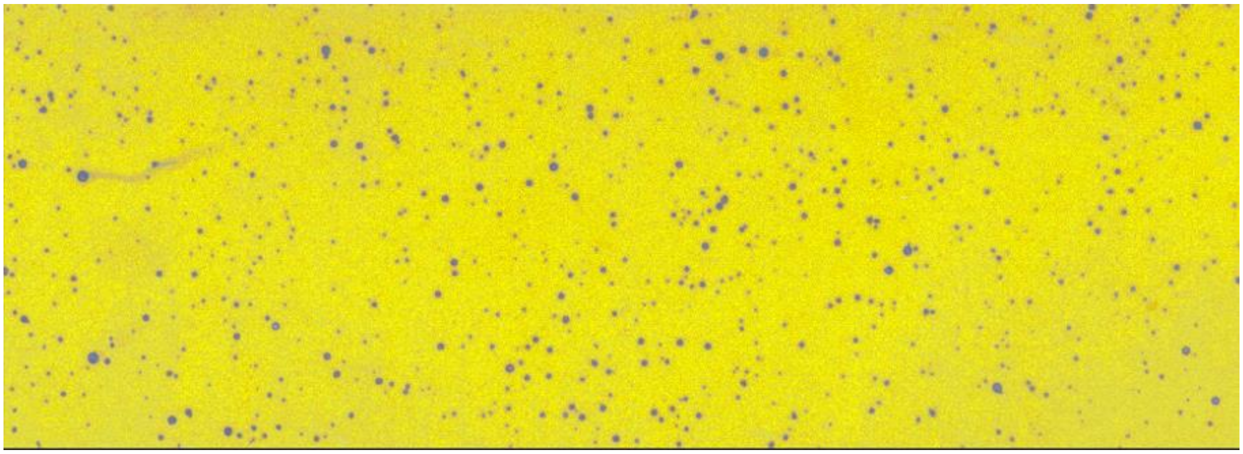


Figure 4-1. A-16 Input image

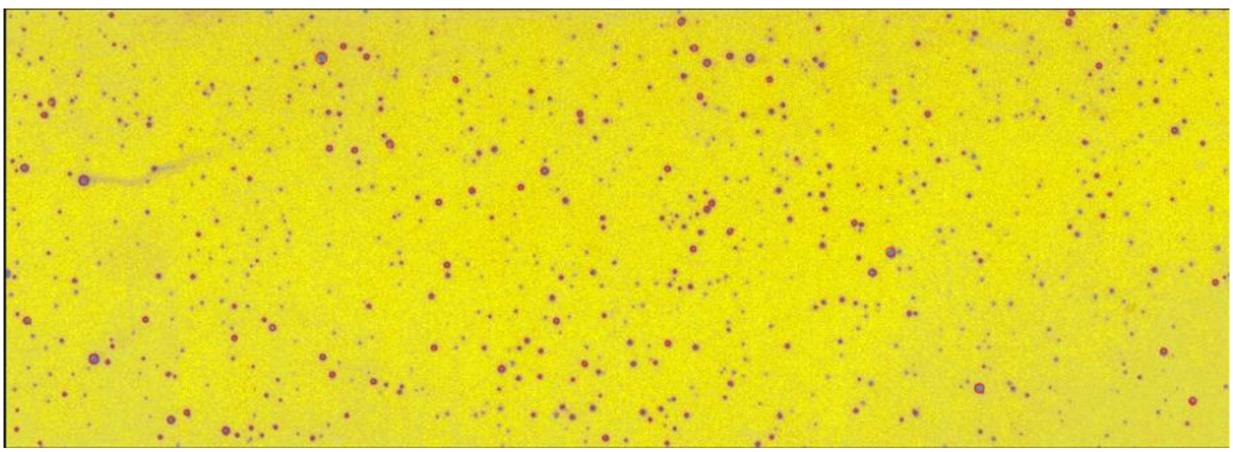


Figure 4-2. A-16 Output image

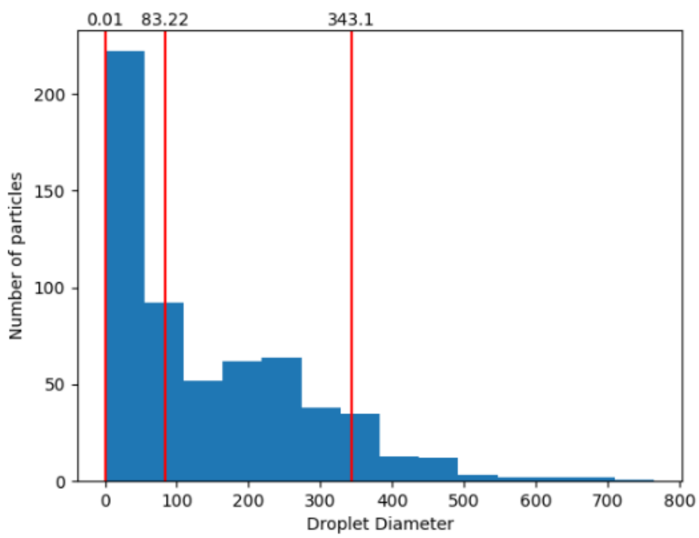



Figure 4-3. A16 Output histogram

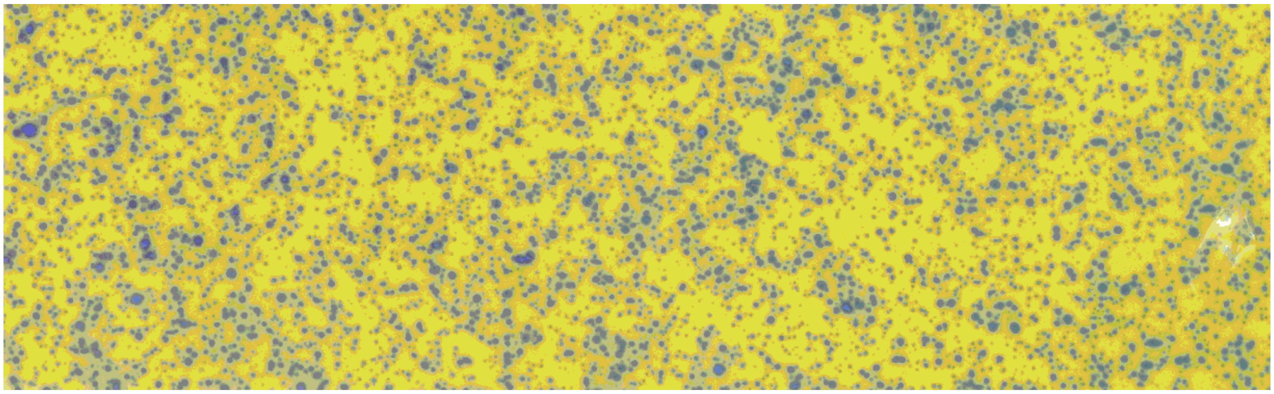


Figure 4-4. A-25 Input image

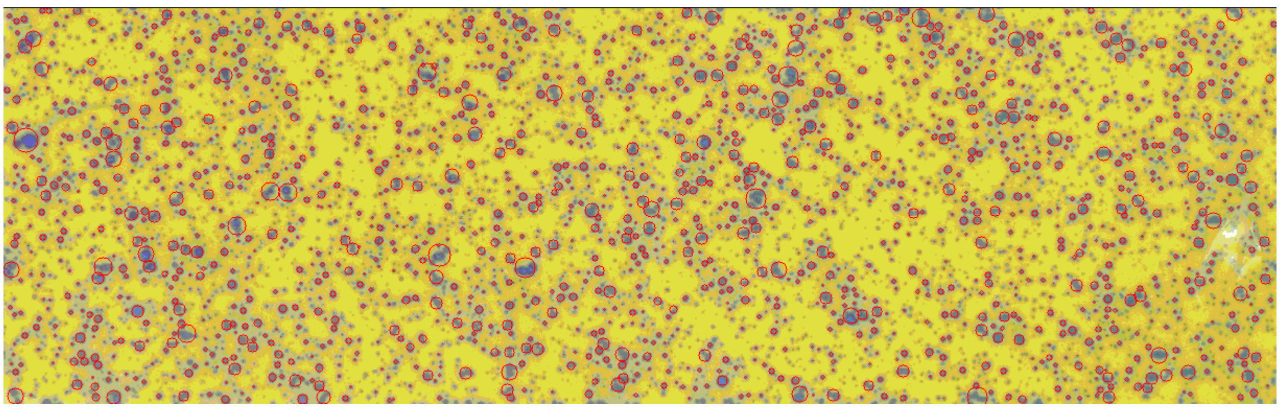


Figure 4-5. A-25 Output image

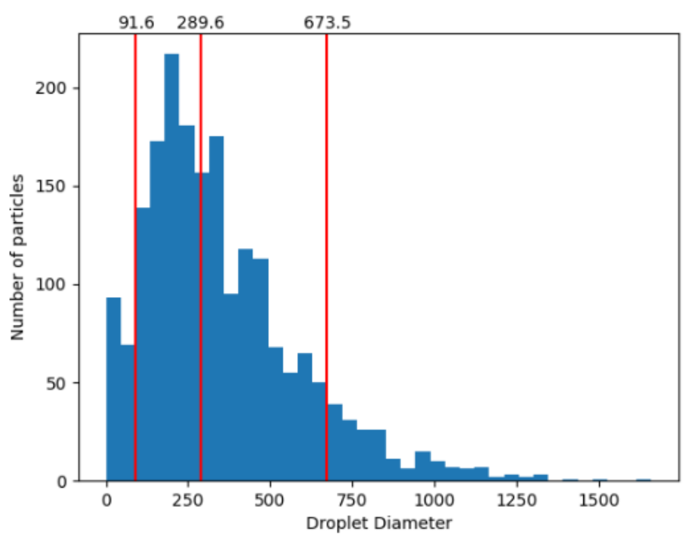



Figure 4-6. A25 Output histogram

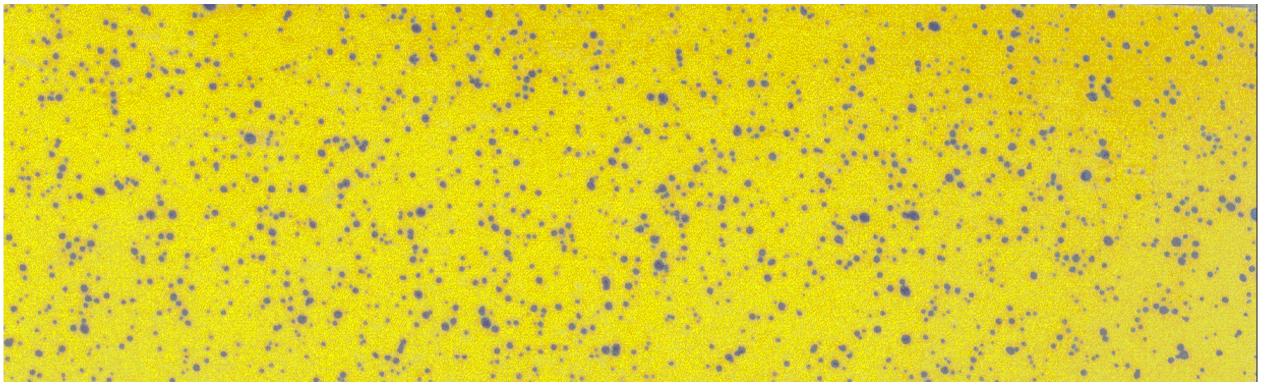


Figure 4-7 A-26 Input image

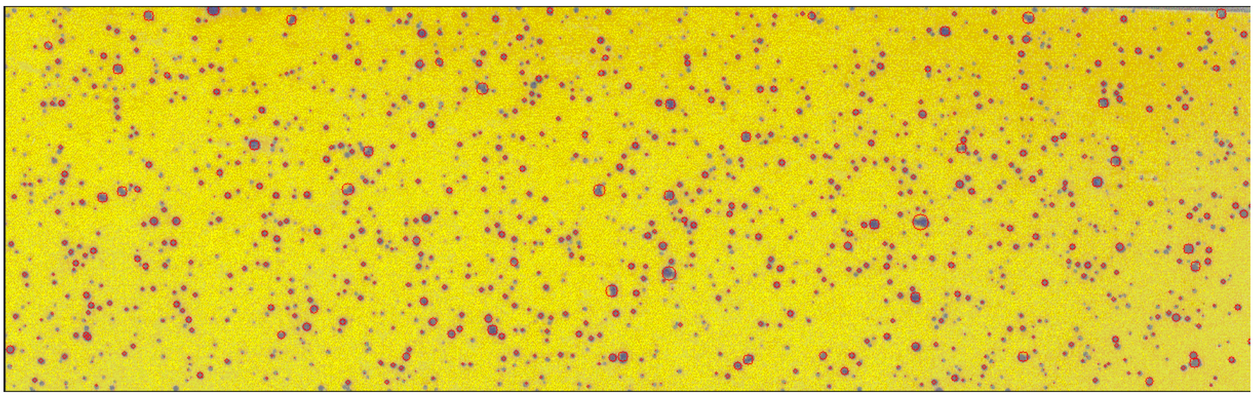


Figure 4-8. A-26 Output image

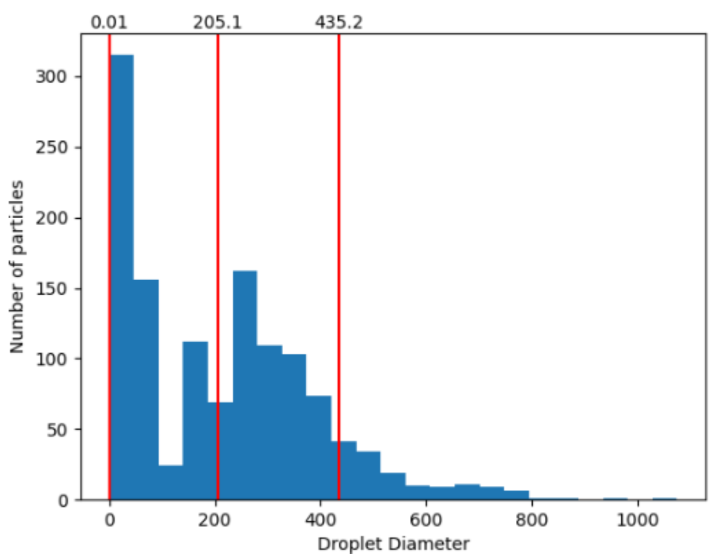



Figure 4-9. A-26 Output histogram

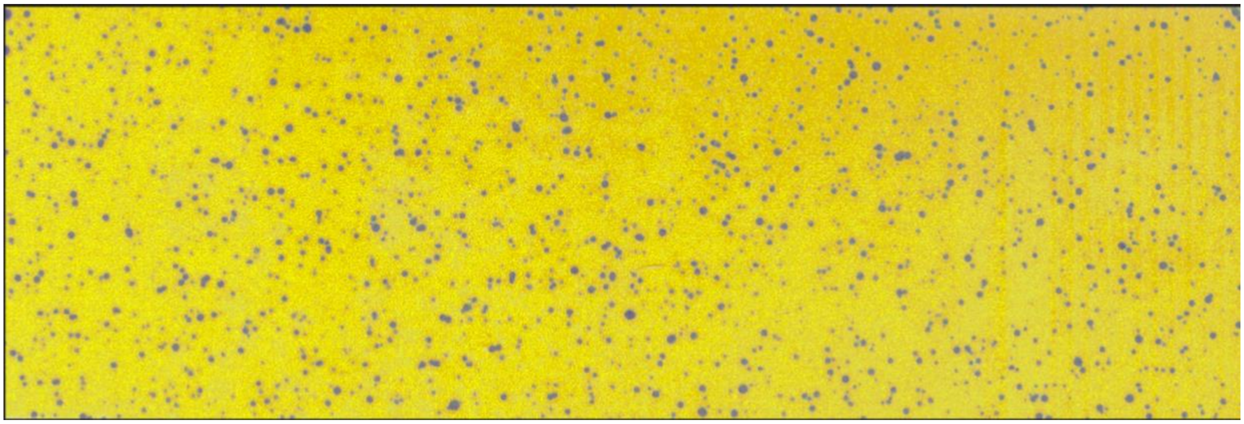


Figure 4-10. A-36 Input image

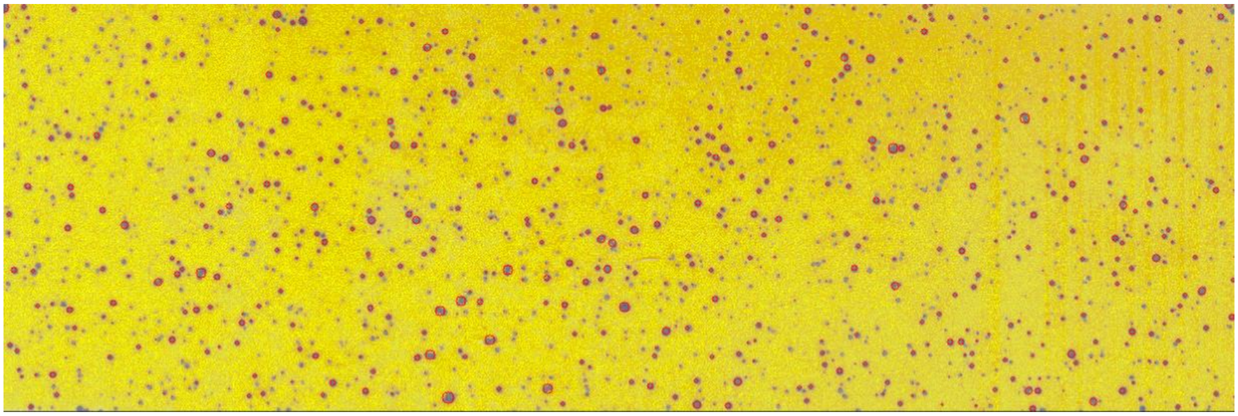


Figure 4-11.  A-36 Output image

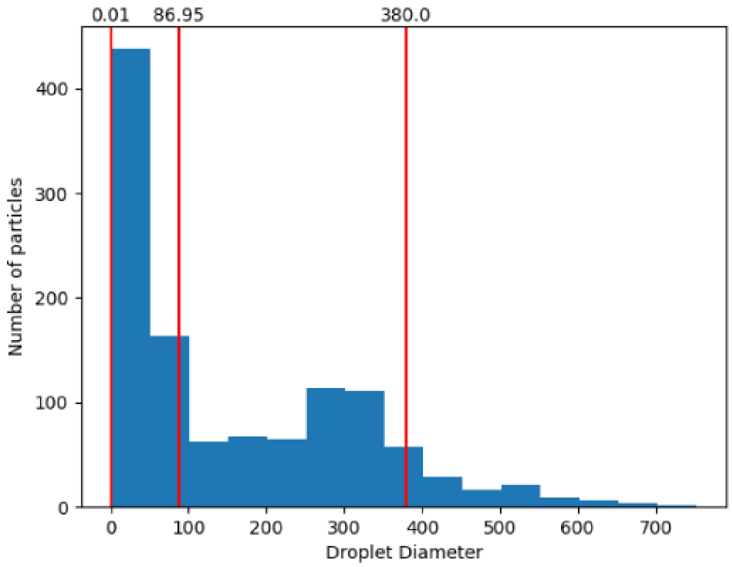



Figure 4-12. A-36 Output histogram

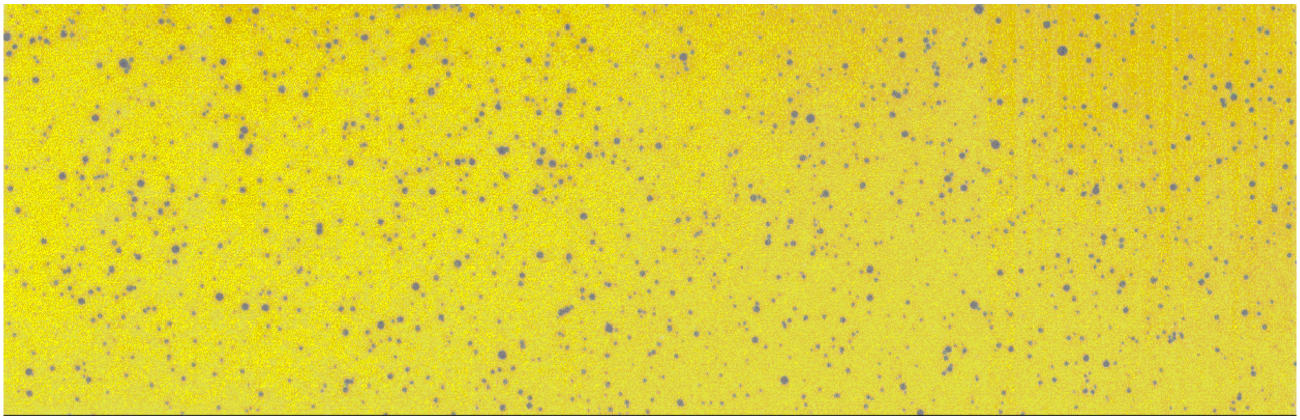


Figure 4-13. A-46 Input image

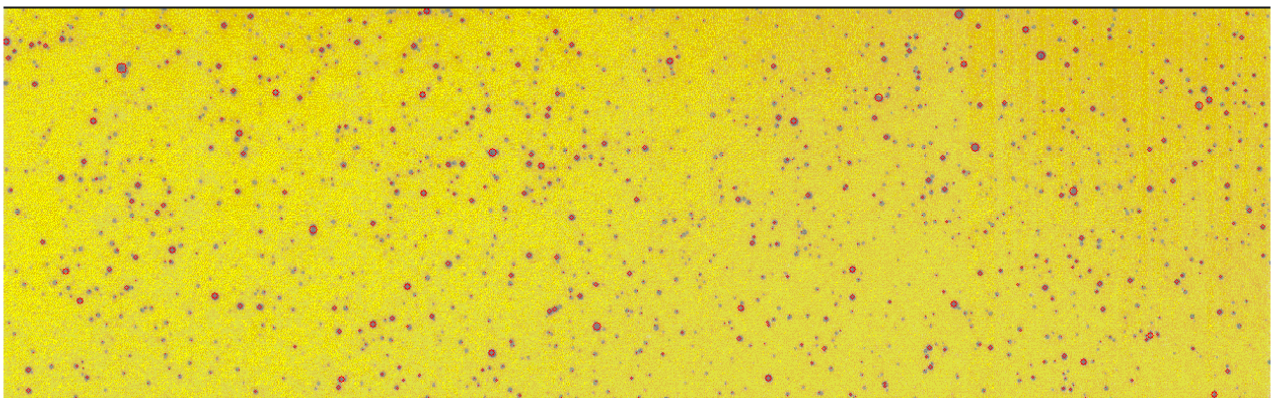


Figure 4-14. A-46 Output image

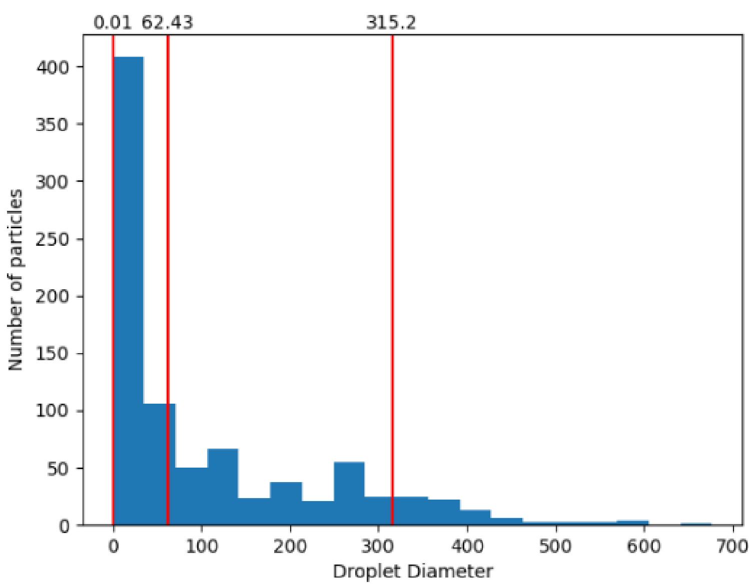



Figure 4-15. A-46 Output histogram

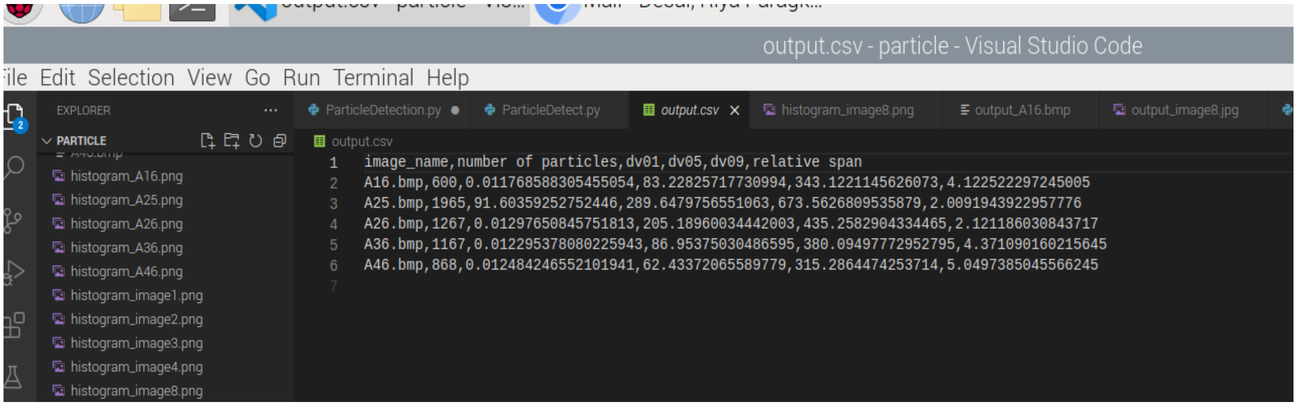



Figure 4-15. Output CSV for AMCD sample slides

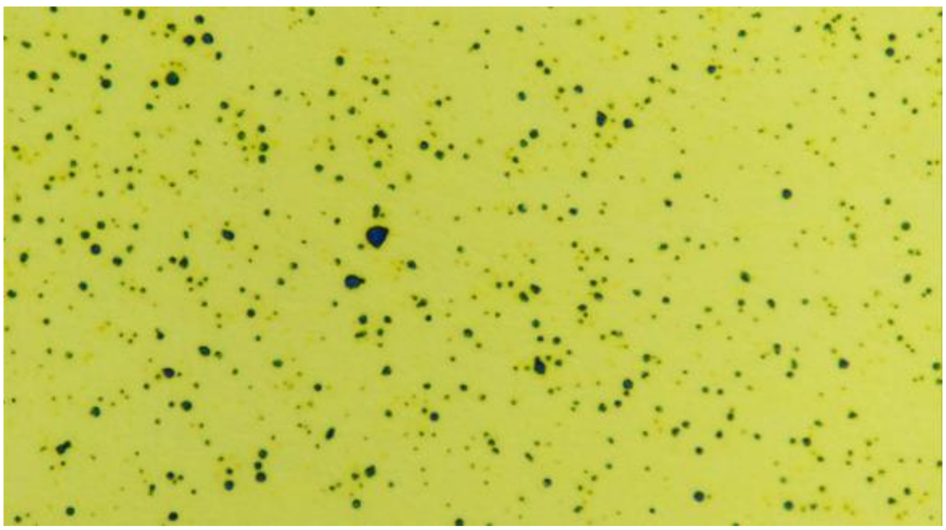


Figure 4-17. Input image 1 for real-time analysis

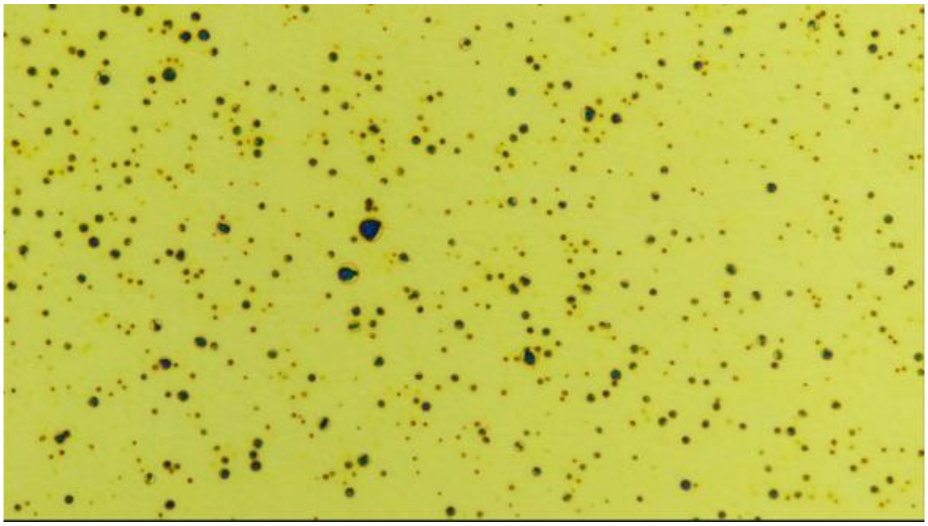


Figure 4-18. Ouput image 1 for real-time analysis

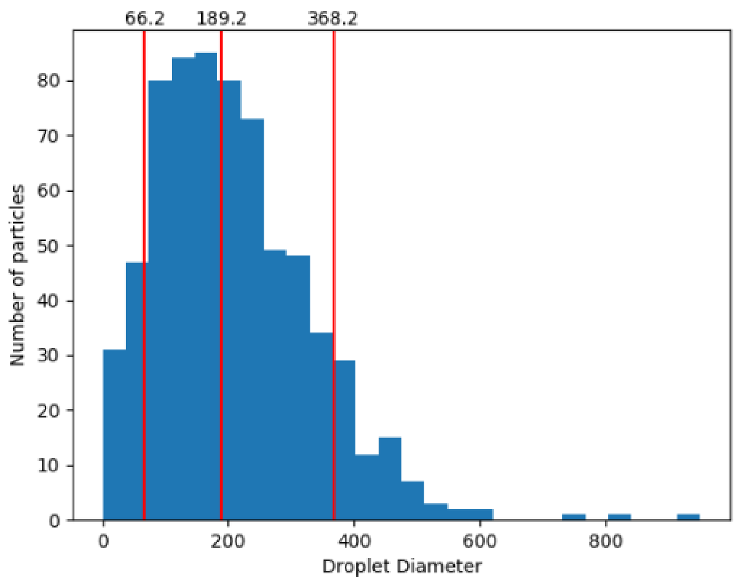



Figure 4-19. Output image 1 histogram for real-time analysis

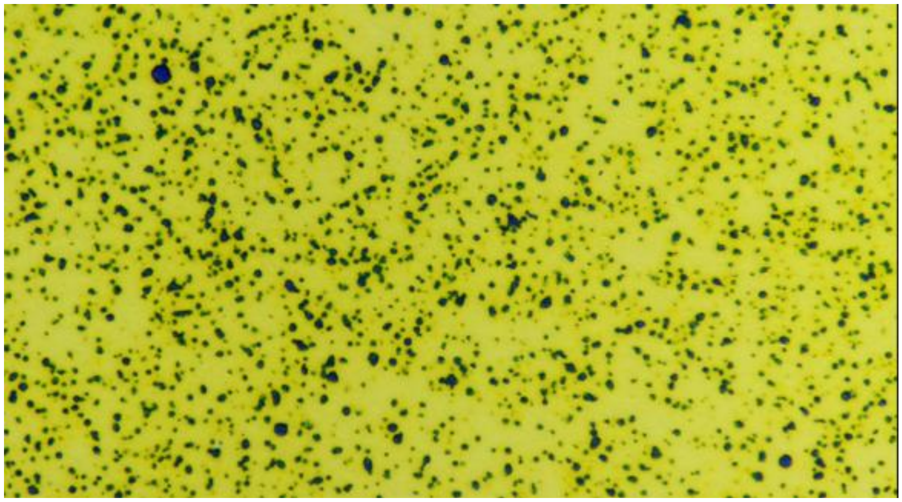


Figure 4-20. Input image 2 for real-time analysis

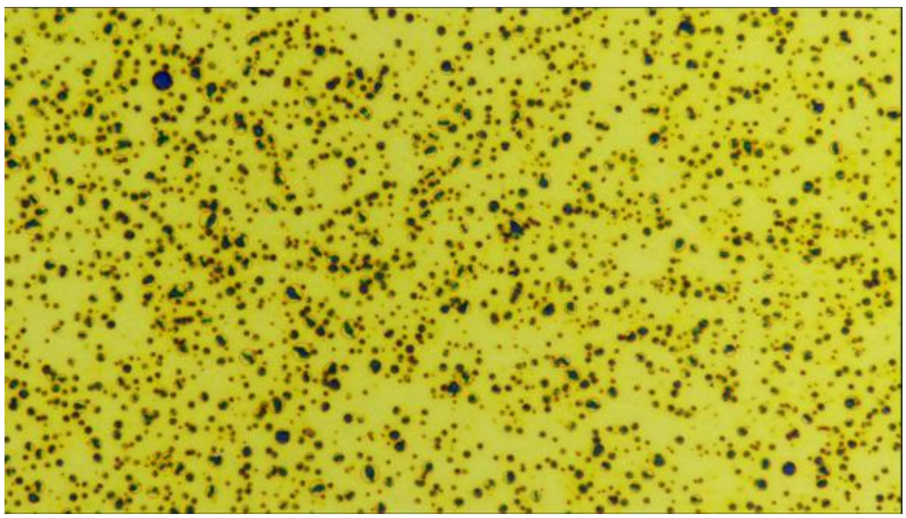


Figure 4-21. Output image 2 for real-time analysis

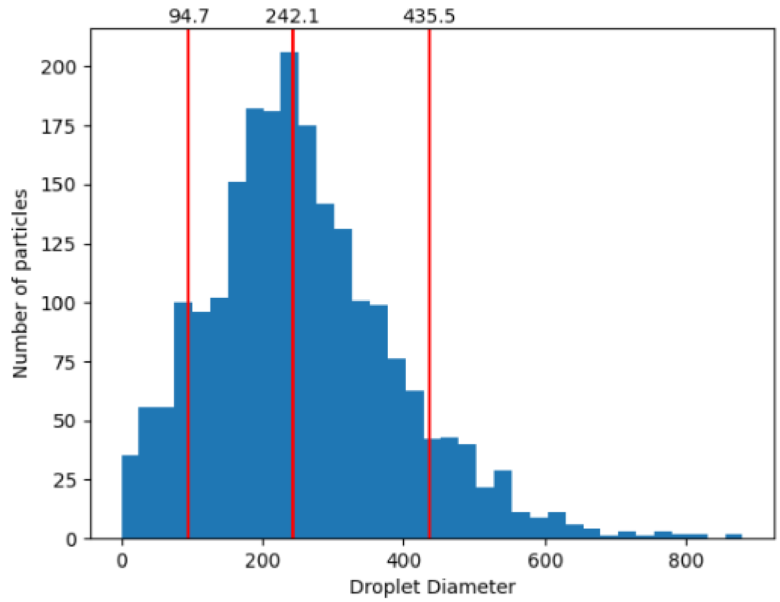



Figure 4-22. Output image 2 histogram for real-time analysis

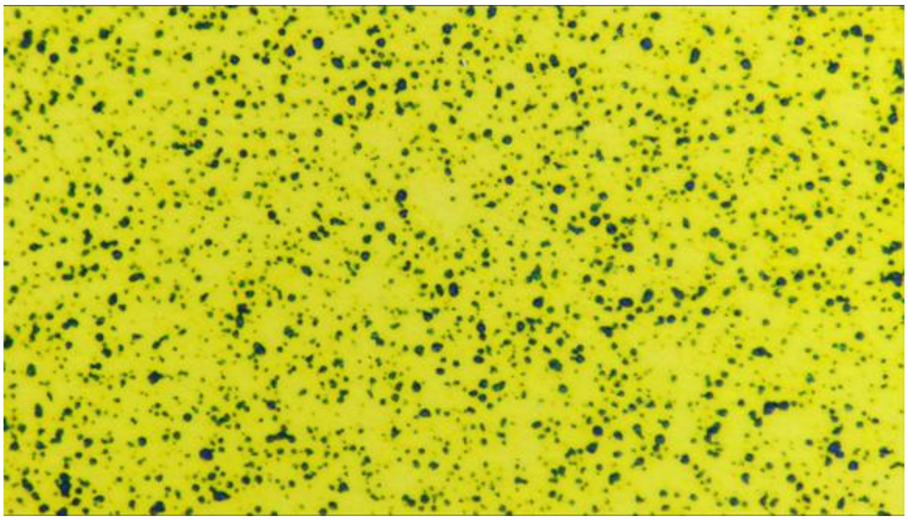


Figure 4-23. Input image 3 for real-time analysis

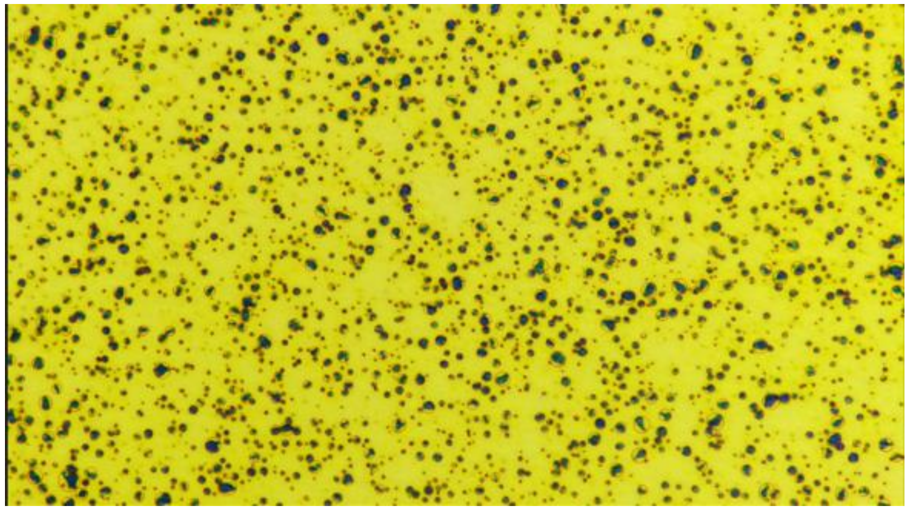


Figure 4-24. Output image 3 for real-time analysis

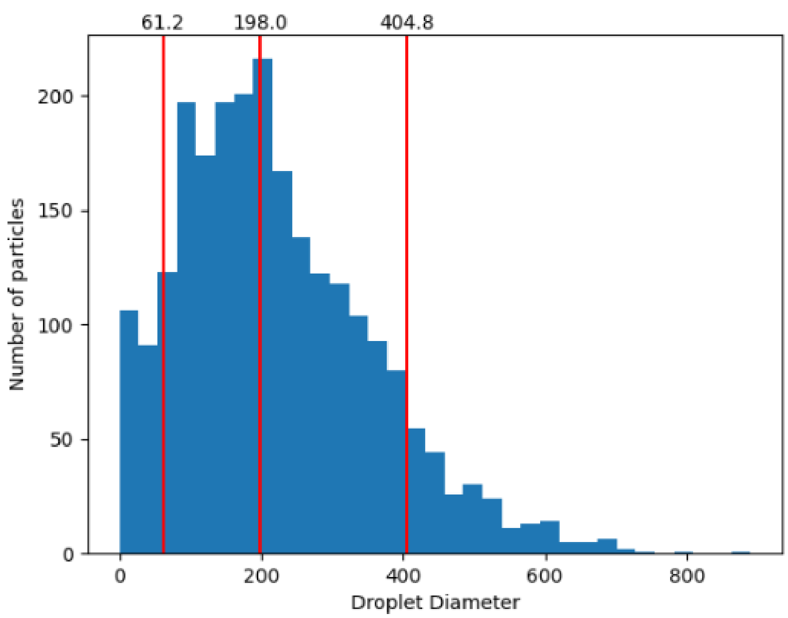



Figure 4-25. Output image 3 histogram for real-time analysis

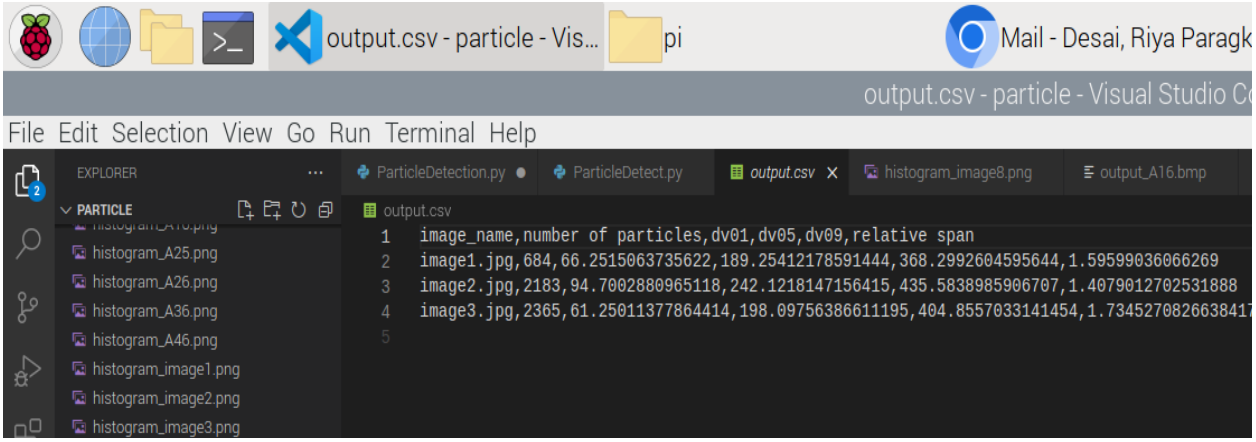



Figure 4-26. Output CSV for real-time sample slides

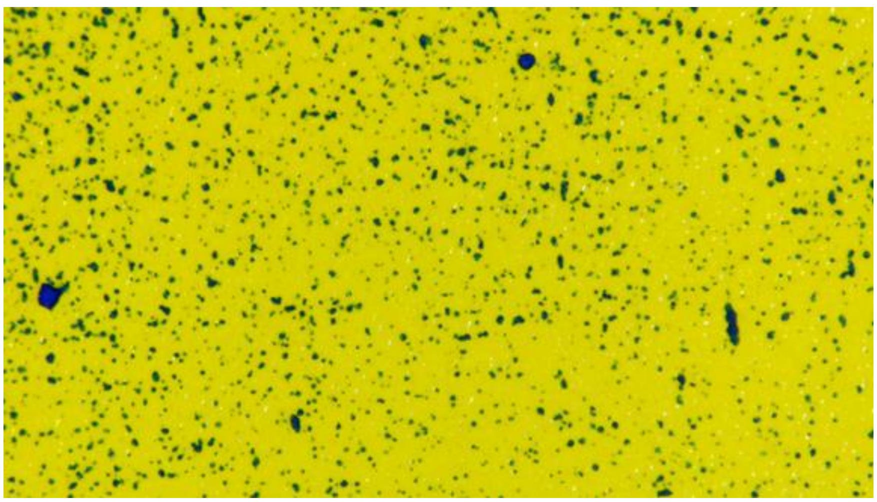


Figure 4-27. Input image 1 for reliability test

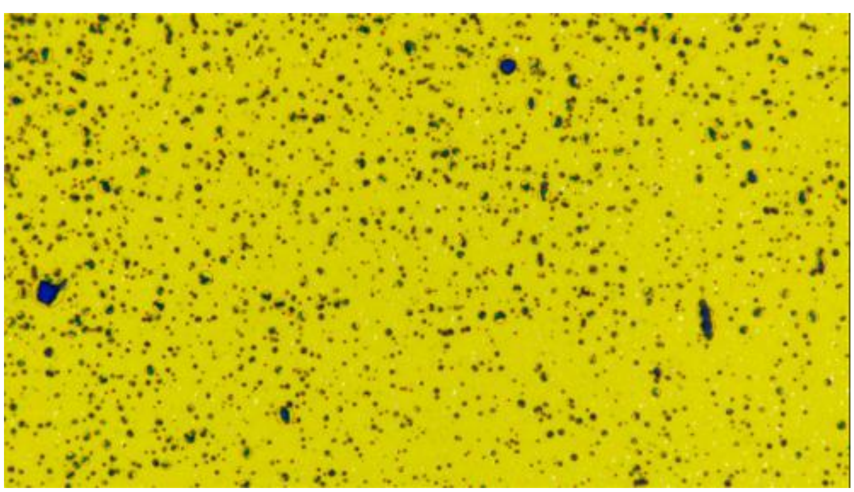


Figure 4-28. Output image 1 for reliability test

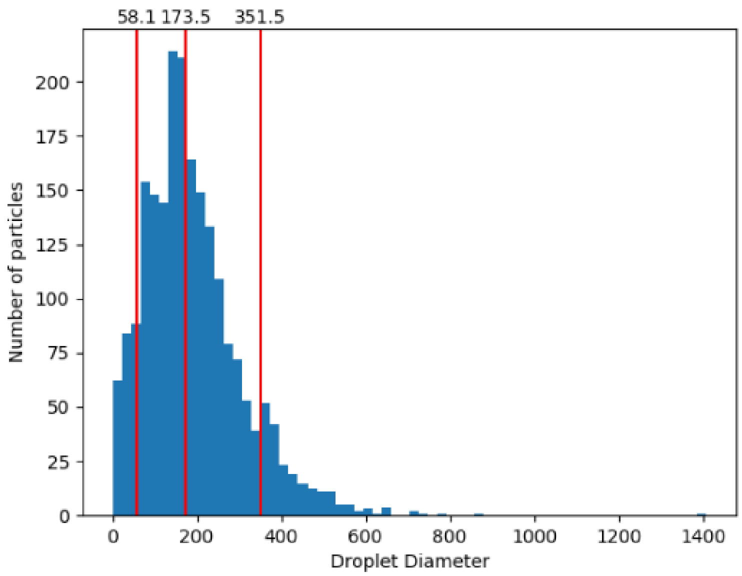



Figure 4-29. Output image 1 histogram for reliability test

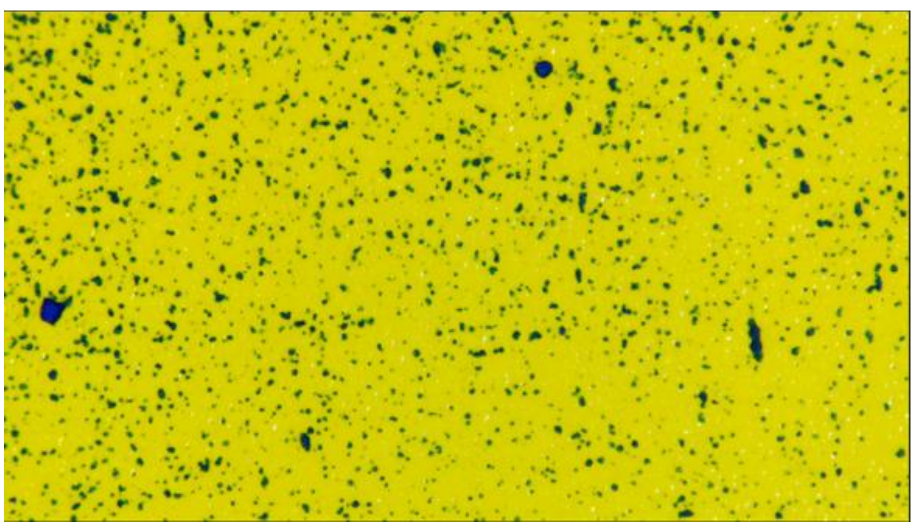


Figure 4-30. Input image 2 for reliability test

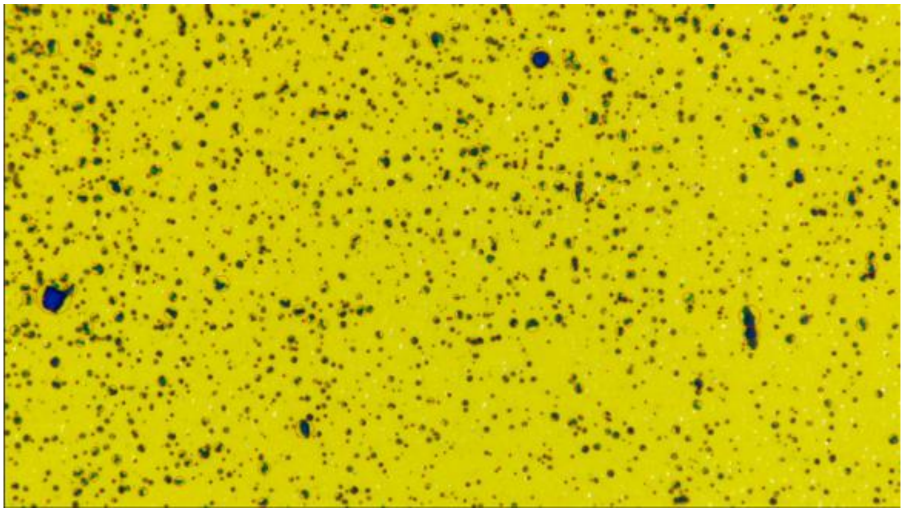


Figure 4-31. Output image 2 for reliability test

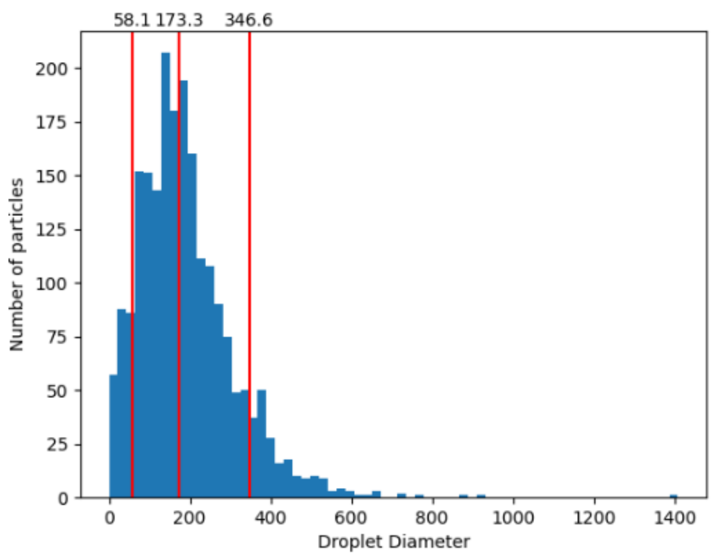



Figure 4-32. Output image 2 histogram for reliability test

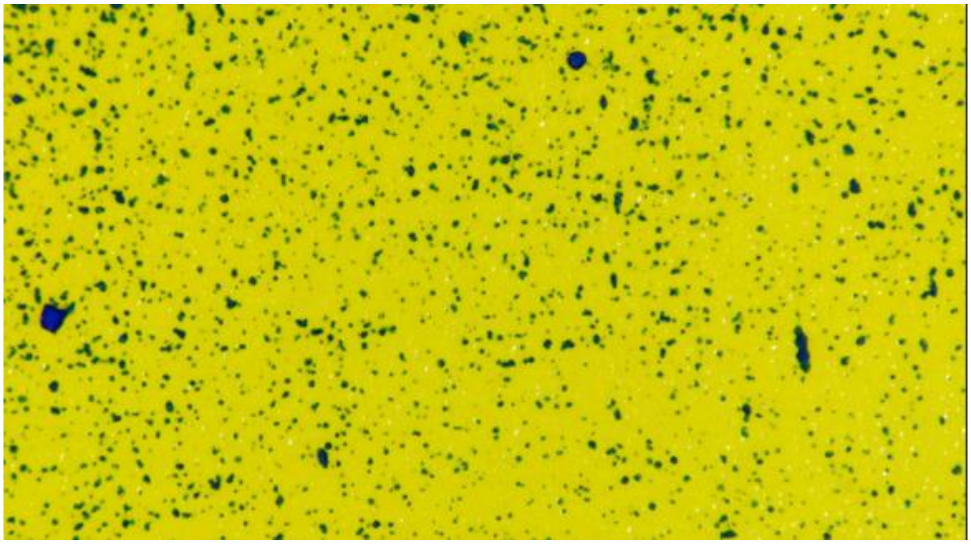


Figure 4-33. Input image 3 for reliability test

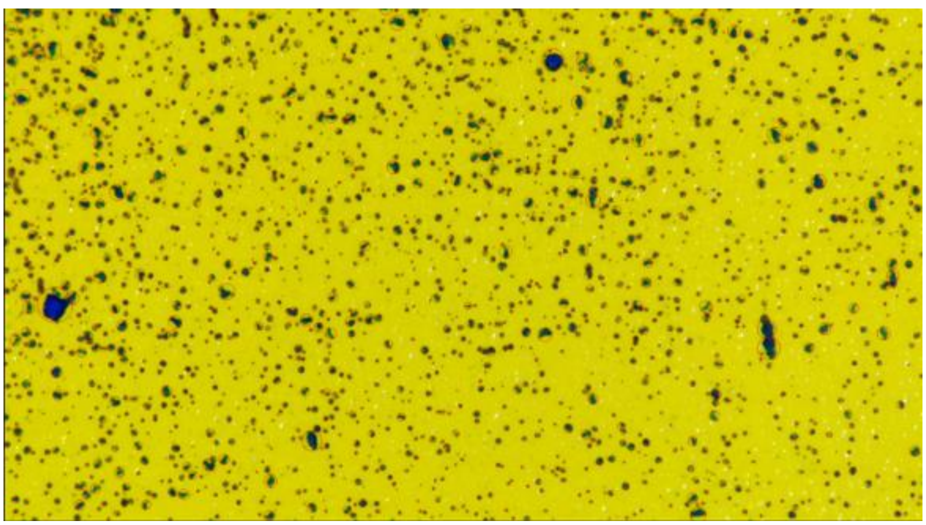


Figure 4-34. Output image 3 for reliability test

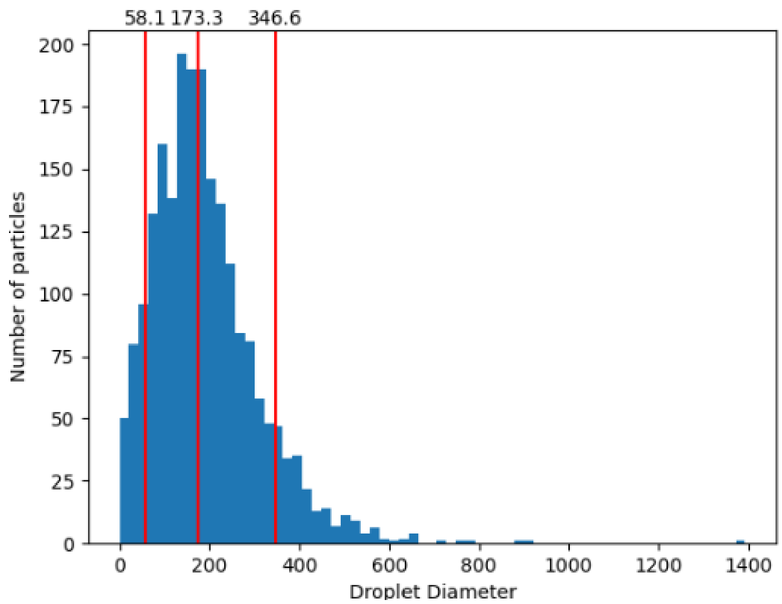



Figure 4-35. Output image 3 histogram for reliability test

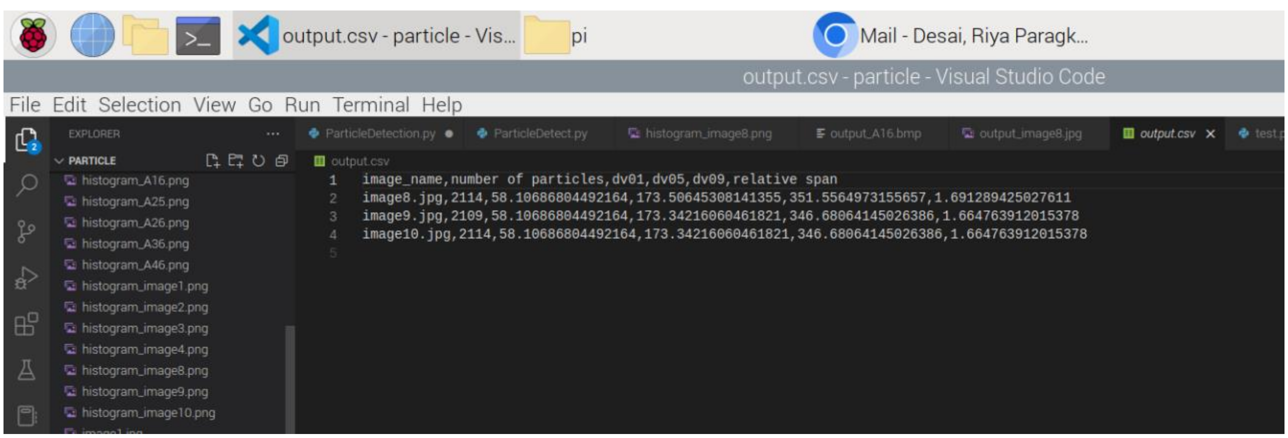



Figure 4-36. Output CSV for reliability test

## LIST OF REFERENCES

[1] O. US EPA, “Introduction to Pesticide Drift,” US EPA, Aug. 01, 2014. Available: https://www.epa.gov/reducing-pesticide-drift/introduction-pesticide-drift

[2] “Continuing EduCation Earn 1 CEU in Integrated Pest Management Factors affecting spray drift of pesticides.” Available: https://www.certifiedcropadviser.org/files/certifications/certified/education/self-study/exam-pdfs/119.pdf

[3] Beyaz, Abdullah & Dağtekin, Metin & ÇİLİNGİR, İbrahim & Gerdan Koc, Dilara. (2017). Evaluation of Droplet Size Spectra for Agricultural Pesticide Applications Using Water Sensitive Paper and Image Analysis Techniques. Fresenius Environmental Bulletin. 26. 7717–7723. Available:https://www.researchgate.net/publication/322294925_Evaluation_of_Droplet_Size_Spectra_for_Agricultural_Pesticide_Applications_Using_Water_Sensitive_Paper_and_Image_Analysis_Techniques

[4] “Anastasia Mosquito Control mosquito born diseases, St. Augustine, FL, St Johns County,” Anastasia Mosquito Control District. Available: https://amcdsjc.org

[5] “Understanding Droplet Size – Pesticide Environmental Stewardship,” Pesticidestewardship.org, 2012. Available: https://pesticidestewardship.org/pesticide-drift/understanding-droplet-size/

[6] “Publication 442-031 Droplet Chart / Selection Guide.” Available: https://www.mssoy.org/uploads/files/virginia-coop-ext.pdf

[7] “Spinning Impinger,” Leading Edge. Available: https://leateam.com/product/impingers-spinners-2/.

[8] G. A. Matthews, “Determination of Droplet Size,” PANS Pest Articles & News Summaries, vol. 21, no. 2, pp. 213–225, Jun. 1975, doi: 10.1080/09670877509411402.

[9] “SpotOn Spray Pattern Test Paper,” Gemplers. Available: https://gemplers.com/products/spoton-spray-pattern-test-paper

[10] C. Black, “A Lesson in Spray Quality and Droplet Size.” Available: https://pesticideresources.org/wps/hosted/spray-quality-droplet-size.pdf

[11] “Ultra-Low Volume Spray Machine | Central Mass Mosquito Control Project,” Available: www.cmmcp.org. https://www.cmmcp.org/adulticide-program/pages/ultra-low-volume-spray-machine.

[12] “User Manual Operation of the Turnkey Phase Doppler Interferometer (TK-PDI) Spray Drop Size and Velocity Measurement TK1 and TK2 PDI Systems High-precision measurement systems for energy, environmental, and industrial applications.” Available:https://www.artium.com/_files/ugd/ee8bc1_4a41052e3e234759878123e36fda628e.pdf

[13] “Digital Image Processing Basics - GeeksforGeeks,” GeeksforGeeks, Jan. 26, 2018. https://www.geeksforgeeks.org/digital-image-processing-basics/

[14] Maximinusjoshus, “Understanding the concept of Channels in an Image,” featurepreneur, Jun. 08, 2021. https://medium.com/featurepreneur/understanding-the-concept-of-channels-in-an-image-6d59d4dafaa9

[15] R. P. Ltd, “Raspberry Pi 4 Model B specifications,” Raspberry Pi. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/?variant=raspberry-pi-4-model-b-8gb.

[16] “DATASHEET Raspberry Pi 4 Model B Release 1,” 2019. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf

[17] “Raspberry Pi Camera Module V2 - DEV-14028 - SparkFun Electronics,” www.sparkfun.com. Available: https://www.sparkfun.com/products/14028

[18] “Microscope lens for the Raspberry Pi High Quality Camera - 0.12-1.8x,” shop.pimoroni.com. Available: https://shop.pimoroni.com/products/microscope-lens-0-12-1-8x

[19] “Pi Camera Module Interface with Raspberry Pi using Python | Raspb..,” www.electronicwings.com. Available: https://www.electronicwings.com/raspberry-pi/pi-camera-module-interface-with-raspberry-pi-using-python

[20] “Python | Thresholding techniques using OpenCV | Set-1 (Simple Thresholding) - GeeksforGeeks,” GeeksforGeeks, May 06, 2019. Available: https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-1-simple-thresholding/

[21] “scipy.ndimage.measurements.label — SciPy v0.8.dev Reference Guide (DRAFT),” library.isr.ist.utl.pt. Available: http://library.isr.ist.utl.pt/docs/scipy/generated/scipy.ndimage.measurements.label.html.

[22] A. Rosebrock, “Watershed OpenCV,” PyImageSearch, Nov. 02, 2015. Available: https://pyimagesearch.com/2015/11/02/watershed-opencv/#:~:text=The%20output%20of%20the%20watershed.

[23] “OpenCV 3 Watershed Algorithm : Marker-based Segmentation I - 2020,” Available:https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Watershed_Algorithm_Marker_Based_Segmentation.php

[24] Wu, Y.; Li, Q. The Algorithm of Watershed Color Image Segmentation Based on Morphological Gradient. Sensors 2022, 22, 8202. https://doi.org/10.3390/s22218202

[25] E. Technologies, “Fundamentals of image contours,” Medium, Jan. 02, 2020. Available: https://evergreenllc2020.medium.com/fundamentals-of-image-contours-3598a9bcc595#:~:text=Image%20contouring%20is%20process%20of.

# BIOGRAPHICAL SKETCH

Riya Paragkumar Desai has completed her Bachelor of Technology in Electronics and Communication Engineering from Manipal Academy of Higher Education, Dubai, United Arab Emirates. She has completed a couple of projects in the field of Internet of Things during the course of her undergrad. She was successfully able to publish her paper on “Mobile Application Integrated Intelligent Energy Audit System for Smart Home”, during the final year of her undergraduate degree. Later, she went on to pursue her Masters of Science in Electrical and Computer Engineering from the University of Florida,FL. She majored in Signals and Systems. She worked as a Research Assistant for a year, during which, she worked under the guidance of Professor Eisenstadt on a CDC funded project. She worked on making a droplet imager using image processing algorithm and IoT hardware setup to test insecticides. She successfully graduated with her Masters of Science degree.