DOCTAR CASE-STUDY REPORT

Q1-**Exploratory Analysis**

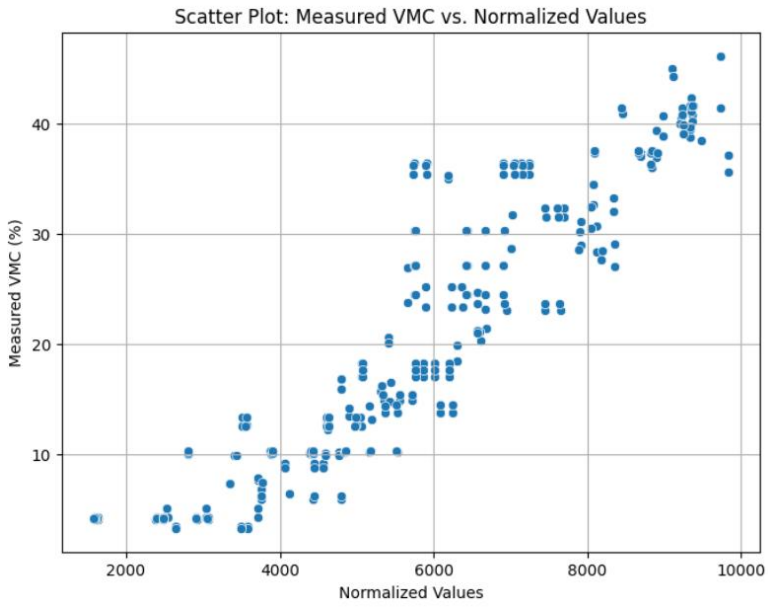


Figure 1: Plot of Measured VMC vs. Normalized Values.

The scatter plot between the Measured VMC and Normalized Values shows a strong positive linear relationship, and the Pearson Correlation Coefficient is 0.9162, which indicates a high linear relationship between the two variables.

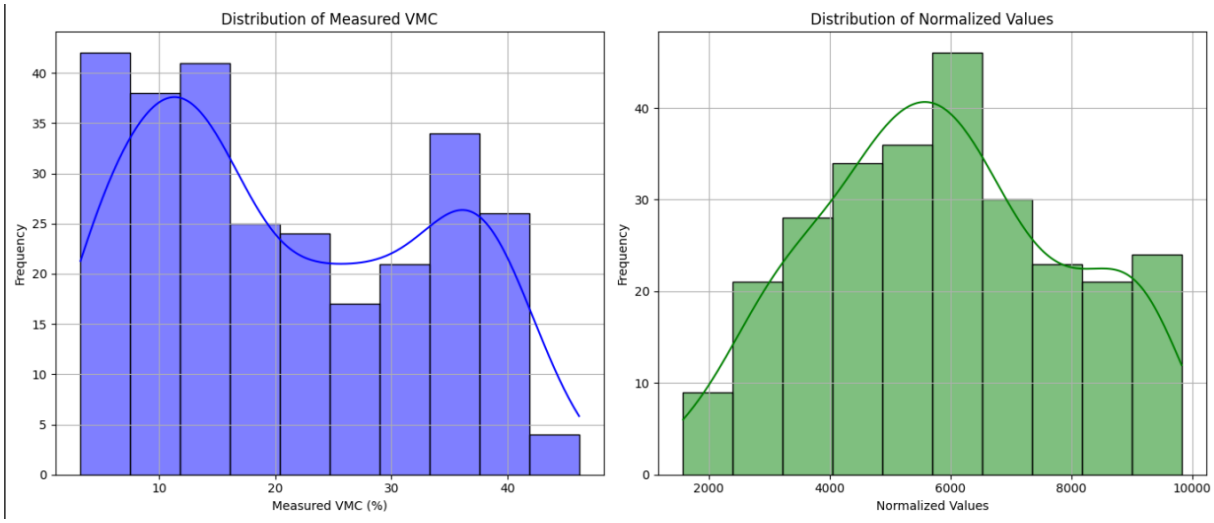


Figure 2: Distribution Histograms of Two Variables.

The histogram indicates that both variables do not perfectly fit with normal distribution but have multi-model tendencies. This suggests that while the linear relationship is strong, there may be variance shifts in different value ranges.

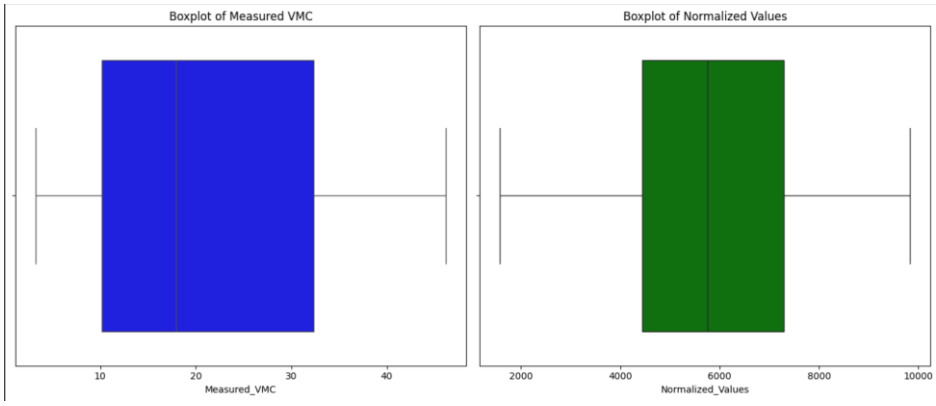


Figure 3: Box Plots of Variables

The Box plots are used for detecting any outliers, and the distributions are reasonably symmetric, supporting the use of linear regression. After that analysis, as a first approach, I applied linear regression to see the results then evaluate the residual.

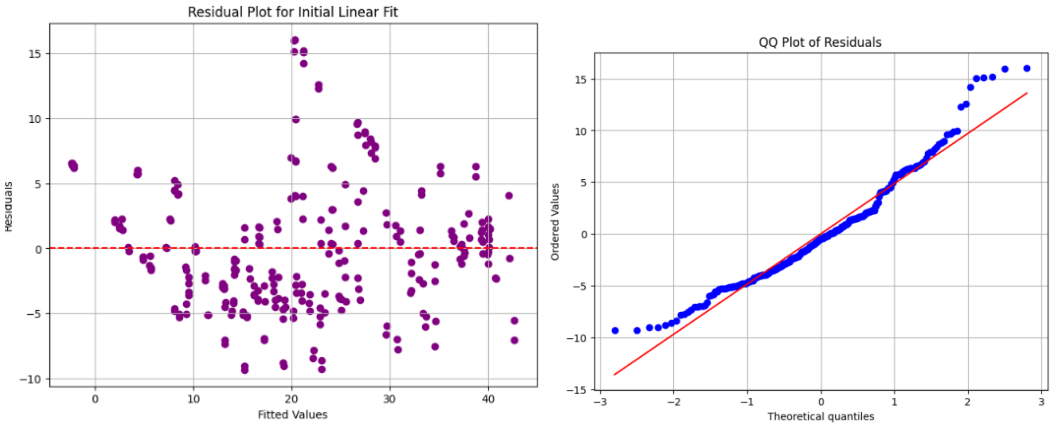


Figure 4: Residual Plot and QQ Plot of Residuals

That QQ plot indicates that the residuals follow a normal distribution with some deviation at the tails, and the residuals are scattered randomly around zero, which supports the use of linear regression. Further analyse the data and find the perfect possible regression line, I applied different transformations to the data.

Original	0.9162
Log X	0.8689
Log Y	0.9024
Log-Log	0.9075
Square Root X	0.9010
Square Root Y	**0.9206**
Inverse X	-0.7473
Inverse Y	-0.7887

Table 1: Correlation Results of Transforms

As you can see on the table, Square Root Y gives a slightly better result than the original. Then, as a first straightforward approach, I used linear regression to fit a line to transformed data.

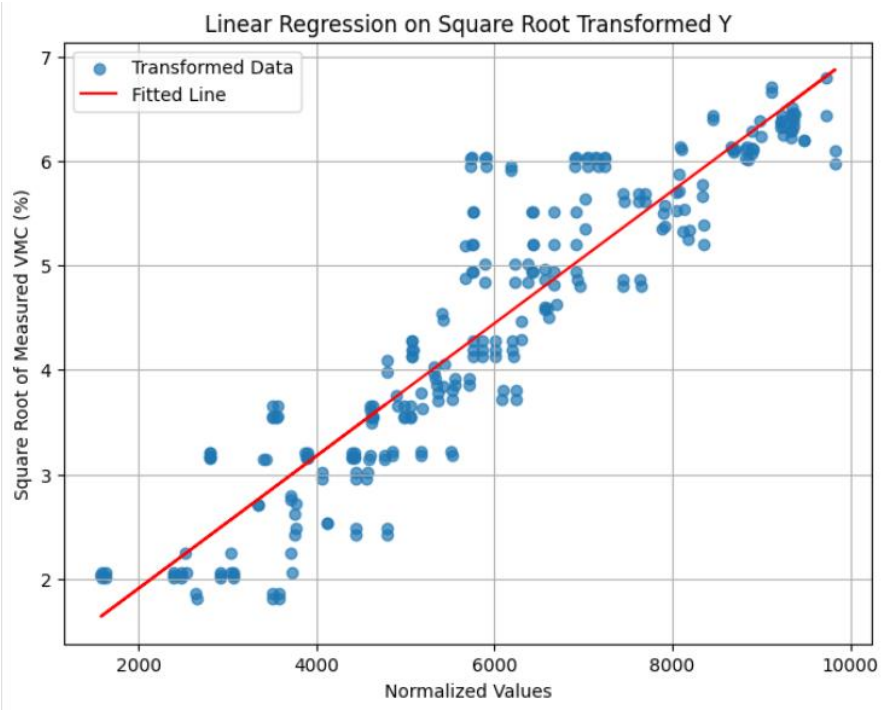


Figure 5: Plot of the Square Root Transformed Y with Linear Regression.

As shown in Figure 5, it gives slightly better results than the original data. The untransformed data already shows a very high Pearson correlation, applying a square root transform on Y slightly increases the correlation to 0.9206 as a result, the relationship becomes even closer to linear when we apply a square root Y transformation.

Part b) **Model Design and Validation and Selection**

Methods	MAE	MSE	R^2
Linear Regression	4.2160	29.8504	0.7981
Polynomial Regression	4.1130	30.0653	0.7901
Isotonic Regression	3.2700	21.7915	0.8478
Monotonic GBM	3.3286	21.9536	0.8467

Tabel 2: Candidate Model and the Performances for Untransformed Data

Methods	MAE	MSE	R^2
Linear Regression	3.8090	24.1567	0.8393
Polynomial Regression	3.6815	23.6299	0.8428
Isotonic Regression	2.8007	15.4897	0.8969
Monotonic GBM	3.2557	17.8442	0.8813

Tabel 3: Candidate Model and the Performances for Square Root Y Transformed Data

When analyzing the metrics from Table 2 and Table 3, we observe that linear approaches, such as Linear Regression and Polynomial Regression, provide a good approximation of the relationship between Measured VMC and Normalized Values. Among the strictly linear models, Polynomial Regression (Degree 2) consistently provides lower error rates and better R^2 scores compared to simple Linear Regression, especially when applied to the transformed dataset.

However, when we consider non-linear models, the results clearly indicate that Isotonic Regression delivers the best overall performance. Isotonic Regression consistently achieves the lowest Mean Absolute Error (MAE), the lowest Mean Squared Error (MSE), and the highest R^2 scores across both the original and transformed datasets. This confirms that the relationship between the sensor readings and soil moisture contains non-linear patterns that are better captured by monotonic, non-parametric models.

Furthermore, even after applying a square root transformation on the target variable (Measured VMC) to linearize the relationship, Isotonic Regression remains the most accurate model. This demonstrates the flexibility and robustness of Isotonic Regression in handling monotonic, non-linear trends in the data.

In contrast, Polynomial Regression emerges as the best linear alternative when the goal is to maintain a simpler, parametric model structure. While it cannot fully capture the complex patterns as effectively as Isotonic Regression, it provides a reasonable balance between model simplicity and prediction accuracy

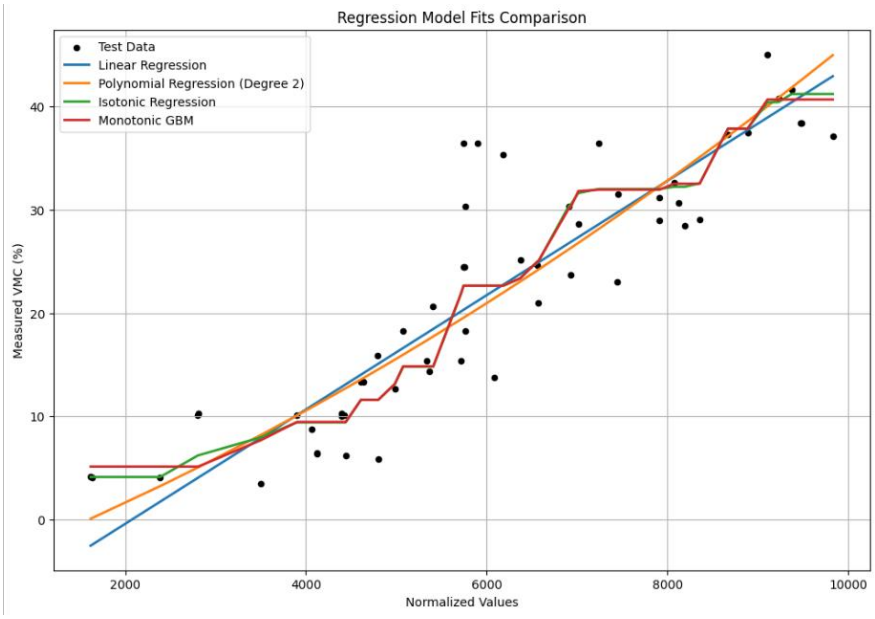


Figure 6: Regression Models for Original Data

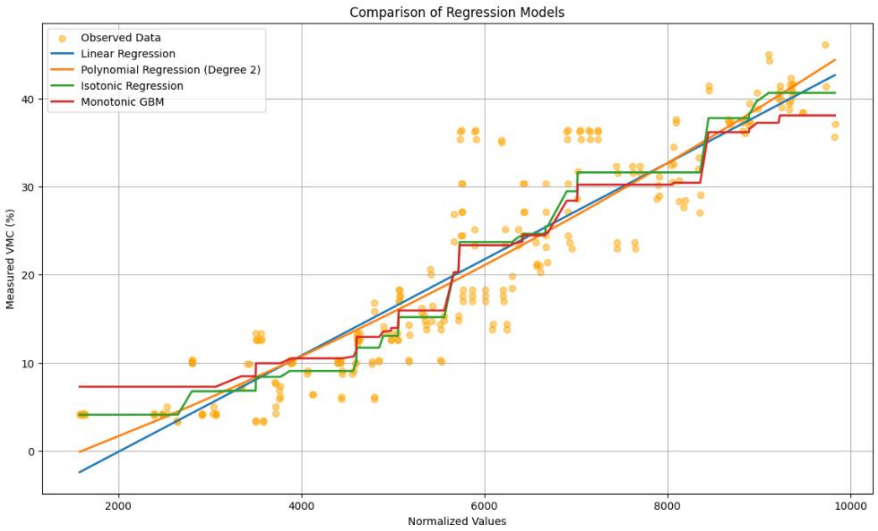


Figure 7: Regression Models for Transformed Data

Part c) **Robustness & Drift Monitoring**

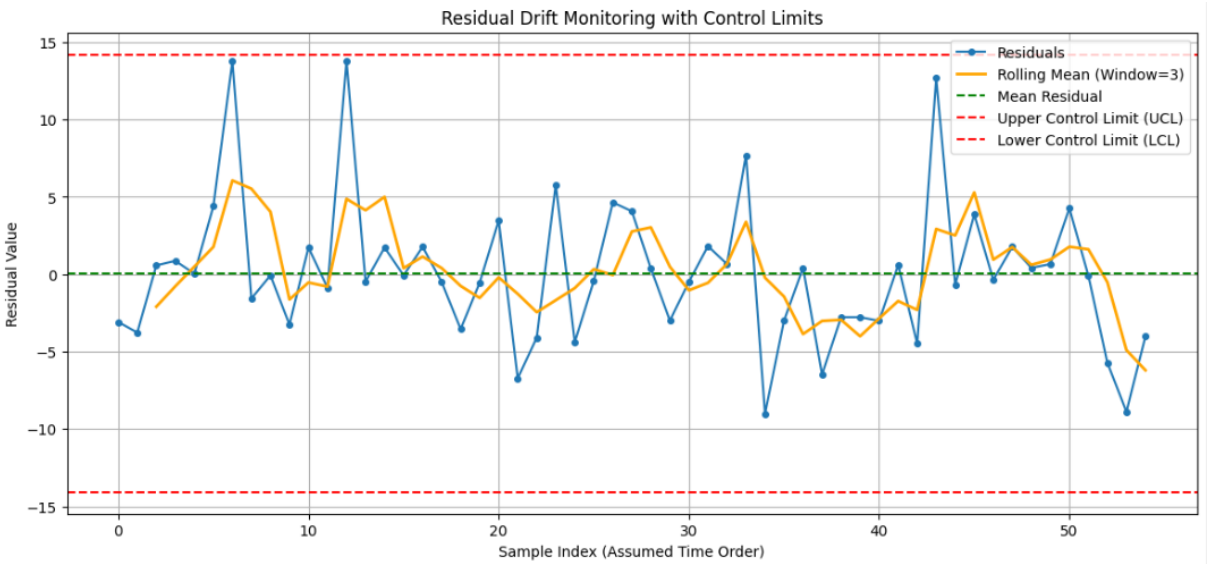


Figure 8: Plot of Drift Monitoring for Original Data.

The residual drift monitoring chart displays model residuals over time with rolling mean and control limits set at three times the standard deviation. The residuals fluctuate around zero, and both the individual residuals and the rolling mean remain well within the control limits. This indicates that the model is stable and not affected by data drift. The monitoring approach effectively tracks potential deviations and ensures the model maintains consistent performance over time.

Q2-**Black-Fly Detection**

I used two different approaches for Black-Fly detection. The first one is SAM + Object Classifier, and the second one is SAM + CLIP integration. SAM (Segment Anything Model) crops all the objects it detects in the image, then sends them either to the object classifier in the first approach or to CLIP in the second approach. The object

classifier did not perform as well as CLIP. The primary reason for this is the presence of unclear and heavily stained images in the dataset. Due to the dataset quality, the model sometimes misclassifies the stains on the trap images as flies, leading to wrong detections. Cleaning the dataset could lead to much better results. For the CLIP approach, I fine-tuned CLIP with black-fly images, which resulted in a much better CLIP classifier. Since I considered the test dataset to be cleaner, I trained the model with the test dataset instead, and it produced better results. You can see the results in the images in Appendix. The best and most accurate approach for this task would be to annotate the black-flies on the trap images with bounding boxes and then use an algorithm like YOLOv8.

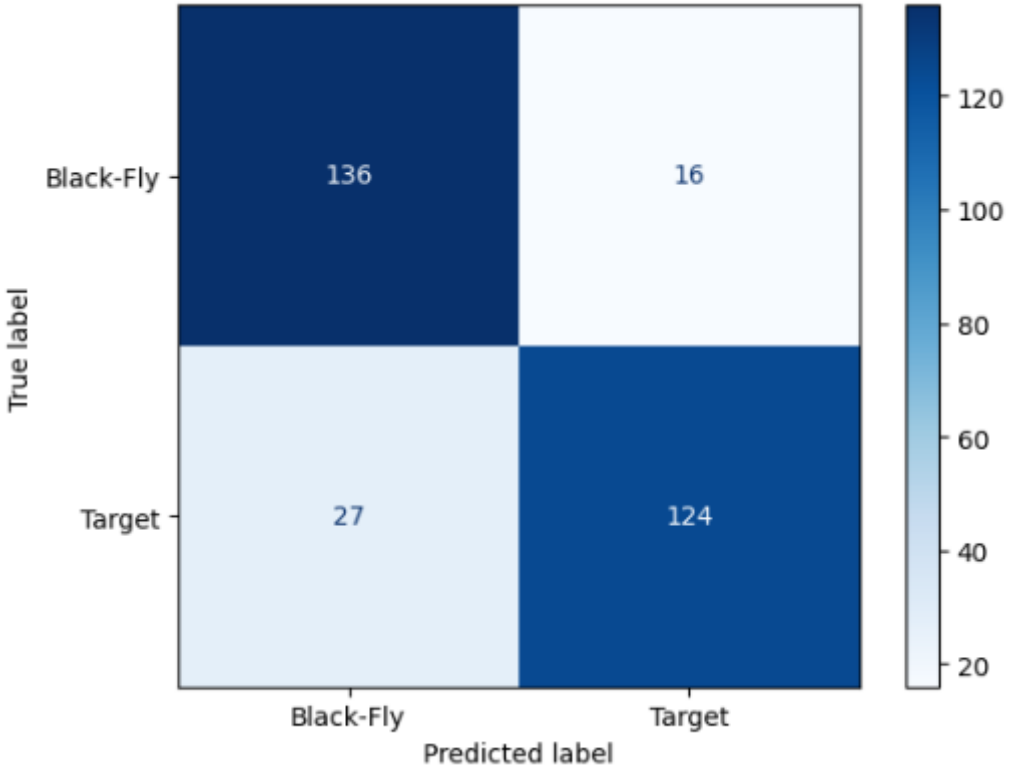


Figure 9: Confusion Matrix of the Object Classifier.

Even the Classifier gives good result when we try on the Trap images it gives wrong prediction so I preferred to use my second approach SAM+CLIP for that purpose however if the classifier trained with better dataset that probably give better and simpler approach for that task.

Q3)

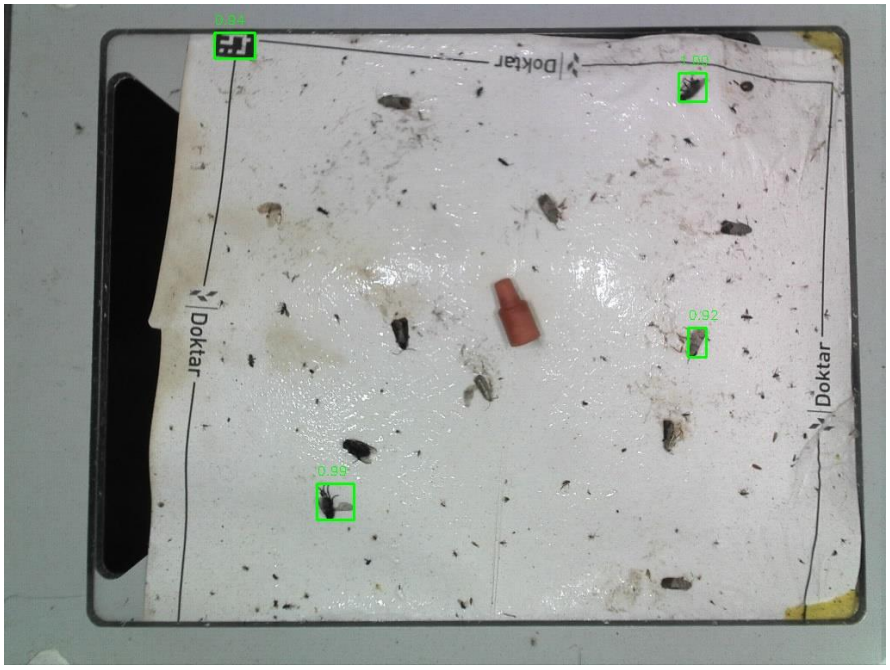
To make the black fly detection pipeline available as an API production-ready, the system would be made such that it can support the multi-step processing approach you have established. The user loads Trap images through a client interface, either mobile or web-based. The request is handled by an API service built utilizing FastAPI to control the flow of the pipeline. The initial process entails

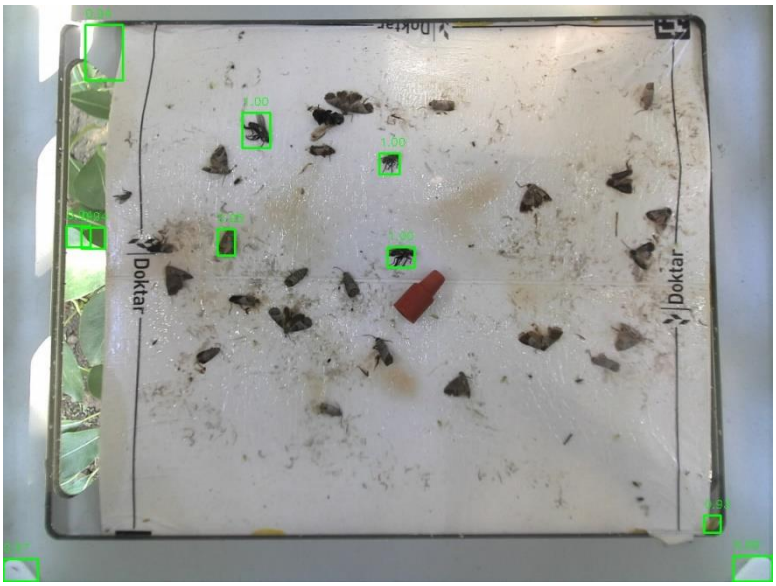
sending the image to a segmentation module that employs the Segment Anything Model (SAM) for generating possible insect masks. The regions are cropped and successively passed to a CLIP-based classifier, which matches the crops with the black fly class by a threshold degree of similarity. The system aggregates the black fly detection, annotated bounding boxes are illustrated on the input image, and the final result is sent to the user. This multi-stage detection ensures that small insects can be accurately localized and classified even for low-resolution images.

The API and model services would be containerized with Docker and could be deployed on a cloud-based GPU-enabled server (e.g., AWS EC2 or GCP Compute Engine). The segmentation and classification models can be hosted in the same container or transferred to microservices for scalability.
TorchServe would be utilized for efficient serving of the models, and FastAPI would offer upload image and retrieve result endpoints. The images would be stored using cloud storage services like AWS S3, and a simple PostgreSQL database can be utilized for storing predictions and user interactions for traceability and monitoring. The API and models would enable continuous integration and deployment pipelines, managed through GitHub Actions, to offer rapid updates and versioning. Security levels like JWT-based authentication, HTTPS, and input validation would protect the API and user data. System health and prediction performance would be tracked with monitoring tools like Prometheus and Grafana to ensure reliability and scalability.

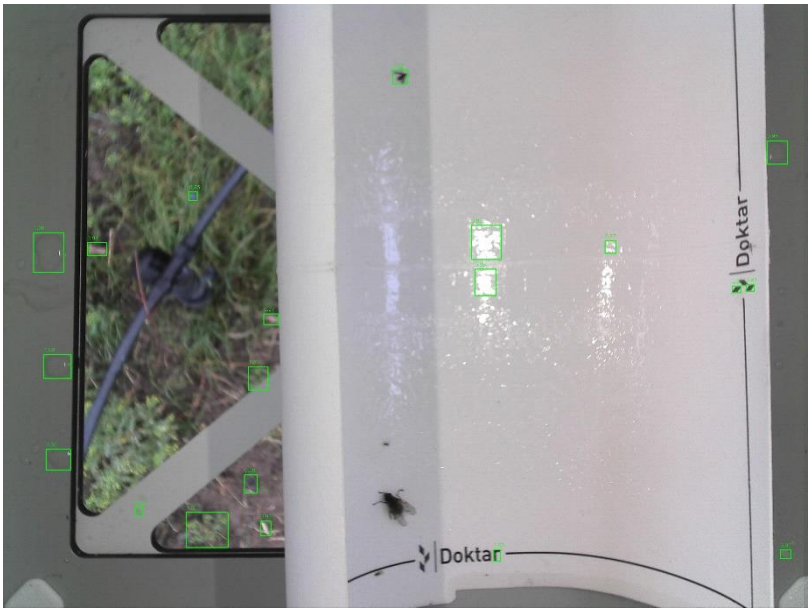
Appendix

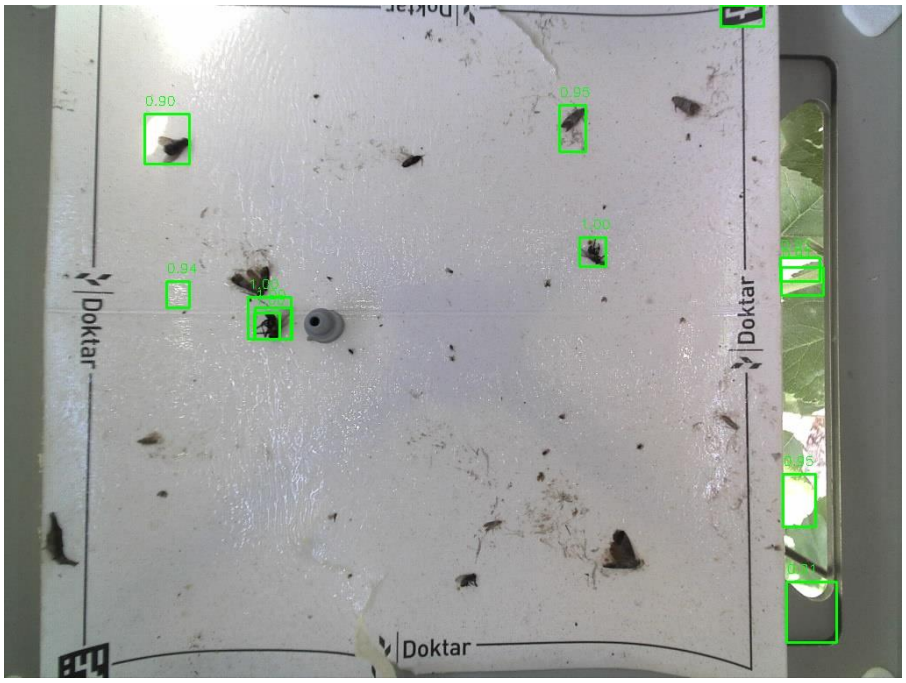
SAM+Object Classifeir

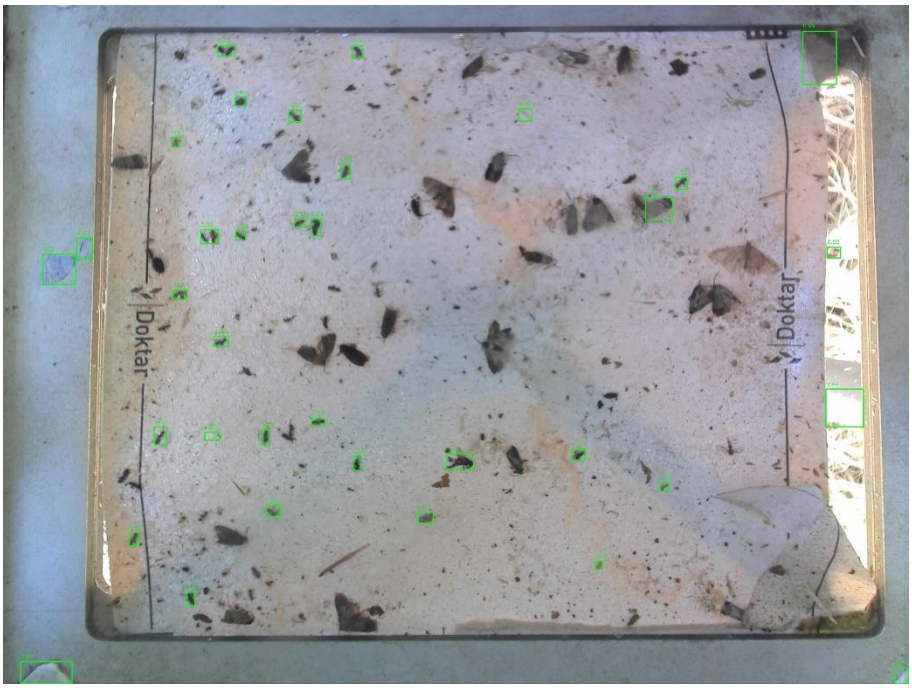



Doktar
Doktar

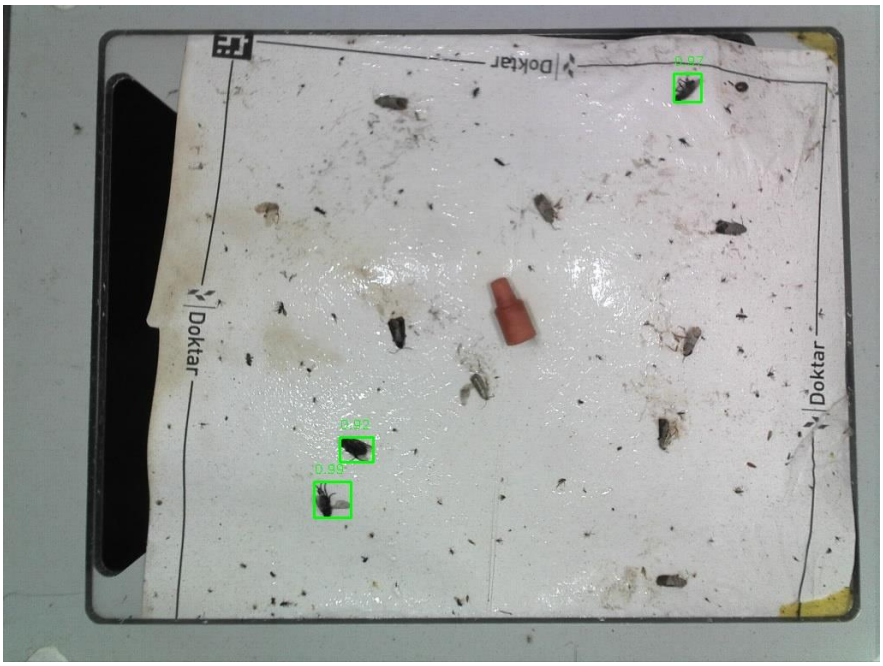

Doktar
Doktar
Doktar

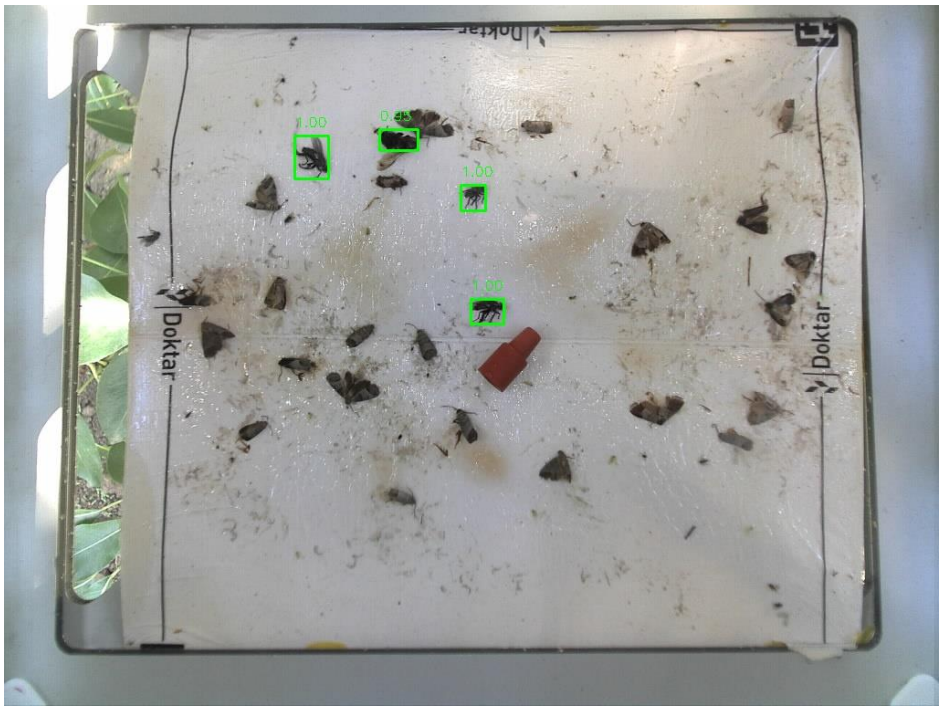

Doktar
Doktar

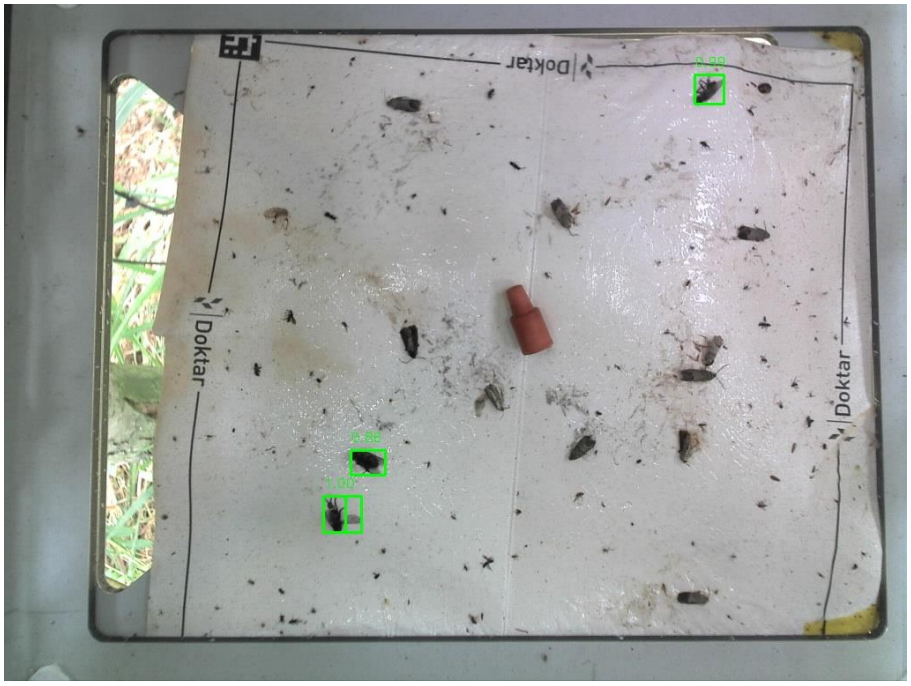

0.90
0.95
1.00
0.94
Doktar
Doktar
Doktar

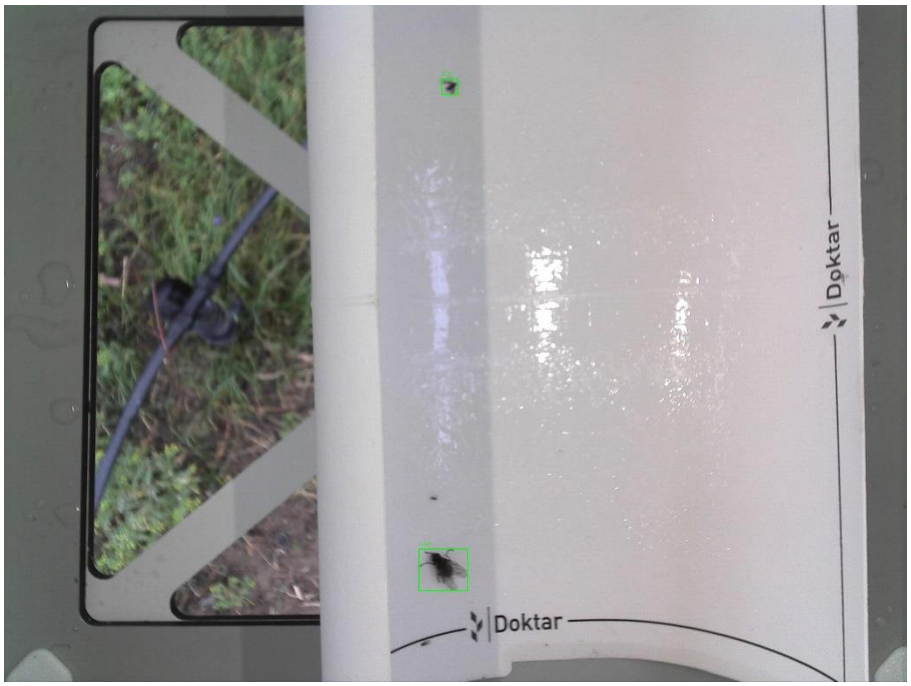

Doktar
Doktar

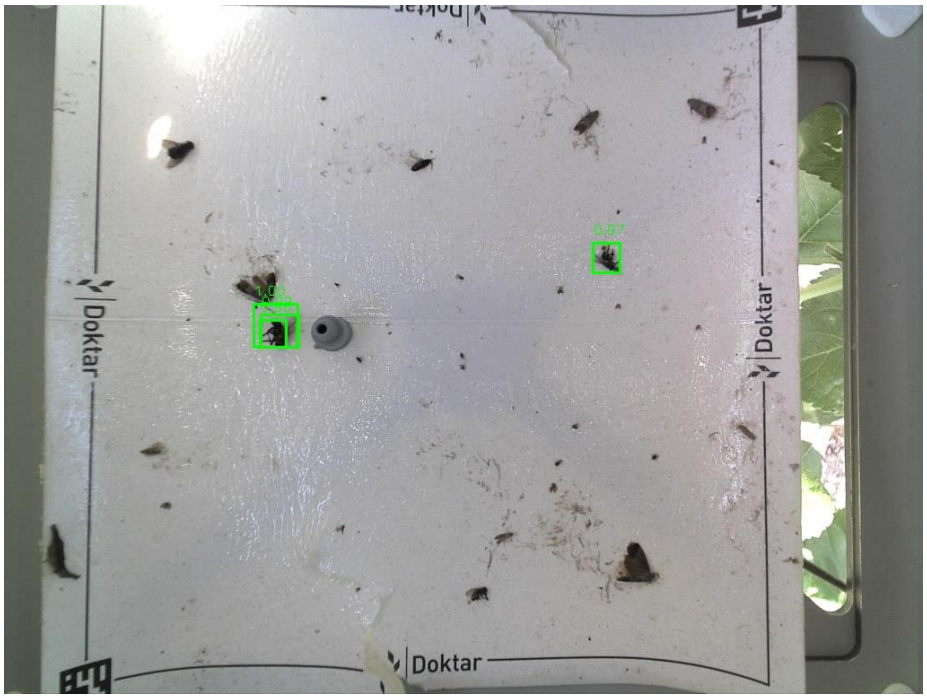
SAM+CLIP

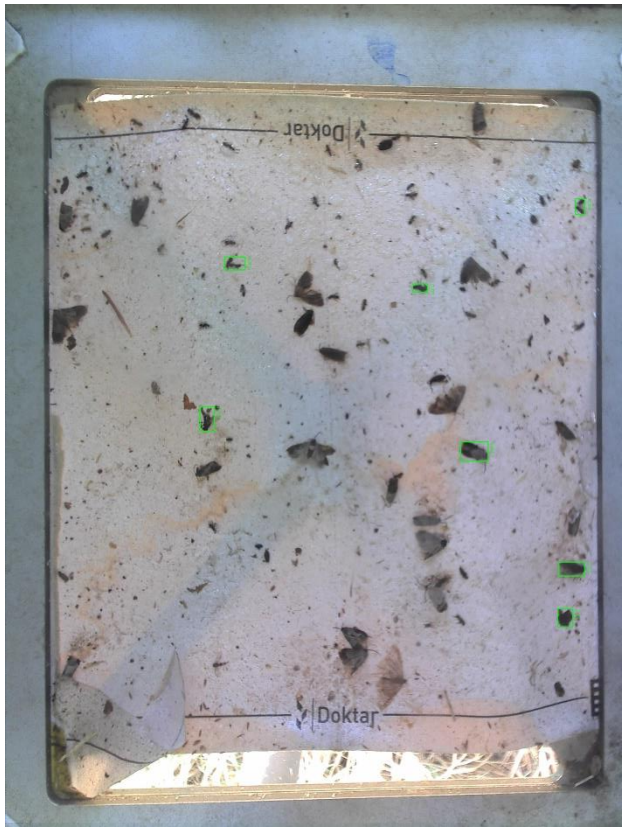



1.00
1.00
1.00
Doktar
Doktar


0.99
1.00
Doktar
Doktar


Doktar
Doktar


Doktar
Doktar
Doktar


Doktar
Doktar