Forside

Eksamensinformation

Specialization Project in Computer Science

Besvarelse afleveres af

Botond Zoltán Horváth
stud-botond@ruc.dk

Magnus Hartø Trojahn
hartoe@ruc.dk

Maurits Villas Jeppesen Puggaard
mvjp@ruc.dk

Martin Jarner Vasegaard Jørgensen
mjvj@ruc.dk

Marcus Rønnow Stjernø
marost@ruc.dk

Eksamensadministratorer

IMT Tilmelding & Eksamen

imt-eksamen@ruc.dk

Bedømmere

Maja Hanne Kirkeby

Eksaminator

majaht@ruc.dk

☎ 4674-2515

Henrik Bulskov

Intern medbedømmer

bulskov@ruc.dk

☎ 3169-5589

Besvarelsesinformationer

Titel: Design, implementation and analysis of an cost-effective pollen trap

Titel, engelsk: Design, implementation and analysis of an cost-effective trap

Antal tegn: 62723

Antal normalsider: 48

Afleveringsdato: 04-06-2024

Tro og love-erklæring: Ja

Indeholder besvarelsen fortroligt materiale: Nej

Roskilde Universitet

Design, implementation and analysis of an cost-effective pollen trap

Botond Zoltán Horváth, 79368, stud-botond@ruc.dk
Magnus Hartø Trojahn, 71389, hartoe@ruc.dk
Maurits Villas Jeppesen Puggaard, 71425, mvjp@ruc.dk
Martin Jarner Vasegaard Jørgensen, 71801, mjvj@ruc.dk
Marcus Rønnow Stjernø, 71370, marost@ruc.dk

Examiner: Maja Hanne Kirkeby, majaht@ruc.dk

Second examiner: Henrik Bulskov, bulskov@ruc.dk

GitHub Repository with scripts used in the project:
https://github.com/Semester-2-Master/Image-Capture
https://github.com/Semester-2-Master/Backend

With special acknowledgements to **Bo Thorning** for assistance with prototype fabrication and **Ulf Rørbæk Pedersen** for help with mathematical formula.

Abstract

This project develops a pollen trap designed to collect and image pollen, allowing for detection and identification. The goal is to make local and real time pollen counts available to ordinary consumers and pollen allergists. This approach contrasts with the current centralized system in Denmark, which provides pollen counts from only two locations. By offering real-time, localized pollen data, the project aims to demonstrate the potential value of such information for everyday consumers.

Table of content

Abstract..........3
1 Introduction..........5
1.1 Background..........6
1.2 Motivation..........6
1.3 Problem formulation..........7
2 Requirements..........7
2.1 Validation requirements..........8
2.2 Limitations..........9
3 Design and Analysis..........10
3.1 Particle sensor vs Image Recognition..........10
3.2 Dark- and bright- field microscopy..........11
3.3 Polen intake systems..........13
3.4 Local vs Cloud Storage..........15
3.5 Local Processing vs Cloud Processing..........16
3.6 Sampling Surface..........18
3.7 Pollen Processing..........20
4 Implementation & Collector Design..........23
4.1 Tape and mechanisms..........23
4.2 The box..........24
4.3 Microscope lights..........25
4.4 Microscope lens implementation..........25
4.5 Airflow and collection tube..........28
4.6 Calculating a pollen count using our box design..........30
4.7 Availability and cost effectiveness..........31
4.8 Software Implementation..........32
5 Evaluation & Test..........37
5.1 Evaluating Feasibility of Pollen Trap..........39
5.2 Comparison to State of the Art (Astma-Allergi Danmark)..........45
6 Conclusion and future work..........46
7 References..........47

1 Introduction

Pollen, the male reproductive powder of flowering plants, plays a crucial role in the natural world. Carried by wind and insects, pollen grains travel vast distances to fertilize other plants, ensuring the continuation of countless species. However, for many people, pollen is not just a symbol of plant life, it is a trigger for seasonal allergies (*Pollen Allergies*, n.d.).

The microscopic pollen grains contain proteins that can trigger an immune response in some individuals. When inhaled, the body mistakenly identifies these proteins as a threat, leading to symptoms like runny nose, itchy eyes, and difficulty breathing. Understanding pollen types and their abundance becomes crucial for those managing allergies (*What Is Pollen? | Northeast Allergy*, 2023).

Although pollen counts are often reported in weather forecasts, they are usually 24-hours delayed and/or only collected in very limited locations - but the weather forecasts and apps allow people with allergies to plan outdoor activities for low pollen days (*What Is Pollen Count, and How Does It Change?*, n.d.). Understanding pollen counts helps people manage their allergies, but is also very important in agricultural studies and for understanding changes in biodiversity. (*An integrative environmental pollen diversity assessment and its importance for the Sustainable Development Goals, 2022*)

Pollen counts are normally conducted manually by state-funded bio labs and organizations - this approach requires specially trained individuals, and is often a slow and time consuming process. Our aim is to make pollen counts more available and reliable for everyday citizens - the project aims to complete this goal by developing a pollen monitoring machine.

1.1 Background

Building upon prior bachelor work (*Bachelor (2023)*), this continuation project aims to delve deeper into the integration of pollen traps for more accurate pollen analysis. We want to investigate the possibilities for pollen traps, and try to make an affordable and accessible version that can give a better representation of pollen in Denmark, than the one we have now.

During the previously mentioned bachelor project, we realized that it is possible to automate the pollen count process. Our goal now is to make it feasible to deploy multiple pollen traps across the country without requiring specialized attendance.

As students of physical computing, we are part of an academic program that emphasizes the intersection of hardware and software to create innovative solutions. With our collaboration with Fablab, we get the opportunity to create prototypes that replicate our visoned solution as closely as possible.

1.2 Motivation

Here in Denmark, Astma-Allergi Danmark (AAD) plays a vital role in pollen monitoring. They collect pollen samples from two stations in Viborg and Copenhagen. However, the current system has limitations:

- **Limited Availability:** With only two stations located in Copenhagen and Viborg, the collected data provides a limited snapshot of pollen distribution across Denmark. This can lead to inaccurate forecasts for areas far from monitoring points.
- **Data Delays:** The traditional methods used by AAD involve manual analysis of collected samples, leading to delays in data updates. The current system only shows the pollen numbers from the last 24 hours, and not the current pollen numbers for the day. They do forecast the next five days in advance based on an estimate.
- **Cost Constraints:** Sophisticated equipment and expertise are often required for traditional pollen collection, making it potentially expensive. This limitation restricts the number of monitoring stations and hinders a comprehensive understanding of pollen trends.

These limitations highlight the need for more accessible pollen monitoring systems in Denmark. This project aims to create such a system - a pollen trap for local and real time pollen monitoring.

1.3 Problem formulation

The primary research question for this project:

"How can we make pollen monitoring easy and accessible for ordinary consumers and pollen allergy sufferers, enabling them to determine the pollen count without compromising on accuracy?"

To answer this research question we will develop a pollen trap that will allow us to measure and analyze the pollen count from a specific area. The trap will be used to create a greater understanding of how to make pollen counts accessible and accurate.

2 Requirements

In this section, we will discuss the key requirements that have influenced the development of this project. These requirements stem from various sources, including external benchmarks and our own demands. Most importantly, we aim to align our requirements with those of Astma Allergi Danmark to ensure high standards and relevance. Additionally, we drew inspiration from the Pollen Sense product, which significantly influenced the physical requirements for this project. Our goal is to integrate the best practices and innovations from these sources to develop a robust and effective solution. We will also mention the limitations we have met during this project. These requirements will be the basis for the evaluation of the project.

2.1 Validation requirements

Aspect	Minimum requirement
Photo	We want the product to be able to take clear images of the pollen input. The images should be clear enough to train a machine learning model on it.
Calibration	The microscope should be able to calibrate the distance to the sample for optimal image capturing. When the calibration is done the microscope should be stable and locked in place, so that it doesn't move.
Portable	The product should be able to be placed anywhere, regardless of the environment. We want the product to be as low maintenance as possible. The product should also be able to withstand all kinds of weather, and should also be water resistant.
Intake	The intake needs to collect pollen particles. The amount of air processed should be able to be used to accurately calculate pollen counts. This means we need to process at least 10 liters per minute (Lampugnani & Ebert, 2018).
Pollen count	To validate the product, we want to compare the pollen count we get, with the pollen count Astma Allergi Danmark delivers. We need to develop a method for calculating the pollen count based on the samples we collect.
Monitoring period	The product should give an output **at least** once every day, so that it is possible to replicate the daily pollen count. We want to be able to monitor pollen all year around, but most importantly when the pollen season is active.
Surface examined	We aim to examine the largest surface area as possible, however, if we can not figure out how to concentrate the sample we can take a smaller area and give an estimate of the whole surface. We want to look at a minimum of 10 % of the sample collected. Which is a method already in practice (Lampugnani & Ebert, 2018).
Usability	The product should be able to run with minimum human interaction. The results should be easy to retrieve, and the user shouldn't need specialized knowledge to handle the product.
Accessibility / Cost-effectiveness	We want the solutions we pick to be affordable, and available for the common user. The goal is to make a product that is comparably cheaper than other pollen monitoring systems.
Scalable and clean	The code should be scalable to multiple units, as well as expandable and easy to read.

2.2 Limitations

This section outlines the key limitations encountered during the development. These limitations mainly involve the scope of the Machine Learning integration, the capabilities of the chosen microcontroller (ESP32).

2.2.1 Machine Learning and scope limitation

ML will be mentioned and though it is a big part of making our product work, we have deliberately limited the machine learning part of this project, as it is not our focus point. We want to make a machine that can deliver data that is ready for machine learning. We will be running smaller machine learning models on our data, to prove the feasibility of detection, but will *not* be training and retraining models for classification. Limiting the machine learning scope means that a machine learning pipeline and retraining system would be out-of-scope for the project, but the smaller tests still allow for feasibility and proof of concept testing.

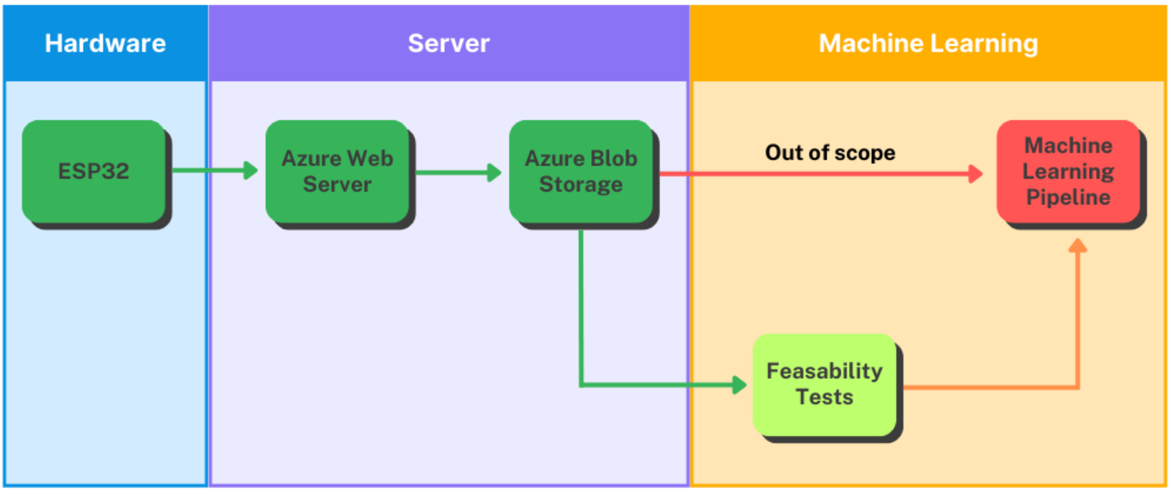

fig 1, *illustration of the project scope*

2.2.2 ESP32

We chose the ESP32 as our microcontroller because it functions effectively as an embedded system. However, it also has some limitations. For instance, the ESP32 has a limited number of pins, so we need to be cautious in how we utilize them. Additionally, while the camera module we use with the ESP32 meets our functional requirements, its image quality is not very high.

3 Design and Analysis

In this section we will explain the different approaches we took to designing the aspects of the project. We analyze different options and explain benefits and disadvantages, and reasons behind our choices.

3.1 Particle sensor vs Image Recognition

This section explains the two measurement methods we have discussed and tested. The measurement solution is crucial to getting results that are usable, and there are two completely different solutions we have explored.

3.1.1 Particle Sensor

A particle sensor filters air through a tiny laser, which then counts the amount of VOC's (Volatile Organic Compounds) in the air. This process offers a highly sensitive and accurate method for measuring VOC levels. The tiny laser beam acts as a precise probe, allowing for the detection of even minute amounts of these compounds.

Pros	Cons
Price: Cheap in materials and time. **Difficulty:** Easy to put together. **Wattage:** Low power usage.	**Range:** Not effective enough to pick up pollen particles. **Specificity:** The sensor is not specific to pollen, triggering on all other airborne particles.

3.1.2 Image Recognition

Image recognition is the act of running a machine learning model on pictures to detect and classify specific objects on the pictures.

Pros	Cons
Specificity: Image recognition readily distinguishes pollen grains from other airborne particles. **Precise:** Images taken of pollen particles are much easier to calculate into pollen numbers.	**Dependent:** Needs custom pollen particle capturing system. **Adjustments:** The product needs to be extremely precise, for the images to be clear enough for a machine learning model.

Prestudy

During testing we discovered that while the particle sensor provided data , its limitations in specificity hinder its ability to accurately measure and identify pollen. The trap cannot distinguish pollen from other airborne particles. Additionally, relying solely on particle size for pollen classification is unreliable due to the overlap in sizes between different pollen types. Based on previous experience from our bachelors project, we knew that pollen identification was possible with image recognition.

3.1.3 Conclusion

Recognizing the limitations of the particle sensor, we decided to pursue the image recognition approach. The main goal of the project is not to create a machine learning model, but to capture pictures of pollen particles, which is essentially creating a system ready for image recognition.

3.2 Dark- and bright- field microscopy

When working with microscopes there are two main methods of viewing the samples. Darkfield- and brightfield -microscopy are contrast-enhancement techniques used in light microscopy that allows us to see samples *(What Is Darkfield and Brightfield Microscopy?, 2020)*.

3.2.1 Brightfield

Using brightfield microscopy to illuminate pollen samples allows for observations of color, as the light is reflected directly through the subjects. This method requires very precise light control, and is often used in modern microscopes, in conjunction with some sort of oil to stain the samples.

Pros	Cons
Colors: Suitable for observing colors of specimens. **Dense areas:** Good for visualizing contrast in dense areas of the sample.	**Contrast:** Low contrast for many biological samples, leading to washed-out images. **Calibration Sensitive:** Very precise light calibration needed for optimal pictures.

3.2.2 Darkfield Microscopy

In darkfield microscopy, direct light is blocked from reaching the objective lens. This creates a dark background, while light scattered by the sample appears bright. This technique is useful for visualizing transparent or unstained samples with well-defined edges.

Pros	Cons
Transparent sample: Excellent for visualizing transparent or low contrast samples. (such as pollen particles.) **Contrast:** Creates high contrast images with bright objects in a dark background. **Setup:** Easy to set up as light needs to hit subjects and not microscope lens directly.	**Dim:** Lower overall light levels in the final image due to blocking direct light. **Noise Sensitive:** Requires careful handling of samples as dust particles can also appear bright.

Prestudy

In the initial testing of the lighting solution for our project, we discovered that light placement was directly correlative with either dark- or brightfield microscopy. Placing a light source so that it was shining directly into the microscope would result in brightfield microscopy, allowing for clearer pictures with colors intact. Doing this caused issues with the tape, since we could see the glue with this lighting, it also required microscope levels of precision to work properly. Instead, when placing the light on an angle, so that we illuminated the tape from the bottom, we achieved darkfield microscopy. This method was much more lenient with placement, and resulted in pictures that were usable as only subjects would be shown, and not the tape.

3.2.3 Conclusion

Using brightfield microscopy would allow for clearer, more distinct pictures of pollen particles, with their natural color intact - but the difficulty in setup takes away from the results and makes getting good pictures difficult. Darkfield has a much larger margin of error, and allows for quicker setup with pictures good enough for use later. Having a much larger margin for error, increases the ease of use of the machine, and aligns with our availability requirement as the machine, with darkfield, requires less setup and maintenance.

3.3 Polen intake systems

An essential aspect of pollen collection is the method of transferring pollen particles from the air into the imaging system. This component involves understanding fluid dynamics, air movement, and related physics. Given our focus on physical computing, we explored various solutions from other actors.

3.3.1 Pollen Sense

The intake system of Pollen Sense is proprietary, involving a slit/hole and a suction system.

fig 2: *Pollen sense sensor* (*Pollen Sense, n.d.*)

Pros	Cons
Hypothetically Simple: Implementing a slit and suction in a box is straightforward.	**Unknown Variables:** Specifics are confidential, potentially hiding crucial aspects like achieving laminar flow, which is mentioned on their website. **Closed Source:** Limited visual references hinder replication.

3.3.2 Cyclone

Olga Saukh and Nam Cao developed a 360-degree cyclone design with open-source schematics that, when attached to a 150-liter-per-minute blower, filters air, letting only pollen pass.

Pros	Cons
Open Source: Detailed 3D print schematics and scientific data are available saving implementation time. **Efficient Filter:** Potentially filters out pollutants, allowing mostly pollen particles through.	**Scientific Complexity:** The filtration's physics are complex and distant from our expertise, potentially leading to usage challenges.

3.3.3 Astma-Allergi Inspired

Asthma-Allergi Denmark employs a pivoting vacuum-like system that deposits pollen onto microscope glass plates, which are manually collected and replaced.

Pros	Cons
Standardized: Aligns with industry practices, helping us make valid comparisons.	**Mechanical Complexity:** Building a similar system requires intricate, time-consuming mechanisms. **Manual:** Requires users to collect the plates.

Prestudy

Our first approach was the cyclone due to its ease of use. This choice minimizes development time, allowing us to focus more on other aspects of the project effectively. However, during system testing, we could not replicate the results from Nam Cao et al.'s automated pollen detection system (N. Cao, M. Meyer, L. Thiele, O. Saukh, (2020) Automated Pollen Detection with an Affordable Technology.). When replicating their test, we found that we could not replicate their results, as we were not able to use its filtering properties.

3.3.4 Conclusion

Based on our research and testing, we decided that using the Pollen Sense solution was the most optimal choice for our project. Although we initially intended to implement filtering through the cyclone, our testing under the specified environment did not replicate the filtering results of the Austrian project. (Nam et al. (2020) Achieving results without the cyclone means we will need to filter out other particles in post-processing.

3.4 Local vs Cloud Storage.

This section outlines the development dilemma faced while choosing a storage solution for the pollen collector prototype. Implementing local (SD card) or cloud-storage requires distinct system architectures, and selecting one is important.

3.4.1 Local Storage (SD Card)

The ESP32 has an SD card reader/writer, it can store images taken from its camera module locally.

Pros	Cons
Offline Capability: Saves images directly without Wi-Fi, enabling deployment in remote locations (provided a manual data collection method exists). **Simple Hardware Integration:** Utilizes existing SD slot and readily available Arduino libraries for data storage.	**Manual Data Transfer:** Requires physical access to the collector for data retrieval, introducing user error and potentially reducing usability. **Storage Limitations:** SD cards have limited capacity compared to cloud storage.

3.4.2 Cloud Storage

Sending data to a cloud storage would make all data centralized, and could be made using developer friendly cloud services.

Pros	Cons
Automated Data Transfer: Leverages Wi-Fi for automatic data transfer, minimizing user involvement and error. **Centralized Processing:** Enables centralized data analysis, enabling deeper insights across geographically dispersed collectors. This also simplifies hypothetical model retraining and enables collaboration between collectors.	**Receiver Complexity:** Requires a purpose-built receiver capable of interpreting the specific data format generated by the ESP32 hardware. This adds complexity to the processing system. **Internet Reliant:** Data transfer cannot be completed without the internet, meaning disruptions could interfere with following systems.

Prestudy

During testing we first implemented direct transfer from the ESP to another device, quickly realizing connection issues were a problem, with no on-board storage. We then implemented SD-card storage onto the ESP for further testing, but then realized that our choice of ESP limited us to a set amount of pins, and that these pins were needed for running both the camera module and the servo for moving the tape. We therefore decided to do cloud storage, as it allowed us to ease maintenance as well as free up pins for the servo.

3.4.3 Conclusion

We opted for cloud storage due to it allowing for large-scale pollen collection initiatives, which aligns with our goal of improving upon Astma Allergi Denmark's methods. However, local storage offers an advantage for areas lacking internet access, which aligns with the portability requirement. With further resources it is worth investigating a hybrid solution that uses both local storage (for offline collection) and cloud storage (for centralized analysis). Storing the data in the cloud allows for ease of use, removing the physical aspect of having to manually go and retrieve the sd card from the box. This aligns with our usability requirement, as well as our availability requirement.

3.5 Local Processing vs Cloud Processing

We have considered the possibilities of processing images locally on the device or on cloud servers. Our choice of image processing highly depends on available hardware and storage solutions.

3.5.1 Local Processing

Local processing involves handling all data directly in the pollen trap. The ESP32 microcontroller would be responsible for capturing microscopic images of pollen, processing these images to detect and identify pollen types, and then storing or displaying the results via its own web server.

Pros	Cons
Reduced Dependence on Internet: Local processing can function in areas with poor or no internet connectivity, ensuring consistent operation. **Standalone operation:** There is no need for a remote server, making it more cost effective and easier to maintain. **Privacy:** Data remains local, reducing concerns about data security.	**Limited Compute Power:** The ESP32 microcontroller has limited processing capabilities, which is not sufficient for complex image processing and pollen identification algorithms. **Power Consumption:** Intensive processing tasks increase power consumption, which would be disadvantageous if we wanted to power the device via a battery. **Storage Constraints:** Limited storage capacity on the device for storing large datasets or machine learning.

3.5.2 Cloud Processing

Cloud processing involves uploading the captured microscopic images to a remote server where the processing, detection, and identification of pollen particles are performed. The results are then made accessible to users via the internet.

Pros	Cons
Scalability: Cloud servers can handle large volumes of data and complex computations, ensuring high accuracy and efficiency in pollen detection and identification. **Storage Capacity:** Cloud infrastructure provides virtually unlimited storage, allowing for extensive data archiving and analysis. **Continuous Updates:** Algorithms and software can be updated continuously on the server without needing to modify the device firmware, ensuring the system stays current with the latest advancements in pollen detection.	**Internet Dependence:** Requires a stable internet connection to upload images, which may not be reliable in all locations. **Continuous cost of operation:** Cloud storage and servers have a monthly cost, which makes it more expensive to operate the device long-term. **Data Security and Privacy:** Transmitting data over the internet introduces potential security risks and privacy concerns.

Prestudy

When discussing processing we had already decided on using cloud storage, this heavily directed us toward also having processing being in the cloud, as these systems could be integrated together.

3.5.3 Conclusion

After evaluating the options, we decided to opt for cloud processing. The ESP32 microcontroller lacks the necessary computational power to efficiently handle the complex image processing tasks required for accurate pollen identification. Therefore our device uploads pictures to a cloud storage, where cloud servers would be able to run advanced algorithms to perform these tasks efficiently. Cloud processing also allows for continuous updates and improvements to the pollen detection algorithms, ensuring that the system remains effective and up-to-date without installing updates locally on the device. These aspects play into the usability requirement.

3.6 Sampling Surface

We tested multiple surfaces, to see which would yield the best results when combined with our ESP32 and microscope. The 3 main tests we did were on office tape, and glass and each of these presented their own pros and cons:

3.6.1 Office Tape

Pros	Cons
Disposable: Readily available and inexpensive. **Simplicity:** Easy to use and collect samples.	**Background Noise:** Microscoping tape with glue can cause background interference in images. **Sample Inconsistency:** Pollen may stick unevenly, affecting image clarity. **Non-reusable:** Cannot be cleaned and reused.

3.6.2 Glass plate

Pros	Cons
Optimal Collection: Smooth surface for pollen collection. **Optimal Imaging Surface:** Offers a clear platform for image capture, also state of the art for microscopes.	**Fragile:** Brittle and prone to breakage. **Maintenance:** Needs to be cleaned to be reused. **Cost:** More expensive. **Difficulty:** Moving glass would result in a more complex mechanism.

3.6.3 Movable glass plate

Pros	Cons
Continuous Collection: A system with multiple movable plates could theoretically enable continuous collection, with one plate capturing pollen while another is being analyzed. (This can also be achieved using tape, which is much cheaper and easier to work with.)	**Increased Complexity:** Designing a mechanism to move multiple plates, like a "caterpillar-wheel" or a robotic arm, would significantly add to the project's complexity. **Calibration Challenges:** Ensuring precise and consistent movement of the plates would be crucial for both focusing and maintaining a consistent distance between the pollen and the microscope lens.

3.6.4 Conclusion

During prototyping we conceptualized multiple ways to build a mechanism that would move multiple glass plates, so that we could collect and capture at the same time. Ultimately, we decided to do collection on tape, based on the accessibility and simplicity of it - moving glass plates would require mechanisms that would take away from the maintainability of the machine - this decision supports our availability and useability requirements.

3.7 Pollen Processing

We process a certain amount of air, but a microscope can only capture a small section of the surface we collect on. Therefore, we need to strategize how we observe the air we've processed. We considered three approaches: a mechanical system that moves the microscope to take pictures of the entire surface area where pollen lands, a razorblade mechanism that concentrates all pollen into a small area for easier imaging (similar to the Austrian project)(Nam et al. (2020), and mathematical approximation to estimate the pollen count over the entire collection area.

3.7.1 Mechanical Photography System

Pros	Cons
Complete Coverage: Ensures all pollen on the surface is captured in images.	**Mechanical Complexity:** Requires an intricate setup with servos, pulleys, or other mechanisms to move the microscope. **Calibration Challenges:** Moving the microscope could disrupt its calibration and focus, creating a need for an autofocus system.

3.7.2 Mechanical Concentration System

Pros	Cons
Complete Coverage: Ensures all pollen is concentrated and captured in one place.	**Mechanical Complexity:** Similar to the photography system, it requires a complex arrangement of servos and pulleys. **Surface Dependency:** The razor blade must move across a glass surface; tape would be unsuitable as it could be cut.

3.7.3 Approximation

Pros	Cons
Reduced System Complexity: No mechanical adjustments are needed, saving time and simplifying the system.	**Approximate Results:** Provides an estimate rather than a precise count. **Mathematical Dependence:** Requires detailed knowledge of the pollen trap's specifications to accurately calculate the pollen count.

Prestudy

During testing and prototyping, we realized that to give an approximate pollen count, we need to calculate the amount of pollen particles in 1 cubic meter of air. To do this, multiple variables need to be defined.

The first variable is the flow rate - we need to understand how much air we process. During testing we measure our flow rate in meters per second, but we need to convert it into liters per minute to use it on our final equation. We do this by using the following equation:

$$Flow\,rate\ (L/m)\ =\ \Pi\ \bullet\ R^{2}\ \bullet\ 60\ \bullet\ Velocity\ \bullet\ 1000$$

R *is radius of intake in m*

Velocity *is meters per second measured at intake.*

Besides the flow rate, we also need to know the precise size of both our intake- and microscope examination-area.

The field of view calculation we need for our pollen count looks like this, where the **AreaMicroscope** is the microscope examination area, and the **AreaIntake** is the size of the intake system, where the pollen is concentrated on the tape.

$$P\ =\frac{AreaMicroscope}{AreaIntake}$$

With the equation for P set up, as well as the flow rate, we can decide on the rest of the values for use in the approximation solution, and solve for **C** - Concentration:

N = *Numbers of observed pollen particles*
P = *Area in percent number (1% = 0.01)*
F = *Flow rate in m3 per hour*
T = *Time in minutes*

$$C\ =\ \frac{N}{F\ \bullet T \bullet P}$$

3.7.4 Conclusion

Due to the complexity of the mechanical systems, we chose to use the approximation method. Although this reduces the reliability of our pollen counts, pollen counting is generally an approximation process. The key is to ensure statistically significant sample sizes for fair approximations, which in this formula is a long enough sampling time or area examined.

4 Implementation & Collector Design

In this section, we will explain how we implemented the approach we took to make the pollen trap. We will showcase the different mechanisms, microscope, chips and circuits with illustrations.

4.1 Tape and mechanisms

To collect the pollen to be analyzed we use officetape (1), where the pollen falls on the sticky side of the tape (2). The tape rolls on a custom 3D-printed mount (3) that lies on two screws, where the purpose of the mount is to hold a glass plate (4) that lies under the tape, to stop the airflow from pushing down on the tape, and ruin the microscope calibration. To make fine adjustments to the length of the microscope, there is a 3D-printed adjustable mount (5) for the glass plate that can be screwed up and down to move the glass plate. This is the calibration for the microscope, assuring images can stay in focus. The tape then runs to a 3D-printed mount (6), so the tape's flow can be controlled via a servo motor on the backside of the plate.

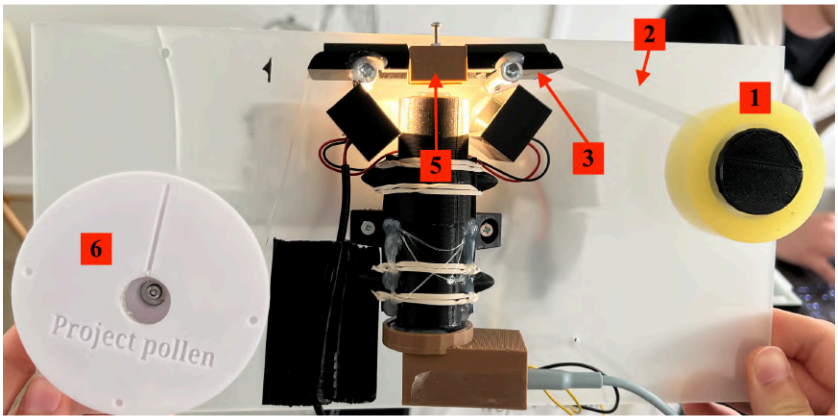

***fig 3**, Front view of the pollen trap mechanisms*

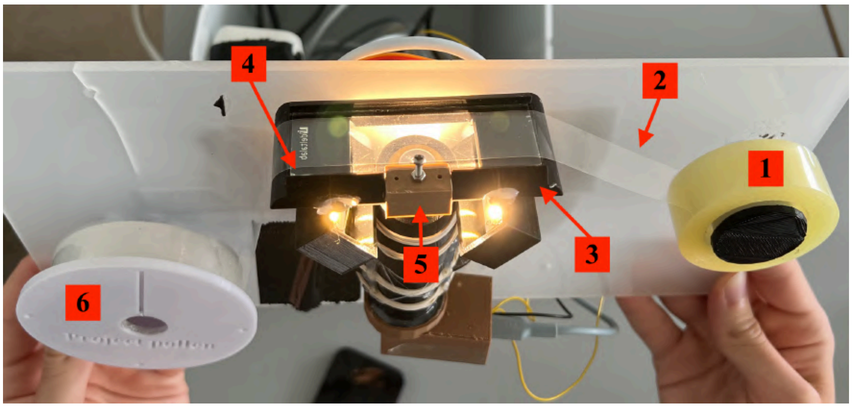

***fig 4**, Picture from the top of the pollen trap mechanisms*

4.2 The box

To contain the mechanisms of the microscope and to make the pollen collection possible, there are some requirements for the box that need to be met. The box is made from acrylic and is taped with duct tape on the sides to allow for darkfield microscopy on the inside. To collect the pollen, we need the box to be tight so a vacuum can be made by the fan (5). On the top of the box we have the air intake (1), and on the bottom of the box the airflow can come out with a cut out hole (2). To have enough power for the ESP32 /ESP32CAM, fan and lights, we have installed an extension cord (3) inside of the box. We also have a controller for the fan (4) in the box so we can control the airflow of the fan. The box needs a wifi connection and a power source for setup.

fig 5, outside picture of pollen trap box

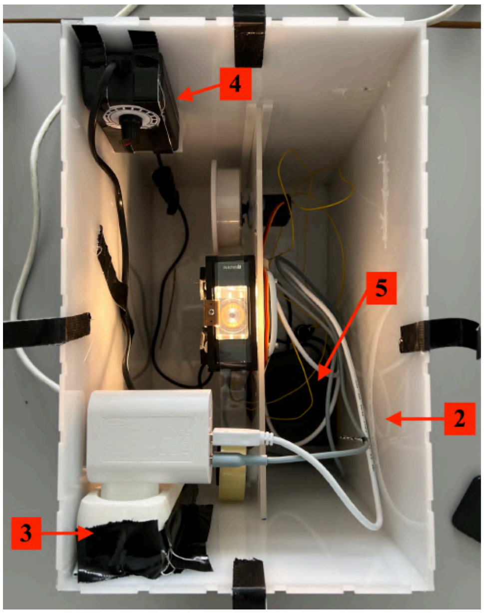

fig 6, picture of inside of the pollen trap box

4.3 Microscope lights

To get good results from the camera there needs to be the right amount of lighting for the microscope. To do this we installed two 3D-printed boxes (1), each of them containing three white 3V LEDs and a 100 Ω current limiting resistor (3) connected in series. The two boxes are connected in parallel to a 12V power supply.

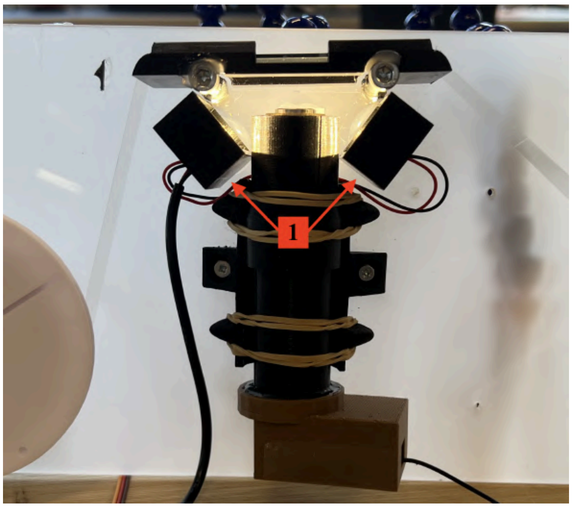

***fig 7**, Front view of t of the microscope lights*

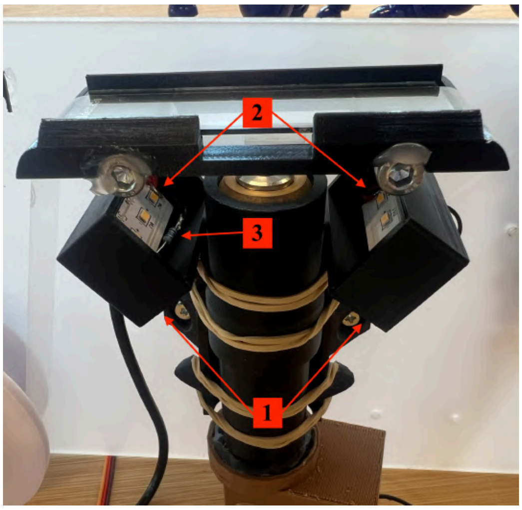

***fig 8**, Microscope light from above*

4.4 Microscope lens implementation

For the microscope we use a 10x lens (1) to get most optimal results, based on the distance we take the pictures from. The lens is inserted in a 3D printed tube (2) that connects a 3D-printed box containing an ESP32CAM that takes the pictures (3). The tube containing the lens is strapped to a custom 3D-printed mount (4) with elastic bands that makes it stay in place and easily adjustable during development. After we found the correct distance for the lens, we hot glued the tube to the holder to fix it in place. There is also a focusing screw for fine adjustment which lifts the glass plate in its holder.

In our concentration equation, we need to define intake and collection area sizes, and to do this, we need to understand the field of view we get through the microscope.

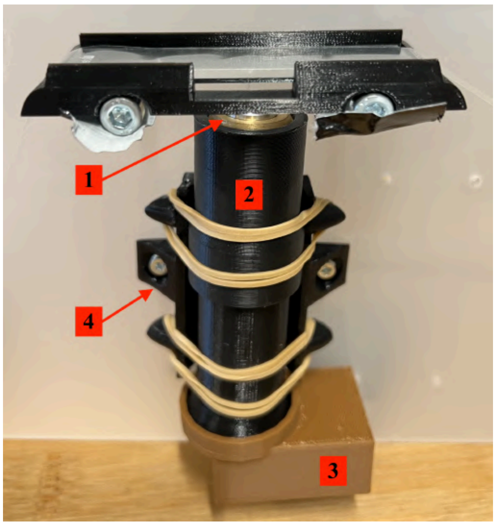

***fig 9**, Front view of the microscope mount*

We can calculate the relationship between intake area and microscope field of view using the equations explained in section 3.7.3, and here we want to insert our measured values.

$$P = \frac{AreaMicroscope}{AreaIntake}$$

To do that we first need to calculate the area we see through the microscope, compared to the total area we collect pollen on.
To calculate the area of intake and microscope, we set up the following formulas, to first calculate our intake area:

Area of circle formula:

$$\Pi * R^2$$

The funnel end has a diameter of 1.5cm, and we assume that the collection area is the same and that all pollen collected sticks to the tape, therefore we can calculate the final value of our AreaIntake as:

$$\Pi * 0.75^2 = 1.767 cm^2$$

We assume that a dandelion pollen is, on average, 35 micrometers in diameter (*Dandelion - Taraxacum*, n.d.). We calculate the amount of pollen particles by dividing the size of a pollen particle (48 px) by the height (1200 px) and width (1600 px). Therefore, calculating the amount of pollen grains fitting on both the x and y axis on our photo means that the width of our image is 33,33 dandelion particles (the amount that fits on an image)* *35um* and the height is 25 dandelion particles * 35 um.

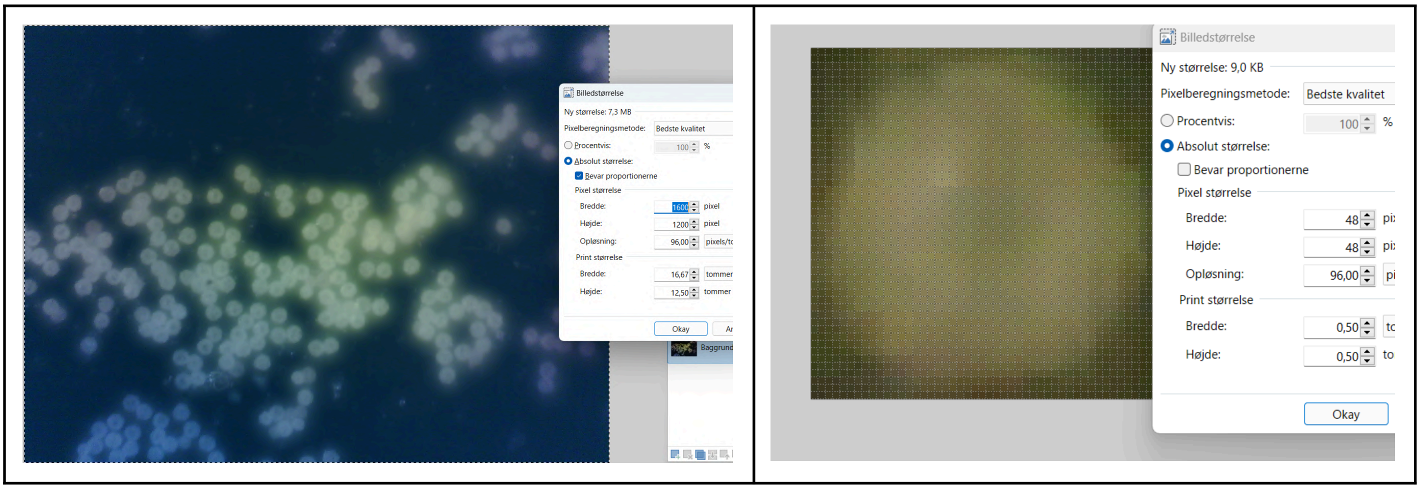

Fig 10 / **Fig 11**. *Measuring field of view by calculating size of image by pollen size*

Width = $35\upsilon m \bullet 33,33 = 1166\upsilon m$

Height = $35um \bullet 25 = 875um$

Seeing as our image is a rectangle, and we have our assumed specifications, we can then calculate this into centimeters, by multiplying by 1000 (as 1 mm is equal to 1000 microns).

$$1166um \bullet 1000 = 0.1166cm$$

$$875um \bullet 1000 = 0.0875cm$$

Calculating the final area of our AreaMicroscope, can then be done with the formula for the area of a rectangle:

$$A = L \bullet W$$

$$AreaMicroscope = 0.1166cm \bullet 0.0875cm = 0.0102\ cm^2$$

The field of view is the portion of the complete collection area that we photograph.
We define FOV as P, and calculate it as follows, using the aforementioned equation for p:

$$P = \frac{0.0102cm^2}{1.767cm^2} \bullet 100 = 0.57\%$$

The value P will be used to calculate the pollen counts, as a ratio for what our image represents of the total collected pollen. We realize that a coverage percentage of 0.57% is lower than our requirements, and plan to compensate for this with multiple factors, e.g the amount of air we process.

4.5 Airflow and collection tube

To collect the pollen we use a fan (1) that creates a vacuum for the pollen to be sucked in (*Editbar Centrifugal Fan 12 V, 220 V*). The pollen particles then fall down a 3D-printed funnel (2) that directs the pollen to the tape. The air is blown out of the bottom of the box to create the vacuum effect (3). Furthermore to make the box *partially* water resistant we have designed pillars with velcro (4) as well as a roof (5), which are placed over the funnel, to protect it from rain.

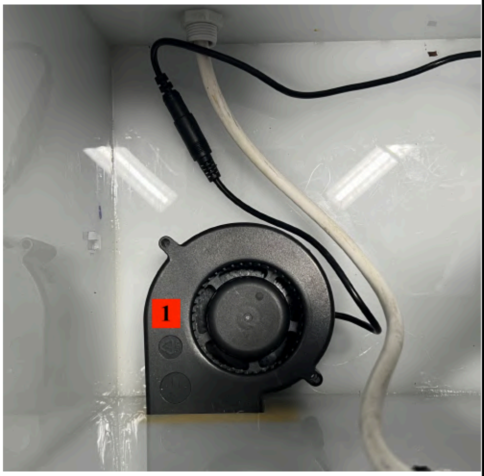

fig 12, *Picture of the fan inside of the pollen trap box*

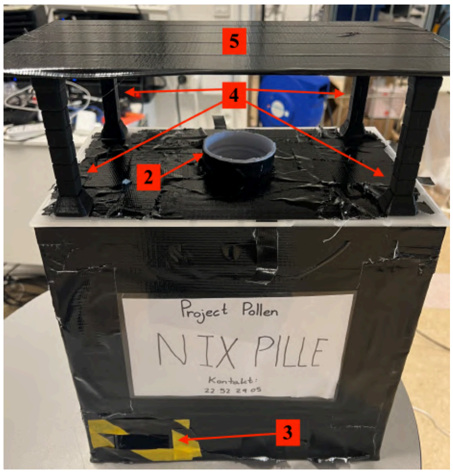

fig 13, *picture of the poles and roof of the pollen trap box*

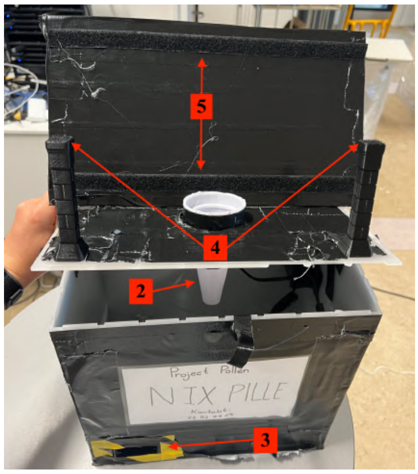

fig 14, *picture of the poles and roof of the pollen trap box*

To calculate the airflow our fan can deliver, we use the equation in 3.3 Polen intake systems, with R being the radius of our funnel entrance in meters and Velocity being the wind speed measured by the entrance of the funnel:

$$Flow\ rate\ (L/m) = \Pi \bullet R^2 \bullet 60 \bullet Velocity \bullet 1000$$

$$R = 0.0235\ \mathrm{m}$$

$$Velocity = 1.4\ meters\ per\ second$$

We then sub our values in:

$$Flow\ rate(L/min) = \Pi \bullet (0.0235)^2 \bullet 60 \bullet 1.4 \bullet 1000$$

After calculations we end up with a result:

$$\Pi \bullet (0.0235)^2 \bullet 60 \bullet 1.4 \bullet 1000 \approx 145.74\, L/m$$

In our requirement, we wanted a flow rate of a minimum of 10 L/m, but the higher airflow we get, the more air we can measure and give a more accurate pollen count per cubic meter. The result we get of 145.74 L/m is better than what we expected, and gives us room for adjustments, when we calculate how much air we want to process, due to our fan being adjustable.

We can then convert this into cubic meters per minute by dividing by 1000, as there are 1000 liters of air in a cubic meter:

$$F = 145.74 / 1000 = 0.145\, m^3/min$$

We realize that our flow rate is significantly higher than our requirements and will leverage this to compensate for our FOV being lower than our goal.

4.6 Calculating a pollen count using our box design

With the calculation we have done earlier in the 4.5 Airflow and collection tube and the 4.4 Microscope lens implementation segments, we can now make a calculation that estimates the pollen count concentration in m^3/h.

P = Area in percent point
F = Flow rate in m3 per hour
T = Time in minutes
N = Numbers of observed pollen particles

$$C = \frac{N}{F \bullet T \bullet P}$$

We insert our values, in this hypothetical example we have observed 5 pollen particles:

$$C = \frac{5}{0.145m^3 \bullet 360 \bullet 0.0057} = \frac{5}{0{,}3} = 16.80\ \frac{N}{m^3}$$

This means that during collection (6 hours/360 min) the average cubic meter of air would have had a pollen count of 16.

4.7 Availability and cost effectiveness

To ensure cost effectiveness and project transparency, a table of materials and price is created.

Product	Price	Link
Mat acrylic 3 mm	89 kr.	https://shorturl.at/bZN4M
Servo motor	55,76 kr.	https://shorturl.at/xkaTp
Office tape	16,19 kr.	https://shorturl.at/15hOT
Fan	203,65 kr.	https://shorturl.at/agVNJ
Microscope	137,62 kr.	https://shorturl.at/m5gDA
Esp 32 & camera module	149 kr.	https://shorturl.at/swlLD
3d print fillment	160 kr.	https://shorturl.at/qkHGD
Duct Tape	100 kr.	https://shorturl.at/a5dpa
Glass plate	20 kr.	https://shorturl.at/yBrLQ
TOTAL	**931,22 kr.**	
Azure hosting	**App Service Basic Plan:** Approximately $50 per month. **Blob Storage:** Pay-as-you-go plan at $0.018 per GB, with 7.692 images per GB.	https://shorturl.at/v1ryA

The affordability of the trap removes a significant barrier of entry, potentially allowing individuals to do pollen collection in their own home. Importantly, the project also facilitates the possibility of local pollen counting. These factors support both our availability and useability requirements.

4.8 Software Implementation

This section provides a detailed examination of all the software systems developed for the project. The core components include:

- **ESP32 firmware** Manages image capture, servo control, and image transfer.
- **ASP.NET Server**: Hosted on an Azure Web App, this server handles incoming data from the ESP32.
- **Azure Blob Storage**: Responsible for storing the captured images.

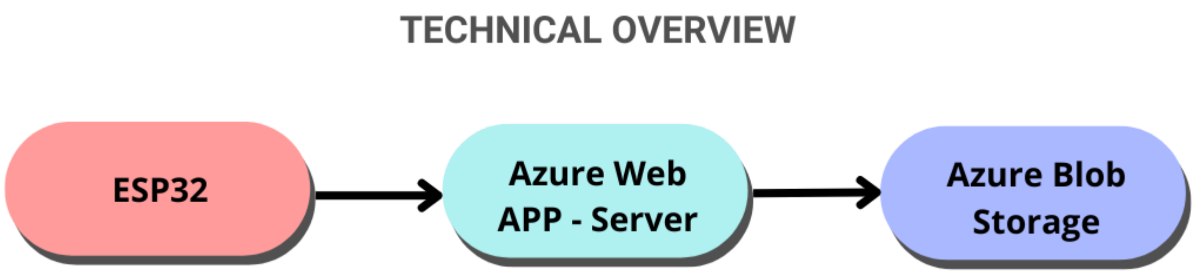

fig 15, *illustration over system components*

Each system will be described in detail, followed by an explanation of their interactions. Following the individual descriptions, we will detail how these components work together to achieve the project's goals.

4.8.1 Pollen Trap Software

The ESP32 system handles image capture, servo control, and image transfer for the project. Developed on the Arduino platform, due to its extensive library support and user-friendly environment. Arduino offers numerous libraries that allow the use of transfer protocols and hardware components such as servos. The integrated development environment (IDE) provided by Arduino simplifies library management, enabling quick downloads and implementation (What Is Arduino?, 2018).

The code follows clean code principles to achieve loose coupling, ensuring ease of maintenance and readability. It separates functionalities into different .cpp and .h files, which are then called by the main program in `Pollen.ino`. Each `.cpp` file contains methods and associated variables or objects relevant to those methods.

The images are assigned filenames based on the current timestamp, ensuring each picture has a unique identifier, which simplifies sorting and organization. For setups involving multiple pollen traps, an additional function is required to provide a unique identifier for each machine, enabling effective sorting and differentiation between the images from different devices.

Libraries used:

- **ServoESP32:** Allows the control of various types of servos.
- **ESP_camera:** Allows interaction with the camera module for image capture.
- **WiFi:** Provides the ESP32 with Wi-Fi connectivity.
- **WiFiClient:** Manages client-side Wi-Fi operations.
- **HTTPClient:** Allows the ESP32 to send HTTP and HTTPS requests over Wi-Fi.
- **Time:** Contains data types for storing system time values.

The ESP32 code uses the following components:

File Name	Description	Functions
Pollen.ino	The main program that orchestrates the execution of other components.	`setup`: Initializes program setup. `loop`: Continuously calls methods from other modules.
wifi_client.cpp **wifi_client.h**	Manages Wi-Fi connectivity.	`init_wifi`: Sets up the Wi-Fi connection.
camera.cpp **camera.h** **camera_pins.h**	Configures the microcontroller pins and camera settings.	`init_camera`: Initializes the camera. `esp_camera_fb_get`: Retrieves the frame buffer data.[1]
hardware_controller.cpp **hardware_controller.h**	Control the servo.	`init_motor`: Attaches the microcontroller pins to the server to send digital signals.

[1] An area of memory used to hold the frame that is captured by the camera (PC MAG, n.d.)

		`motor_step`: Runs the servo for 400ms (configurable). This function is called after a picture is taken in Pollen.ino.
time_config.cpp time_config.h	Manages time synchronization using a Network Time Protocol (NTP) server. Time is used to (uniquely) name images.	`init_time`: Sets up the connection to the NTP server. `get_time`: Fetches the current time.
http_client.cpp http_client.h	Handles HTTP communication to send captured images to a remote server.	`post_picture`: Connects to the server (our Azure Web App), captures an image from the camera, and uploads it with a HTTP POST request. Includes retry logic to handle connection failures.

4.8.2 Server & Storage

The ASP.NET code defines a web API for handling image uploads and processing test requests. It uses Azure Blob Storage for storing uploaded images. The key functionalities are implemented within the `ImageController` class. Our server is hosted on Azure using the Web App Service, chosen for its simplicity and ability to run code via a Docker image without the need to set up a virtual machine.

The following functions are in the ASP.NET Servers `ImageController` class.

Name	Description	Functions/Key Actions
ImageController Constructor	Initializes the ImageController and loads configuration settings.	Uses *ConfigurationBuilder* to build the configuration. We utilize Azure environment variables for secrets when hosted on Azure.
Upload Method	Handles the upload of image files to Azure Blob Storage. This is the primary use of the server.	- Receives an image file via HTTP POST. - Validates the uploaded file. - Retrieves Azure Storage connection string and container name from

		environment variables. - Creates a BlobServiceClient to interact with Azure Blob Storage. - Uploads the file. - Returns the URL of the uploaded blob or an error message if the upload fails.

The deployment process uses a GitHub Action triggered whenever code is pushed to the repository. This action compiles the code into a Docker image and notifies Azure to pull and deploy the new code, ensuring continuous integration and continuous delivery.

GitHub Actions are managed through YAML files, with ours triggered by pushes to the Main branch and comprising two main components: build and deploy. The build process involves checking out the repository code, logging into the GitHub Container Registry, and building and pushing the Docker image to the registry. Afterwards, the deploy job triggers the deployment of the Docker image to an Azure Web App using a webhook, facilitating seamless updates.

4.8.3 Azure Blob Storage

We used Blob storage as it serves as Azure's unstructured data storage service, it handles all CRUD (Create, Read, Update, Delete) operations. These operations can be conveniently managed either through the Azure Dashboard or via the Azure Command Line Interface (CLI). To integrate Blob storage into our application, we acquire the connection string and incorporate it as an environment variable within the App Service, necessary for establishing connections to Blob Storage Objects within the ASP.NET server.

The storage system largely operates seamlessly, allowing easy access to all stored images through the dashboard interface. For more advanced tasks such as bulk image deletion, we use the Azure Command Line Interface. By executing the following command:

```
az storage blob delete-batch --account-name pollenstorage --source
pollencontainer --pattern *
```

Similarly, to retrieve a comprehensive list of stored images, we use the following command:

```
az storage blob list --account-name pollenstorage --container-name
pollencontainer --output table
```

These commands enable efficient management of our image storage within the Azure environment.

4.8.4 System interaction

This section will provide an overview of the interactions between the systems, going from the ESP32 capturing an image, posting it to the server and the server storing it in the blob storage.

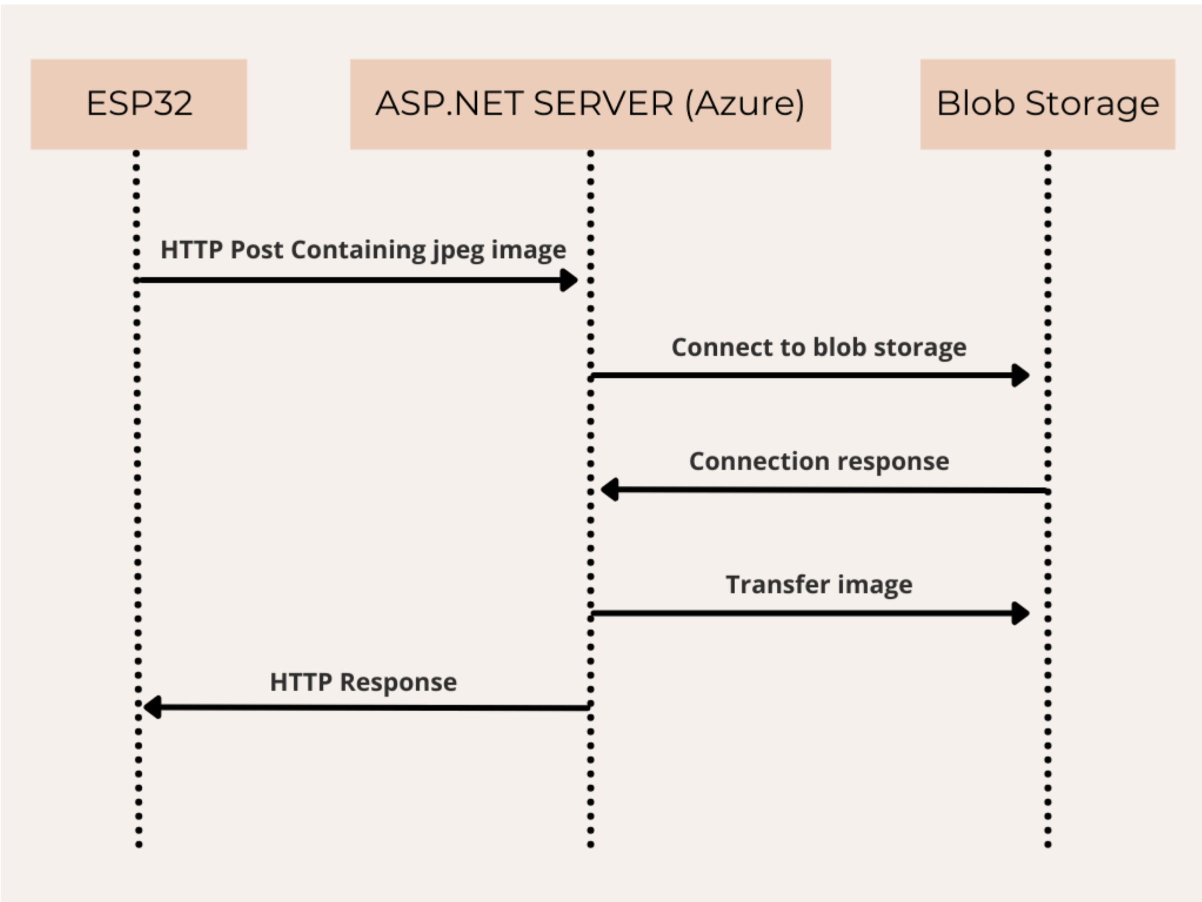

***fig 16**, Interaction of systems*

Figure 16 illustrates the interaction flow between the ESP32, ASP.NET server, and Azure Blob Storage, assuming all requests are approved. The process is as follows:

1. The ESP32 sends a JPEG image via an HTTP POST request.
2. The server receives the image. If the server has received a valid image the server then attempts to connect to Azure Blob Storage.
3. Upon establishing a successful connection, the server transfers the image to Azure Blob Storage.
4. The HTTP response is OK.

We have implemented a retry system on the ESP32. If the HTTP response is not OK, the ESP32 waits and then sends a new HTTP POST request with the image. This approach ensures that temporary server connection issues do not disrupt the process. If the connection to the blob storage fails, the server will return an internal server error.

5 Evaluation & Test

In this section, we will address to which extent we meet the requirements outlined at the beginning of the report. We will provide an overview by adding an evaluation column to our requirements table, where we refer to the segment that answers each requirement. If a requirement is not fulfilled, we will describe to which extent we meet the requirement. By evaluating our progress against the initial requirements, we can provide a thorough assessment of our project contributions. This helps in identifying areas for improvement and setting a clear path forward for future development.

Aspect	Minimum requirement	Evaluation
Photo	We want the product to be able to take clear images of the pollen input. The images should be clear enough to train a machine learning model on it.	4.3 Microscope lights Partially solved: Images are not clear enough - ML model can be trained, but precision would be low.
Calibration	The microscope should be able to calibrate the distance to the tape for optimal image capturing. When the calibration is done the microscope should be stable and locked in place, so that it doesn't move.	4.4 Microscope lens implementation Requirement met.
Portable	The product should be able to be placed	4.2 The box

	anywhere, regardless of the environment. We want the product to be as low maintenance as possible. The product should also be able to withstand all kinds of weather, and should also be water resistant.	Partially solved: Needs battery and satellite networking implemented to completely fulfill requirement.
Intake	The intake needs to collect pollen particles. The amount of air processed should be able to accurately calculate pollen counts. This means we need to process at least 10 liters per minute (Lampugnani & Ebert, 2018).	4.5 Airflow and collection tube Requirement met.
Pollen count	To validate the product, we want to compare the pollen count we get, with the pollen count Astma Allergi Danmark delivers. We can calculate our pollen count by using this equation:	4.6 Calculating a pollen count using our box design Requirement met.
Monitoring period	The product should give an output at least one time every day, so that it is possible to replicate the daily pollen count. We want to be able to monitor pollen all year around, but most importantly when the pollen season is active.	4.6 Calculating a pollen count using our box design Requirement met.
Surface examined	We aim to examine the largest surface area as possible, however, if we can not figure out how to concentrate the sample we can take a smaller area and give an estimate of the whole surface. We want to look at a minimum of 10 % of the tape.	4.1 Tape and mechanisms Partially solved: We are examining 0.57% of the tape, but are compensating with exposure time and amount of air processed.
Usability	The product should be able to run with minimum human interaction. The results should be easy to retrieve, and the user shouldn't need specialized knowledge to handle the product.	4.8.2 Server & Storage Requirement met.
Available / Cost effective	We want the solutions we pick to be affordable, and available for the common user. The goal is to make a product that everyone can afford and use.	4.7 Availability and cost effectiveness Requirement met.
Scalable and clean	The code should be scalable to multiple units, as well as expandable and easy to read.	4.8.1 Pollen Trap Software Requirement met.

5.1 Evaluating Feasibility of Pollen Trap

Our goal is to demonstrate the feasibility of creating machine learning-ready data. To assess this, we will be running the feasibility tests outlined in the limitations section, as well as manual data comparison. These tests will directly measure how well we have achieved this objective.

5.1.1 Manual detection

We have evaluated a pollen collectiontest from 15:35 May 30 to the 31st 2:25.Looking at the images it is clear that it is out of focus.

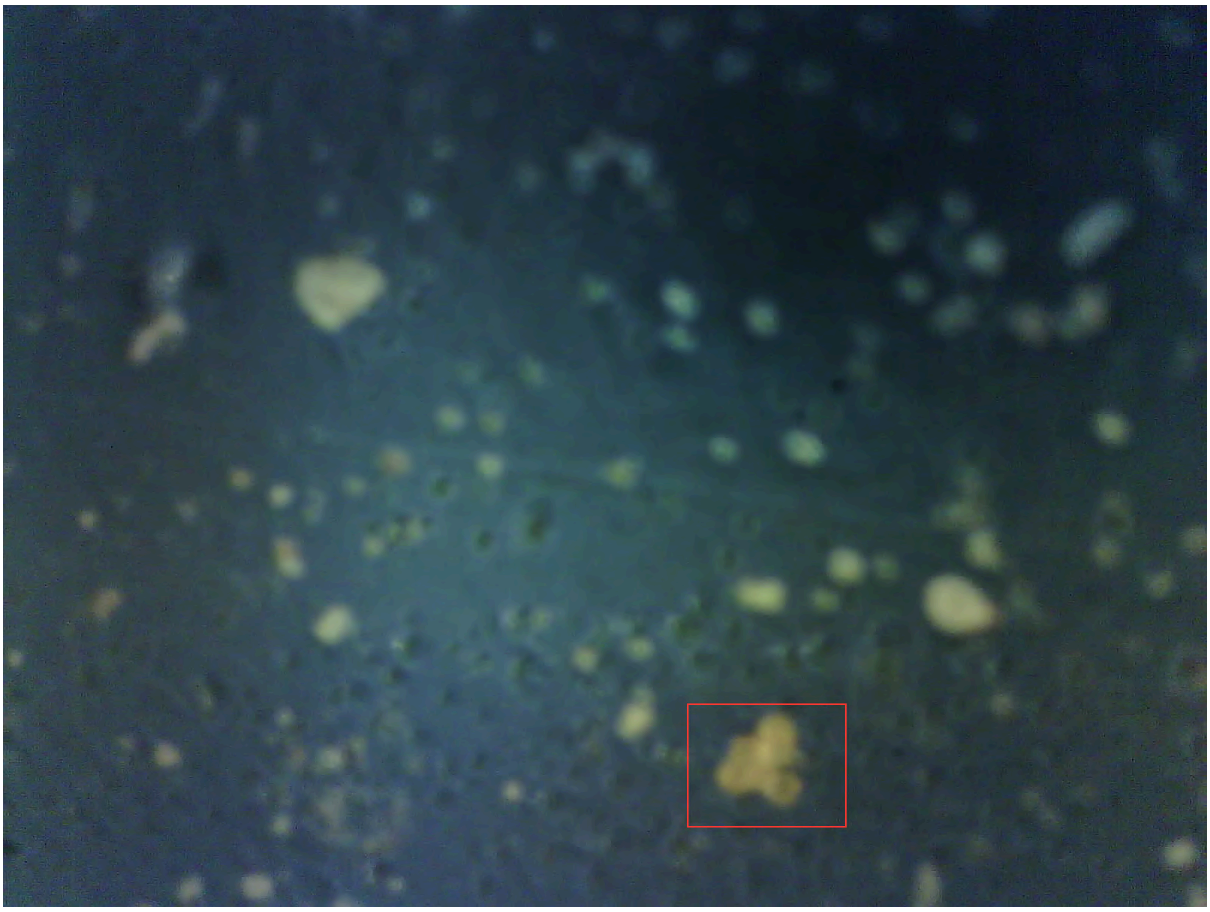

Fig 17*, May 30 16:05*

At 16:05, we observed a mass of objects with a yellow hue, suggesting the presence of pollen. We compared these images to pollen particles from the same tape sample, which were examined through a laboratory microscope.

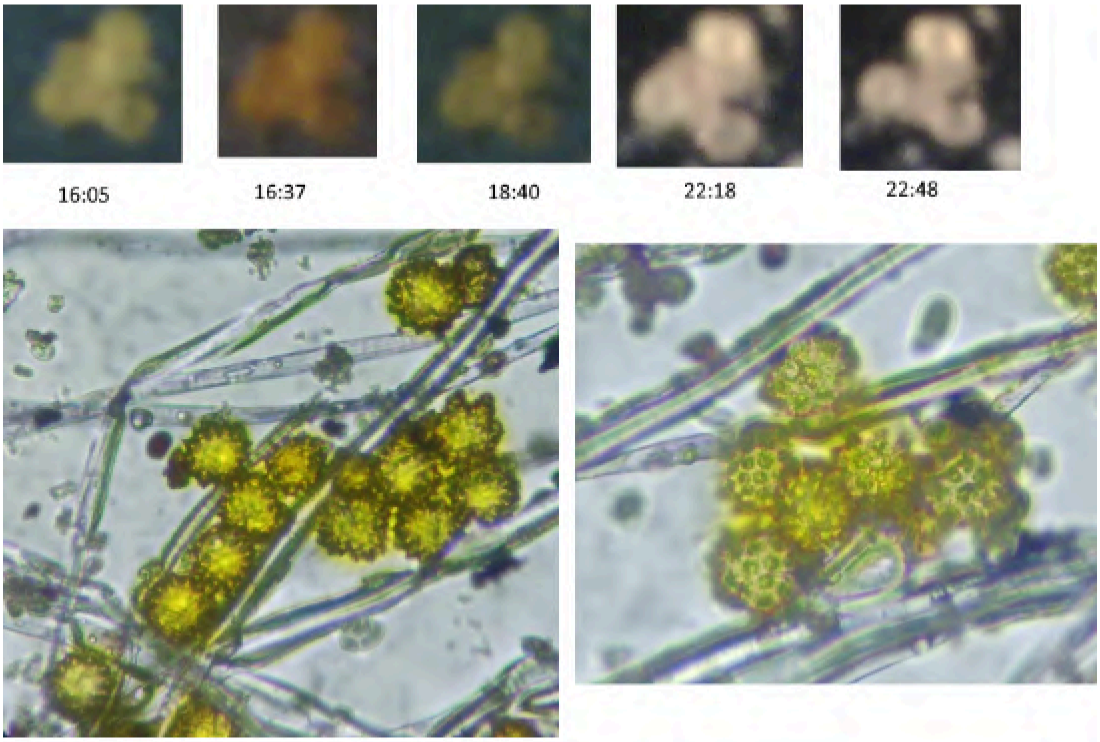

Fig 18*: Timeline of suspected pollen with pictures of Daisy pollen from the same tape sample.*

The similar clustering, combined with the color and shape, indicates the possibility of pollen. Despite some uncertainty, there is a strong likelihood that these objects are pollen, totaling five particles.

fig 19, *counting the potential pollen.*

Based on our observations from May 31 at 02:25, we identified five pollen particles, albeit with some uncertainty. Viewing the sample through a lab microscope revealed that most objects of similar size were indeed pollen (Fig. 20).

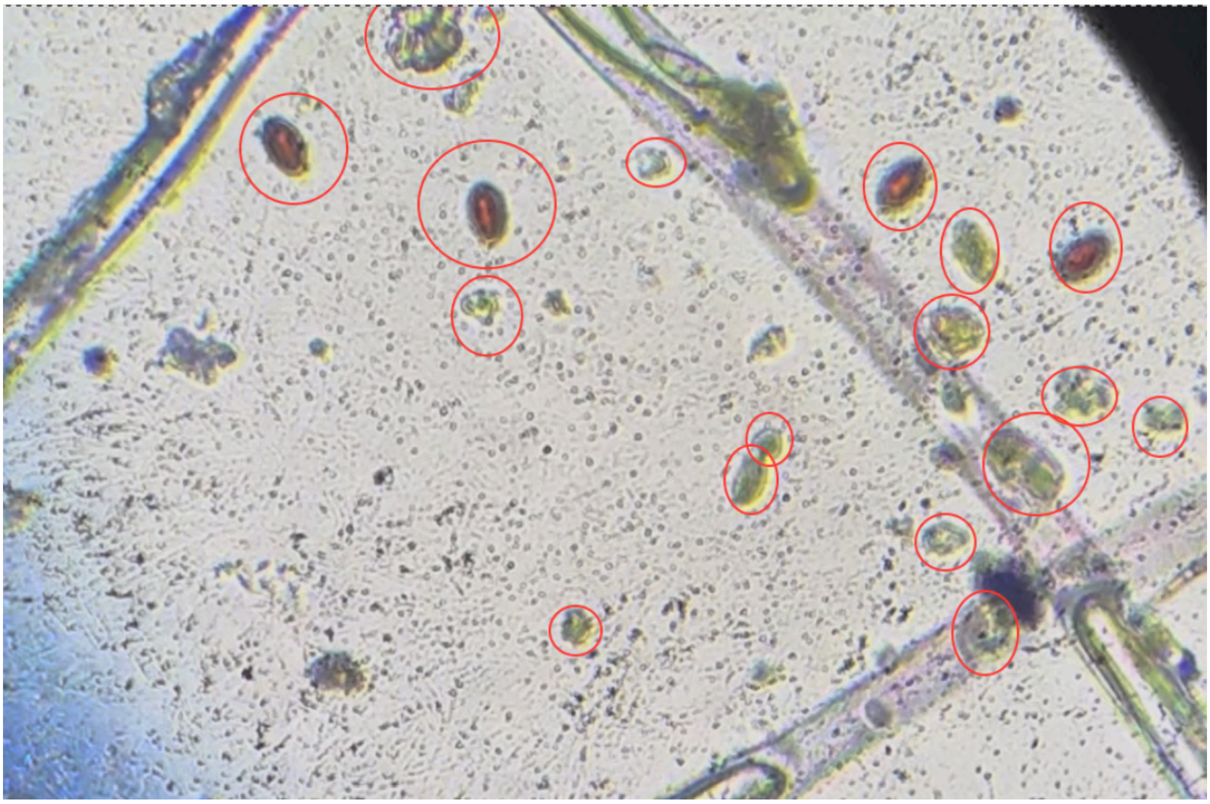

***fig 20**, image of tape sample from may 31*

This observation suggests that more objects, though unidentifiable due to blurriness, likely fit the pollen pattern. Consequently, we can cautiously estimate that there are more than five pollen particles in the image. However, for this evaluation we will use the five particles we are more confident about.

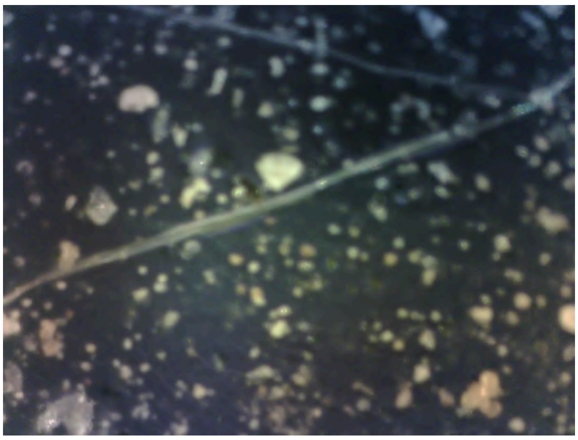	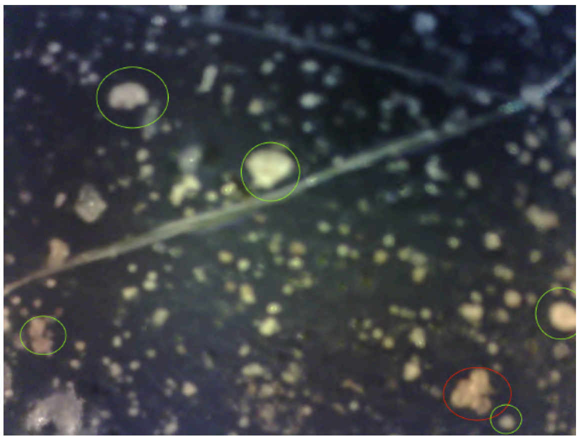
***fig 21**, May 31 20:14 test result*	***fig 22**, May 31 20:14 test result.* *Green circles around possible pollen* *Red circle around possible Daisy pollen.*

Using the formula devised above, we can then calculate the final approximate pollen count with our 5 observed pollen grains:

$$C = \frac{5}{0.145m^3 \bullet 360 \bullet 0.0057} = \frac{5}{0,3} = 16.80 \frac{N}{m^3}$$

The pollen count for May 31, as reported by Astma-Allergi Danmark (n.d., retrieved May 31), was 13. In comparison, our measurement yielded a pollen count of 17. (Fig. 23)

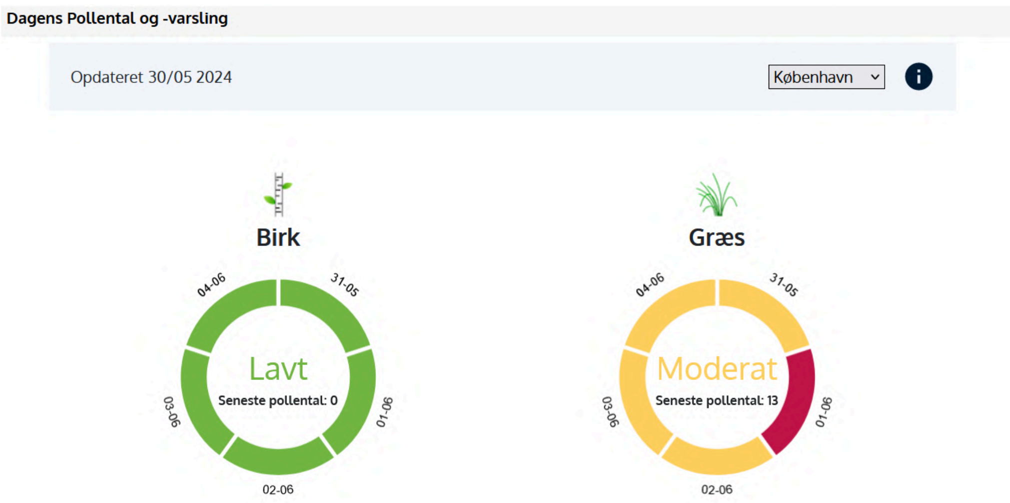

fig 23, website of *Astma-Allergi Denmark where the pollen numbers/counts is shown*

5.1.2 Machine Learning Detection

Using pictures from our pollen trap, we are training a barebones/makeshift image recognition model to do detection of particles on said pictures - we are *not* doing classification. This test is purely for feasibility testing to which degree our data is ready for future machine learning processing.

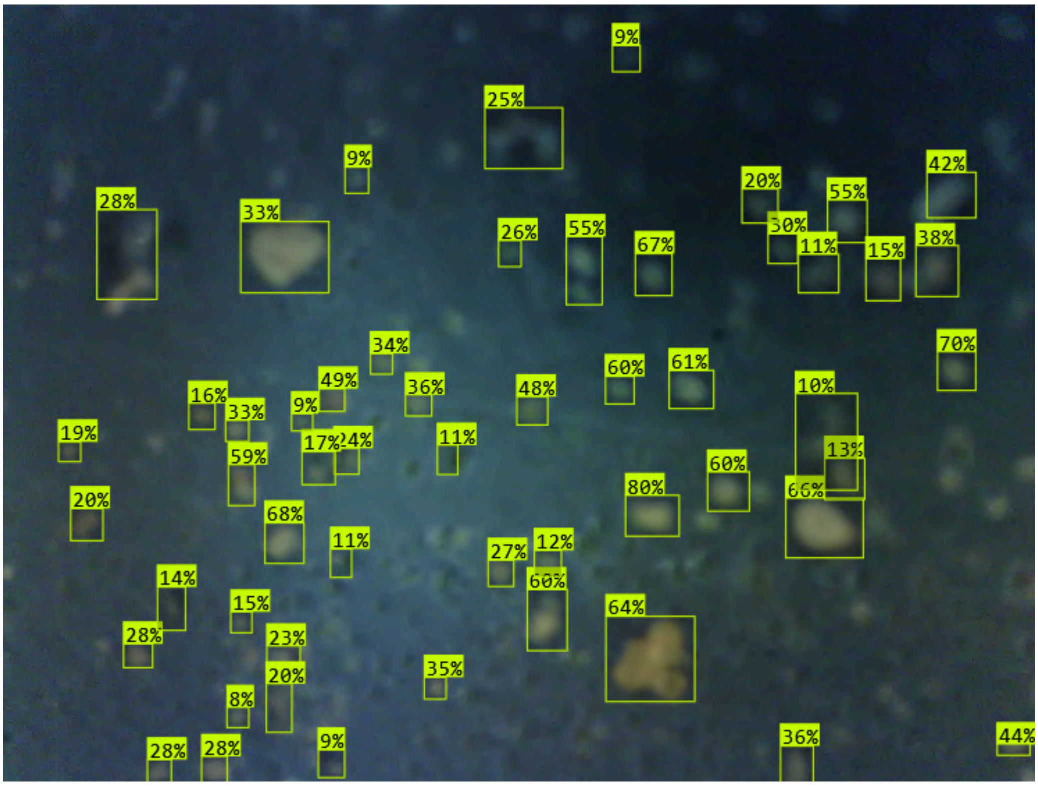

fig 23, *Image recognition model (Roboflow 3.0 Object Detection) on own pictures (percentages is confidence level)*

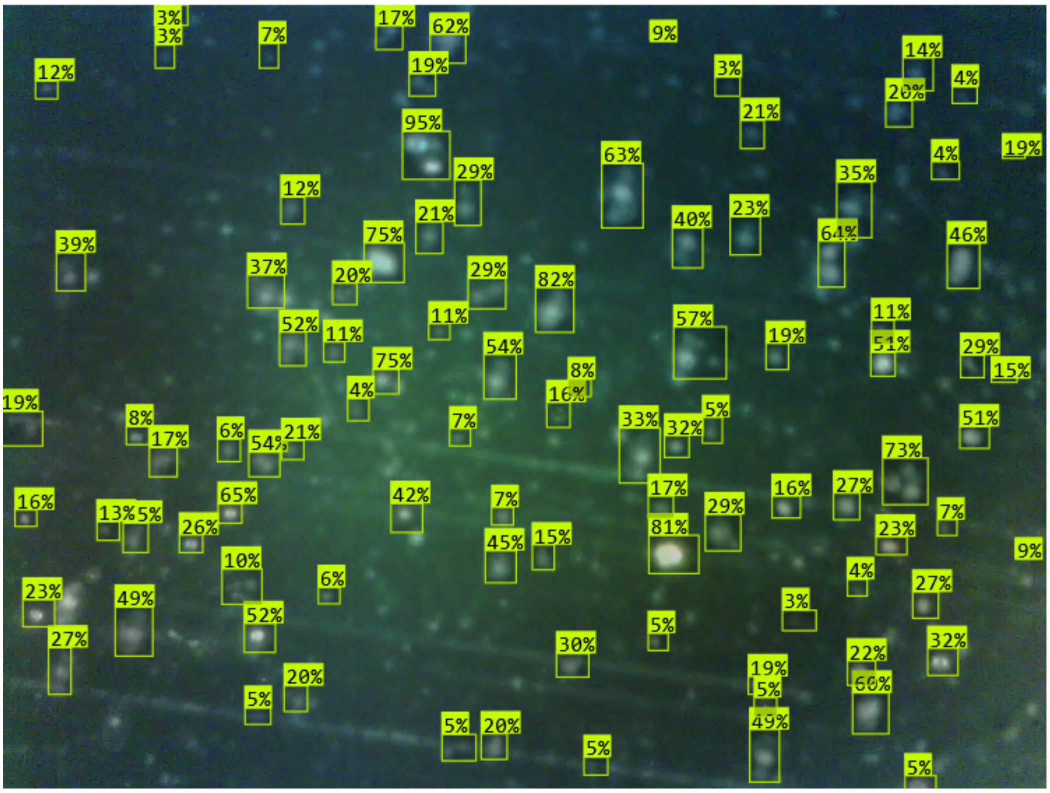

fig 24, *Image recognition model (Roboflow 3.0 Object Detection) on own pictures (percentages is confidence level)*

Despite the images showing low confidence in identifying particles (the model's only class), the results are promising. With a mere 3 training images, 1 for testing, and a final validation image (compared to a preferable 70/20/10 split of 500 images), the model still demonstrates some identification capability. This suggests the project's core concept has merit. While building a fully functional model might be beyond the current scope, achieving particle identification seems feasible.

5.1.3 Detection Tests Discussion & Conclusion

One critique of our system is the lack of an effective method for filtering out non-pollen particles. Although we attempted to use a cyclone filter, this approach was unsuccessful. Implementing an effective particle filtration system would significantly enhance the performance of an image recognition model, as many particles appear very similar.

Alternatively, using a higher-quality image sensor that is easier to focus through the microscope would highlight the distinct characteristics, colors, and contours of different particles. This improvement would also contribute to the effectiveness of an image recognition model by providing clearer and more detailed images.

Based on our manual detection test, we can conclude that our pollen trap can be used to capture pollen particles and take pictures of them under a microscope, to an acceptable degree. However, several potential issues exist with our measurement:
Our sample contains a different type of pollen than the one reported by AAD (grass, we are collecting daisy pollen) and some pollen types in our sample may be unidentifiable due to blurry images.

After successfully training and running our machine learning model on our images from the pollen trap, it becomes evident from the results that our data is able to be used in machine learning - albeit in a very limited manner. This proves the plausibility of using our data in a future machine learning pipeline.

5.2 Comparison to State of the Art (Astma-Allergi Danmark)

Comparing our system to the state-of-the-art system from Astma-Allergi Danmark (AAD) highlights both strengths and areas for improvement.

5.2.1 Strengths and Weaknesses

Our approximation calculations, while theoretically effective for determining average pollen counts during the collection period, are sensitive to outliers due to our small field of view. Over a 6-hour collection period, a variation of just 2-3 pollen particles can significantly impact the final result. Although statistically sound, these outliers prevent us from achieving the same level of robustness and certainty as AAD, which collects data over a 24-hour period, resulting in more reliable numbers.

A notable strength of our system is its ability to produce multiple pollen counts per day. Pollen counts tend to be higher in the morning, so while AAD provides a single average daily count, this number may not accurately reflect the pollen count at any specific time. Our system, generating counts every 6 hours, offers more granular data, capturing the fluctuations throughout the day.

Moreover, AAD's use of two monitoring stations underscores the value of having multiple data points. Our system, being simpler and cheaper to implement, suggests that deploying inexpensive local pollen traps could be valuable. This approach could enhance the accuracy and relevance of pollen data for specific areas, providing more timely and localized information.

Regardless of whether we implement multiple stations, the upfront cost of 931,22 kr is cheap, and the operation cost of a little more than 50$ a month is significantly cheaper than Asthma Allergy Denmark. Although we do not know the specifics of AADs operations cost, we assume they are well above ours, based on the fact that they are a state funded organization, with multiple highly trained aero-biologists, as well as state of the art equipment.

6 Conclusion and future work

We have developed a functional prototype that collects pollen, captures images, and transmits them to a server for storage. This addresses our problem formulation by providing ordinary consumers with access to pollen counts using our mathematical formulas. However, we recognize that the accuracy of our results needs improvement. Although our current system does not meet the desired accuracy standards, it effectively demonstrates the concept of automated pollen traps. The primary challenge lies in enhancing the reliability of pollen detection and identification, mainly due to image blurriness. Despite these limitations, our project proves the potential utility of this approach, highlighting the feasibility of making pollen collection easy and accessible for ordinary consumers and allergy sufferers.

Future work should focus on enhancing image quality and implementing effective image recognition techniques to improve the accuracy and reliability of pollen detection and classification.

7 References

[1] *An integrative environmental pollen diversity assessment and its importance for the Sustainable Development Goals*. (2022, January 8). Retrieved May 14, 2024, from https://nph.onlinelibrary.wiley.com/doi/full/10.1002/ppp3.10234

[2] *Bachelor (2023)*. (n.d.). Retrieved June 3, 2024, from https://ruc-thirdroom.dk/project/bachelor-2023/

[4] *Dagens Pollental og Pollenvarsling fra Astma-Allergi Danmark*. (n.d.). Astma-Allergi Danmark. Retrieved May 31, 2024, from https://www.astma-allergi.dk/dagens-pollental/

[5] *Dandelion - Taraxacum*. (n.d.). Northumbrian Bees. Retrieved May 30, 2024, from https://www.northumbrianbees.co.uk/pollen_gallery/dandelion-taraxacum/

[6] Homemade 3D-printed ESP32+ESP32CAM holder https://www.tinkercad.com/things/kIizefIFYZe-esp32cam-v1

[7] Homemade 3D-printed glass and tape mount https://www.tinkercad.com/things/8YU2QJh4m7U-mount-to-glass-v7

[8] Homemade 3D-printed LED holder https://www.tinkercad.com/things/70PryrAIfV8-light-holder

[9] Homemade 3D-printed Microscope mount https://www.tinkercad.com/things/edH37OrNpuV-microscope-mount

[10] Homemade 3D-printed tape holder https://www.tinkercad.com/things/7riv8930B8i-fabulous-uusam-gogo

[11] Homemade 3D-printed tape plate https://www.tinkercad.com/things/caE0qjsZwl0-tape-plate

[12] Lampugnani, E. R., & Ebert, B. (2018, September 14). *Australian Airborne Pollen and Spore Monitoring Network Interim Standard and Protocols*. Australasian Society of Clinical Immunology and Allergy (ASCIA). Retrieved May 24, 2024, from

https://allergy.org.au/images/stories/pospapers/Australian_Pollen_and_Spore_Monitoring_Interim_Standard_and_Protocols_v2_14092018.pdf

[13] Nam, C. N. K., Meyer, M., Thiele, L., & Saukh, O. (2020). Automated Pollen Detection with an Affordable Technology. Paper presented at 17th International Conference on Embedded Wireless Systems and Networks, Lyon, France. https://graz.elsevierpure.com/en/publications/automated-pollen-detection-with-an-affordable-technology

[14] Nam, C. N. K., Meyer, M., Thiele, L., & Saukh, O. (2019). Poster Abstract: An Automated Real-time and Affordable Airborne Pollen Sensing System. Poster session presented at 18th ACM/IEEE International Conference on Informatio Processibg in Sensor Networks, Montreal, Canada. https://graz.elsevierpure.com/en/publications/poster-abstract-an-automated-real-time-and-affordable-airborne-po

[15] PC MAG. (n.d.). *Frame Buffer*. https://www.pcmag.com/encyclopedia/term/frame-buffer

[16] *Pollen Allergies*. (n.d.). Asthma and Allergy Foundation of America | AAFA. Retrieved May 14, 2024, from https://aafa.org/allergies/types-of-allergies/pollen-allergy/

[17] Pollen Sense. (n.d.). *APS400 Particulate Sensor*. https://www.pollensense.com/pages/automated-particle-sensors

[18] *What is darkfield and brightfield microscopy?* (n.d.).
from
https://bio.libretexts.org/Bookshelves/Microbiology/Microbiology_(OpenStax)/02%3A_How_We_See_the_Invisible_World/2.03%3A_Instruments_of_Microscopy

[19] *What Is Pollen Count, and How Does It Change?* (n.d.). ZYRTEC. Retrieved May 14, 2024, from https://www.zyrtec.com/allergy-guide/outdoors/best-times-low-pollen-count

[20] *What Is Pollen? | Northeast Allergy*. (2023, November 6). Northeast Allergy, Asthma, and Immunology. Retrieved May 14, 2024, from https://neaai.com/what-is-pollen-types-allergies-and-management/

[21] 3D-printed roof poles https://www.thingiverse.com/thing:2990524