Cukurova University Journal of Natural & Applied Sciences (2024) 3(2): 53-64
e-ISSN: 2822-2938

Cukurova University
Journal of
Natural & Applied Sciences

Original Paper

# Development of an Autonomous UAV-Based Irrigation Decision Support System Utilizing Image Processing and Machine Learning Techniques

**Mohamad Bashir AJAM**[1], **Hakan YAVUZ**[1]

[1]Çukurova University, Department of Mechanical Engineering, Adana, Turkey

**ORCID IDs of the authors:** M.B.A. 0009-0006-8477-6325; H.Y. 0000-0002-6166-0921.

**Cite this article as:** Ajam, M.B., Yavuz, H. (2024). Development of an Autonomous UAV-Based Irrigation Decision Support System Utilizing Image Processing and Machine Learning Techniques, Cukurova University Journal of Natural & Applied Sciences 3(2): 53-64. https://doi.org/10.70395/cunas.1511336.

**Abstract**

Efficient management of water resources is essential for sustaining the global food supply amidst growing populations and climate change. Traditional irrigation methods are often plagued by inefficiencies, leading to significant water wastage. This paper presents the development and validation of an autonomous UAV-based irrigation system that leverages advanced image processing and machine learning techniques to optimize water usage in agriculture. The system employs standard low-cost cameras to capture high-resolution aerial images, which are processed to accurately predict the water needs of the plants and inform irrigation deci-sions in real-time also it can do autonomous watering by controlling the electrical water valve in the specified irrigation areas. Comprehensive field tests conducted on pepper crops demonstrate the system's ability to enhance water use efficiency and improve crop yields. By integrating state-of-the-art technologies such as TensorFlow techniques, scikit-learn and Convolutional Neural Networks (CNNs) for machine learnig with an overall accuracy of 91% and Precision of 85%, image analysis and autonomous navigation capabilities, the proposed solution represents a significant advancement in precision agriculture. As a conclusion, the developed system waters the crops where and when it needed. So as a result, it reduced water consumption up to 31.5% while maintaining or enhancing crop productivity, thereby promoting sustainable agricultural practices.

**Keywords:** Precision agriculture, autonomous UAVs, image processing, machine learning, irrigation management.

## 1. Introduction

### 1.1 Background

#### *Global Water Resources and Agriculture*

Water resources are vital for the global population, with agriculture being a significant consumer (Figure 1.). However, increasing demand, deteriorating water quality, and declining groundwater levels threaten these resources. In Turkey, high investment costs and challenging terrain hinder the full utilization of water resources, leading to significant water wastage due to a lack of precise irrigation knowledge [1].

#### *Challenges in Traditional Irrigation Methods*

Traditional irrigation methods often rely on fixed schedules or manual observations, leading to inefficient water use. It often result in overwatering or underwatering, leading to water waste and suboptimal plant health [3]. It also Require significant manual labor for monitoring and adjusting irrigation schedules. So as a result it may involve higher operational costs due to inefficient water use and increased labor requirements [4,5]. While the developed system ensures water is applied only where and when needed. It also can control water valves to manage irrigation autonomously. This approach not only conserves water but also reduces operational costs.

**Address for Correspondence:**
Mohamad Bashir Ajam, e-mail: bashir.ajam123@gmail.com

Received: Jul 5, 2024
Accepted: Sep 10, 2024

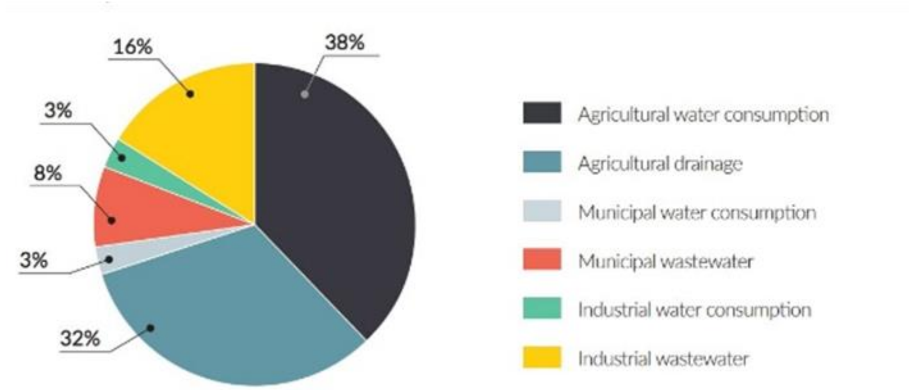



**Figure 1.** Water usage percentages [2]

### *Emergence of Precision Agriculture*

Precision agriculture (PA) uses technology to optimize field-level management, improving the efficiency of inputs like water, fertilizers, and pesticides by considering soil and crop variability [6]. Key technologies include remote sensing, GIS, and variable rate technology (VRT). One way of PA is to use soil mositure sensors across the large fields but it has some limitation. Require a large number of sensors to cover extensive fields, leading to high initial setup costs, also it need regular maintenance and calibration to ensure accuracy, which can be labor-intensive and costly [3]. Another method is to use satellites but it has some limitations. It provide large-scale data but with lower resolution compared to UAVs, making it less precise for field-level water predictions, also its subject to longer revisit times, mean-ing data may not be as up-to-date [7]. While the developed system uses live image processing and machine learning to estimate the water need so it doesn't use soil moistre sensors. Also compared to the satellite images it takes live images and can work for small scale fields so it doesn't have the limitations of the satellites.

### *Role of UAVs in Precision Agriculture*

UAVs are valuable in precision agriculture for capturing high-resolution aerial imagery and real-time data [8]. Equipped with multispectral and thermal cameras, UAVs monitor crop health, soil moisture, and pest infestations, enabling informed decision-making and optimized resource use. But it has the following limitaions like equipment cost; high-end thermal imaging UAVs are expensive, with costs ranging from several thousand to tens of thousands of dollars [9]. Data Processing Costs, it require advanced processing capabilities and software to interpret thermal data, adding to the overall cost. Finally, thermal imaging UAVs Provide detailed thermal maps, but interpreting these maps requires expertise and may not directly translate to water needs [10]. While the developed system of this paper aimed to reduce the cost of the drone by developing a low-cost autonomous drone using open source flight controller and softwares. It eliminated the need of high cost thermal and multispectral camera by using a low cost standart camera supported with machine learing. Fially the data processing is user friendly since it shows te reslts in term of graphes and watering commands even it can do watering autonomously.

## 1.2 Objectives

To address the problems and limiations of the topics that mentioned above, A unique system was developed that integrates low-cost, high-resolution imaging with advanced machine learning algorithms to provide real-time, precise irrigation management. This system aims to overcome the inefficiencies and high costs associated with traditional and current advanced methodologies, offering a practical and affordable solution for farmers.

## 2. Material and Method

## 2.2 System General Description

In response to the critical need for efficient water management in agriculture, this research proposes the development of an advanced Image Processing and Machine Learning-based Irrigation Decision Support System (IDSS) utilizing autonomous UAV technology. This innovative system aims to address the inefficiencies of traditional irrigation methods by providing precise, real-time assessments of the water needs of the plants and to do autonomous irrigation. Also it aims to create a base to make already trined libraries for different crop types and in different locations to make it available for large number of farmers. The system comprises a robust, lightweight au-tonomous UAV equipped with a low-cost standart camera, GPS, and IMU for data collection and an open source flight controller for autonomous navigation. In the field we have ARUCO markers used for planning and defining the irrigarion areas so under each mark-er we have an electrical water valve to receive the watering commands from the UAV. Each makers has a unique ID and is placed in postion where the UAV can easily detects using the camera that's in the UAV. So simply the UAV autonomously takes off, flies to the already spified markers location while it tries to find the markers starting with the maker that has the first ID. If it detects the marker, it tries to center it to the UAV center even if it doesn't reach the specified location of the marker. After the UAV centers the marker it lowers its altidude to 5 m and makes a circle around that area while it records a video and runs a real time water prediction algoriths. Then it continues to the next marker. The result of the water prediction algorithm is that it gives the quantities of plant that needs water and the ones that does not need water, then it gives the result based on the average of that area to give water order or not to the watering valve.

## 2.2. Hardware Setup

### *UAV Frame and Propulsion Systems*

The UAV frame is designed to be lightweight yet robust so carbon fiber tubes and plates was used, ensuring stability and durability. also we used PLA 3D printed material to make the cover , landing gears and some sensitive parts (Figure 2.) . It is designed as hexa copter so it is equipped with 6 high-efficiency DC type motors and propellers, electronic speed controllers (ESCs), and an open-source controller (pixhawks), a raspberry pi (used as a mini computer )and a camera. The propulsion system is calibrated to provide stable flight and precise maneuvering capabilities, essential for accurate data collection. To calculate the total time that the UAV can fly, the following formula can be used; $T_{flight}$=(Q/ $I_{flight\ load}$)*60/1000 .$T_{flight}$ the flight time in minutes. Q, is the LiPo battery capacity in mAh. $I_{flight\ load}$, the total ampers drawn by the motors and other electirical equipments in our case the camera and microproceesor in A. A lithum battary was used with a capacity of 6S 16000mah. And the full load is calculated as 50 A so the total flight time is calculated to be 19.2 min. Due to wind resistance in real life the total flight time was measured to be 16 min 48 s. The time needed for mapping the field of 3600 $m^2$ (length 60 m x width 60 m) can be calculated using the following formula Time = Distance / UAV speed. Also since it will do a circle around the defined points the time needed to make the circle can be calculated by the finding the circumference of the circle 2 $\pi$*radius. The radius of the circle is 15 m and the drone speed is 2.5 m/s. So the time is calculated to be 246.8 s = 4.11 min. Due to wind resistance and the UAV lowers its altitude before staring the circle also it fly higher again so the total measured time needed for the mapping is 6.35 min. The UAV is equipped with a variety of sensors, A high-resolution camera captures detailed aerial images of the agricultural fields also it used to detects marker for mapping of the irrigation area. Also it has a GPS module provides accurate geolocation data, and an inertial measurement unit (IMU) ensures flight stability and navigation accuracy.

### *Autonomous Operation and Control*

The UAV was programmed using python programming language and it implements some open source liabraries such as DroneKit. DroneKit is an open-source library for Python that allows developers to create custom applications for UAVs running the ArduPilot firmware. It provides a high-level API to control and manage the UAV's behavior, making it easier to implement autonomous flight operations [11]. The UAV operates autonomously, following predefined flight paths and executing tasks based on the system's analysis.

- *Flight Planning:* Predefined flight paths are set to ensure comprehensive coverage of the agricultural field. The UAV follows these paths to capture images and collect data.
- *ArUco Markers:* ArUco markers are placed strategically on the ground to aid in the creation of a precise land map and to identify specific areas needing irrigation. These markers, which are two-dimensional barcodes each with a special ID number, are easily detectable by the UAV's camera system (Figure 3.). When the UAV captures images of the field, the ArUco markers provide reference points that facilitate accurate mapping of the terrain. This method enhances the UAV's ability to navigate and precisely locate areas requiring irrigation. So after detecting theses markers the UAV centers the markers and lowers its altidude to 5 m. Then it makes a circle around that loaction while it performs the water need prediction algorithm. After it finishes it continues to the next marker.
- *MissionPlanner* is a ground control station software for the ArduPilot platform. It allows users to plan, simulate, and monitor UAV missions. MissionPlanner provides a graphical interface for setting waypoints, configuring flight parameters, and simulating missions in a virtual environment [12]. A lot of improvement, test and edits on the autonomous code was done using

the mission planner simulation software which shows all the detailed simulations by running the code that's the same as the one used in real UAV.

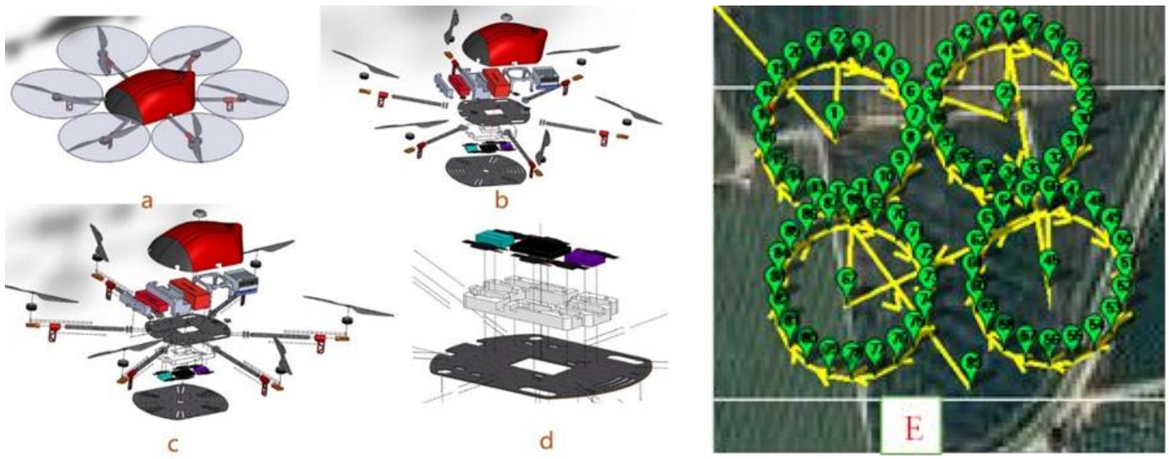



**Figure 2.** a) 3D Design of the UAV with the component's placements. b, c) The parts of the UAV. d) The placement of electronic components between two parts of the carbon fiber. e) waypoints movement of the UAV in the field. Points 1, 23, 45 and 62 are the center of the circles where the ARUCO markers are located

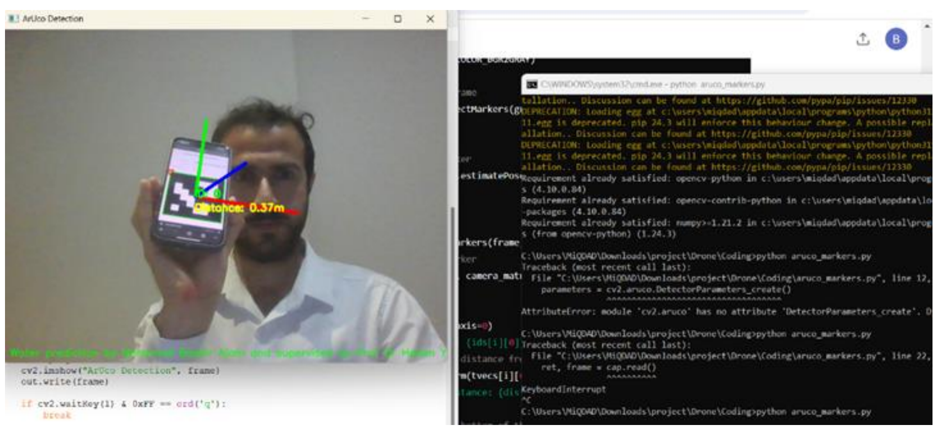



**Figure 3.** ARUCO marker detection algorithm tests and result

## 2.3. Software Setup

The development of the system requires a suite of software tools for image processing, machine learning, and data analysis. It analysis the image to detect if it has plant then it trains the system based on a lot of picures of test pepper crope kept at the lab and grouped in there dataset stressed pepper plant, less watered and good watered. After training the system it saves the training file and loads it into the prediction script to perform real time water need prediction.

### *Image Processing and Machine Learning*

The system employs advanced image processing techniques to preprocess each frame of the captured videos. These include noise reduction, geometric correction, and radiometric calibration to enhance image quality. The primary tools and libraries used are Python, OpenCV, TensorFlow, Keras, Scikit-learn, and Mission Planner. Each of these tools plays a critical role in ensuring the efficient operation and accuracy of the system. The processed images are then analyzed using machine learning models, such as convolutional neural networks (CNNs), to predict the water needs of plants. Python serves as the backbone, enabling the use of OpenCV for image processing, TensorFlow and Keras for machine learning, Scikit-learn for data analysis, and Mission Planner for mission planning and execution. This cohesive setup allows for efficient development, testing, and deployment of the system, ensuring accurate and reliable operation in real-world agricultural settings.

 https://doi.org/10.70395/cunas.1511336 Cukurova University Journal of Natural & Applied Sciences (2024) 3(2)

***Image processing techniques:*** By employing noise reduction, geometric correction, and radiometric calibration, the images are prepared for accurate feature extraction. Color analysis, texture analysis, and vegetation indices provide quantitative measures that are used to assess crop health and water needs. This cohesive approach allows for reliable and efficient irrigation management in agri-cultural settings. Each technique plays a vital role in ensuring the accuracy and reliability of the system.

- *Noise Reduction*; is used to improve the clarity and quality of the images by removing unwanted random variations (noise) by applying Gaussian Filtering and Median Filtering [13].
- *Geometric Correction*; is used to correct distortions caused by the camera's perspective, lens distortions, and UAV movement, ensuring accurate spatial representation of the image using Homography Transformation [14].
- *Radiometric Calibration*; is used to normalize pixel values, correcting for sensor irregularities and environmental conditions, ensuring consistent and accurate representation of the image's brightness and color [15].
- *Feature Extraction:*
  - *Color Analysis*; is used to detect signs of plant health and water stress by examining the color characteristics of the image. Healthy plants typically exhibit specific color ranges that can be quantitatively assessed by mean of Color Histograms [13].
  - *Texture Analysis*; to assess the texture of the crops, which can indicate water availability and health. Texture features such as roughness, smoothness, and regularity can indicate the presence of water stress or other issues. We use Gray Level Co-occurrence Matrix (GLCM) which measures the frequency of pairs of pixel values occurring at a specified spatial relationship. GLCM is used to extract texture features such as contrast, correlation, energy, and homogeneity.
- *Vegetation Indices*; assess plant health and vigor using indices derived from RGB imagery. These indices provide a measure of plant greenness, which correlates with chlorophyll content and overall plant health. Using Visible Atmospherically Resistant Index (VARI) which is an index derived from RGB imagery that provides a measure of plant greenness [16].

***Machine learning models:***

- *Deep Learning-Based Feature Extraction*: particularly Convolutional Neural Networks (CNNs), have revolutionized feature extraction by automatically learning hierarchical features from raw image data. CNNs consist of multiple layers, each ex-tracting increasingly abstract features. **Convolutional layers**; apply convolution operations to the input image, using learna-ble filters to capture spatial hierarchies. These layers detect edges, textures, patterns, and complex structures. **Pooling layers**; reduce the spatial dimensions of the feature maps, retaining the most salient features while reducing computational complexity. Common pooling methods include max pooling and average pooling [17,18].
- *TensorFlow* is an open-source machine learning framework developed by the Google Brain team. is used for training and deploying machine learning models. These models analyze processed images to predict the water needs of crops based on extracted features. Key Features: Supports deep learning and traditional machine learning algorithms.Flexible architecture allows deployment across a variety of platforms (CPUs, GPUs, TPUs).Extensive libraries and tools for building machine learning models.Strong community support and comprehensive documentation [19,20]. Key TensorFlow algorithms and techniques used in our system include:
  - **Transfer learning;** leverages pre-trained models (e.g., VGG16, ResNet) trained on large datasets like ImageNet. By fine-tuning these models on our specific dataset, we achieved high performance with limited data and computational resources [21].
  - **Fine-tuning;** is a process where the pre-trained model is further trained on the new dataset with a very low learning rate. This adjusts the weights of the pre-trained model slightly to better fit the new dataset while retaining the learned features from the large dataset [22].
  - **Data augmentation;** is a critical technique in machine learning, particularly in image processing, to increase the diversity of the training data without actually collecting new data. It involves creating new training examples by applying various transformations to the original images. This helps to improve the robustness and generalization capability of the model. Some techniques used; Rotating the image by a certain degree, Shifting the image along the horizontal (width) or vertical (height) axis, Applying a shearing transformation to the image, which tilts the image in a certain direction, Zooming in or out on the image and Flipping the image horizontally [23].

- *Keras;* is an open-source software library that provides a Python interface for artificial neural networks. It is used on top of TensorFlow to simplify the creation and training of complex machine learning models. Its user-friendly API allows for quick prototyping and experimentation with different neural network architectures. Integration with TensorFlow, making it a powerful and flexible tool for deep learning [24].
- *Scikit-learn;* provides simple and efficient tools for data mining and data analysis. It is used for tasks such as feature extraction, data preprocessing, and the training of machine learning models. Built on NumPy, SciPy, and matplotlib. Open-source and commercially usable under the BSD license [25].

### Model Training and Validation

The water prediction model was trained on a dataset of annotated plant images, with labels indicating the water needs, three sets of pepper plants was used (Figure 4.) . Each set is labeled with relative training set names: "no-water," "less-water," and "good-water." Each plant is carefully watered to ensure accurate labeling. Images are collected using digital cameras or smartphones. More than 400 pictures were captured from different angles, distances and lightening then grouped for training the system. As a result of the training we got the training dataset file that we upload into the prediction code.  The dataset was split into training and validation sets, and the model's accuracy was evaluated on the validation set.

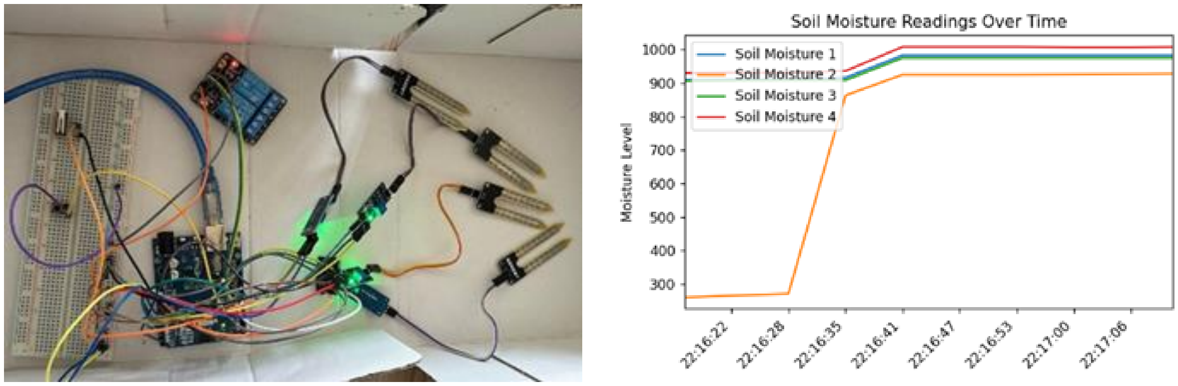
a) b)

**Figure 4.** a) Pepper crops set up in our lab, b) Pepper crops field in Adana city in Turkey

To ensure the accuracy and reliability of the system, the output results from the image analysis are validated by comparing them with in-ground Arduino-based soil moisture sensors data (Figure 5.). Where moisture content and other relevant data have been recorded and analyzed. This robust benchmarking process provides a reliable measure of the system's performance.

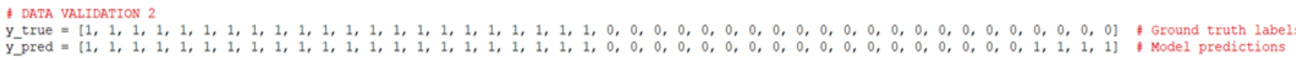
This setup provided accurate and real-time data on the soil moisture levels, which served as the ground truth for the validation process. Also this set automatically water the plant to insure stable pilot plant at the set moisture values all the time.

The predicted water need results are compared with measurements from in-ground sensors to validate the accuracy of the image analysis and machine learning models (Figure 6.). The matrix of the results was created which can show true positive, true negative, false positive and false negative. From that values the system performance interms of, the accuracy, precision, recall, F1 Score is determined.




a) b)

**Figure 5.** a) Arduino and soil moisture sensors as ground truth set up, b) Soil moisture readings



```
# DATA VALIDATION 2
y_true = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  # Ground truth labels
y_pred = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1]  # Model predictions
```

**Figure 6.** Validation matrix that compares the prediction results with the ground truth

## 3. Results and Discussion

As a result two main outputs is optained of this research. The first one is a fully functioning autonomous UAV integrated with water prediction algorithms. The Second output, a base for a user friendly website for water prediction algorithms integrated with AI was developed where a user can upload a video or a picture and it gives the water need predictions with the advices generated with open-AI based on the location and on the crop type.

Evaluation metrics are essential to assess the performance of a water prediction model. They provide a quantitative basis to compare the model's predictions against the actual ground truth data. Here we outline the key evaluation metrics: Accuracy; The proportion of correct predictions (both positive and negative) out of the total predictions made., Precision; The proportion of true positive predic-tions out of all positive predictions made, Recall; The proportion of true positive predictions out of all actual positives in the dataset, and F1 Score; The harmonic mean of precision and recall, providing a balanced measure of accuracy. For the evulation we used 44 test datas, we got the truth ground data gives 22 plants don't need water and 22 plants are stressed and needs water. With an im-proved training the prediction system gave 26 plants don't need water and 18 plants are stressed. With an individual accuracy up to 96.8 % in the ones that doesn't need water and up to 95.99%. in the ones that needs water.


a) b)

**Figure 7.** a) Prediction of plant that doesn't need water with an accuracy of 96.8% and b) Plant needs water with accuracy of 95.99%

Overall for the validation:

**Table 1.** Performance measurement

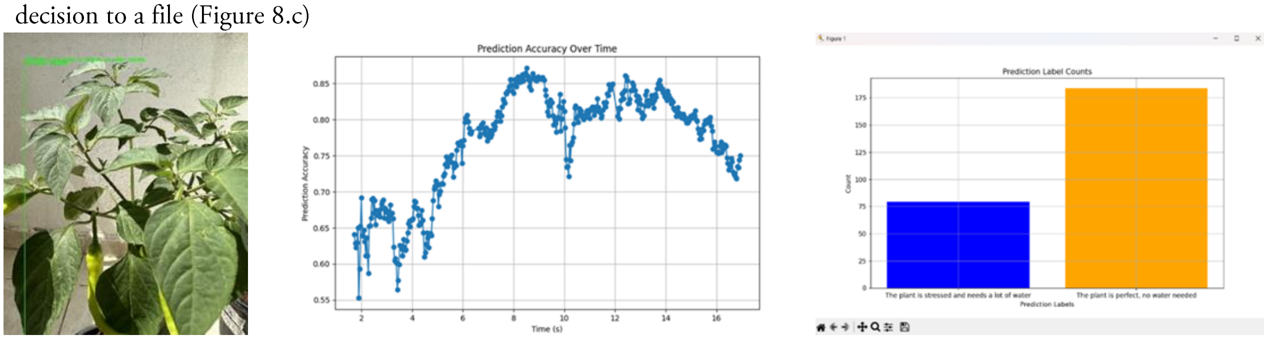
| | |
|---|---|
| Accuracy | 91 |
| Precision | 85 |
| Recall | 100 |
| F1 score | 92 |

**Table 2.** The test data

| | |
|---|---|
| Total tests | 44 |
| True positive | 22 |
| True negative | 18 |
| False positive | 4 |
| False negative | 0 |

After training the system with the data sets, a new picture was uploaded and in this case the system uses the live video that's being recorded in the UAV to do the water prediction of the relative irrigation area. The logic of the prediction script is as follows:

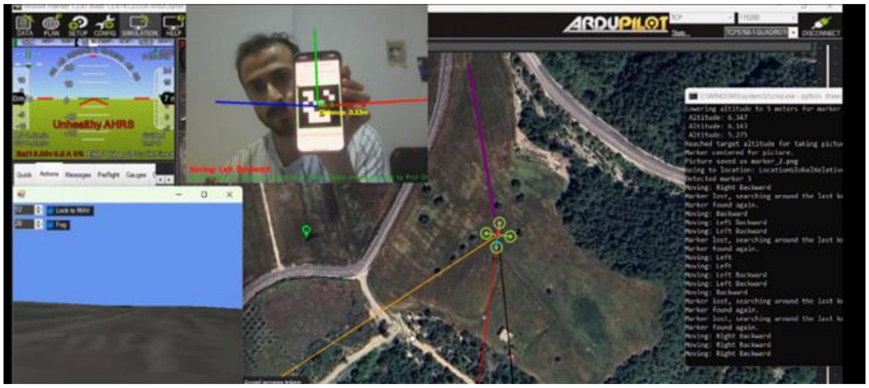
- Video Processing: The script reads the video frame by frame.
- Leaf Detection: Each frame is checked to see if it contains significant green areas (leaves).
- Preprocessing: Frames containing leaves are preprocessed and normalized.
- Prediction: The preprocessed frames are fed into the model to predict whether the plants in the frame need water.
- Annotation: Predicted areas are annotated on the video frames with rectangles and text (Figure 8.a).
- Saving: The annotated frames are saved into a new video with a watermark.
- Plotting: The script plots prediction accuracy over time and the count of prediction labels (Figure 8.b).
- Final Decision: The script determines whether the area needs water based on the most frequent prediction label and saves this decision to a file (Figure 8.c)



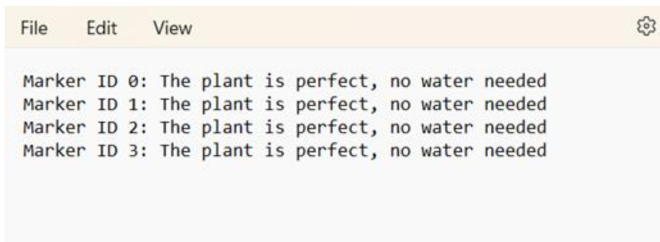

a) b) c)

**Figure 8.** a) The resultant plant with a rectangle showing the plant and the prediction result, b) The accuracy graph over time results, c) The predic-tion result showing the plants that needs water and the ones that doesn't need water

Fully functioning autonomous UAV code successfully integrated with the water prediction algorithm is obtained. As a result it gives the average predicted water need of the irrigations areas which is classified according to the ARUCO marker ID number. So each marker is set as one indepent irrigation area that takes irrigation commands seperatly (Figure 9.b). The autonomous flight code runs successfully in the simulation and in real life. In the simulation it shows the UAV GPS movements,commands sent to the UAV by the code, camera output and the in flight control (Figure 9.a). In the real life we can see the same as the simulation environment when we connect through mission planner software (Figure 10).




a)



File Edit View

```
Marker ID 0: The plant is perfect, no water needed
Marker ID 1: The plant is perfect, no water needed
Marker ID 2: The plant is perfect, no water needed
Marker ID 3: The plant is perfect, no water needed
```

b)

**Figure 9.** a) The full simulation of the autonomous code, b) The prediction code output

a)

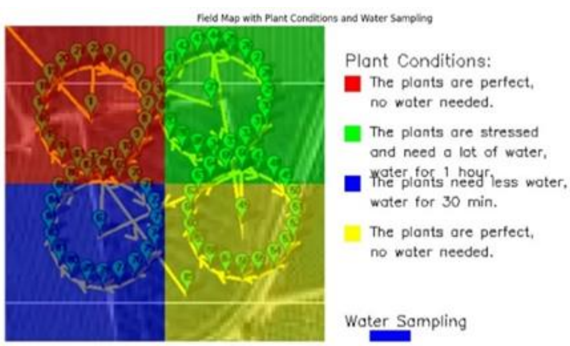
b)

**Figure 10.** a) The flight test of the real UAV, b) The real UAV after combining all the parts

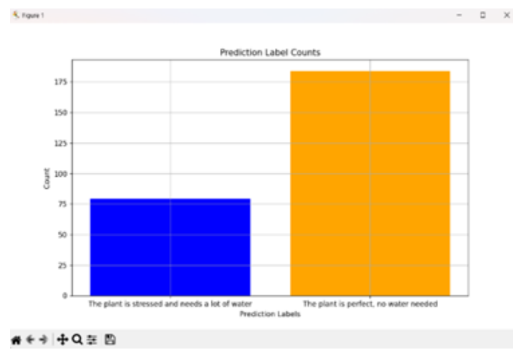
## 4. Conclusion

This study successfully developed and validated an autonomous UAV-based irrigation decision support system (IDSS) leveraging advanced image processing and machine learning techniques. Addressing the inefficiencies of traditional irrigation methods, the system utilizes low-cost, high-resolution imaging to provide precise, real-time assessments of plant water needs. The integration of Python, OpenCV, TensorFlow, Keras, and Scikit-learn enabled the creation of a robust solution that optimizes water usage and reduces labor costs.

Field tests on pepper crops demonstrated the system's effectiveness, enhancing water use efficiency and improving crop yields. The system was tested on a 3600 $m^2$ field of pepper crops. Traditional irrigation methods involved uniform watering schedules, the water consumption was measured the average to be approximately 11.5 liters per square meter per week. In total the average of this field consumed 41520 liters per week continuesly since it was uniform waterig schuales so it led to over watering of some parts of the field and under watering some as a result is consumes a lot of water and effects the crop yields. While the developed system ensures water is applied only where and when needed (Figure 11.). So for this field was devided it into 4 parts each one defined with an ARUCO marker and having an automatic watering valve which is opend and closed to a specific time and that received waterig commands from the UAV. The water consumption was measured it showed that the average consumption of the total field was 7.9 liters per week per square with a total of 28440 liters per week even in some weeks it was lower (Figure 12). The average amout of water saved is 41520 – 28440 = 13080 liters/week. So from these values the average water saving percentage can be calculated; ((13080 liters/week)/(41520 liters/week))*100 = 31.5% water saved.As a result of precious watering the crop yield was improved due to avoiding of over watering and under watering. The UAV captures high-resolution images, processes them to identify irrigation needs, and autono-mously controls water valves to manage irrigation. This approach not only conserves water but also reduces operational costs.

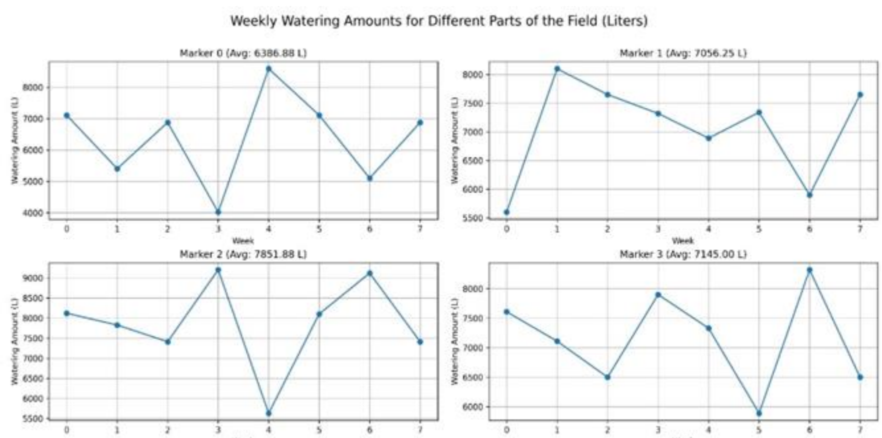



a)

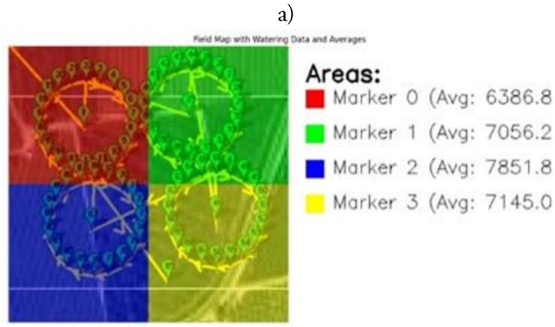



b)

**Figure 11.** a) Irrigation application map obtained after each field mission, b) Graph generated after scaning each area showing the number of frames detected that need water and the numbers that doesn't need water, then it gives the average of that area




a)




b)

**Figure 12.** a) Weekly watering amount of different parts of the field, b) Field map with watering data and divided sections

The results indicate that the autonomous UAV-based system can significantly reduce water consumption while maintaining or enhancing crop productivity. In the test field set up, the field was divided into four equal irrigation areas and as a result of implementing the system the water consumption was reduced up to 31.5% which can be further improved by dividing the field into more than 4, for example 8 areas. By integrating technologies like TensorFlow for machine learning and autonomous navigation, the system repre-sents a significant advancement in precision agriculture. Future work will expand the system's capabilities to include more environ-mental factors and a broader range of crops and agricultural settings. Also to train the system on different crope types and locations and to allow the user train the system in their own crop type and farm then to create liabraires of it for fast reach and combine it with AI for more user friendly approaches. Additionally, further improvements in machine learning models and image processing techniques will enhance the system's accuracy and reliability. The potential for integrating additional sensors and IoT devices to monitor various envi-ronmental parameters will also be explored, further advancing the precision and effectiveness of the irrigation decision support system.

Overall, this development marks a significant step forward in precision agriculture. By leveraging cutting-edge technologies to optimize water usage and improve crop yields, the system not only contributes to sustainable agricultural practices but also offers a scalable solution to address the growing global demand for food and water resources.

## References

[1] [DSI (2020). General Directorate of State Hydraulic Works. Retrieved from https://www.dsi.gov.tr/

[2] Based on data from AQUASTAT (n.d.a); [Mateo-Sagasta et al. (2015)]; and [Shiklomanav(1999)]; Contributed by [Sara Marjani Zadeh(FAO)

[3] Ali, S., Sultan, M., Ahmad, F., Majeed, F., Ahamed, M.S., Aziz, M., Shamshiri, R.R., Sajjad, U., Khan, M.U., Farooq, M. (2023). An Overview of Smart Irrigation Management for Improving Water Productivity under Climate Change in Drylands. Agronomy; 13(8): 2113. DOI: 10.3390/agronomy13082113.

[4] Dong, Y. (2023). Irrigation Scheduling Methods: Overview and Recent Advances. IntechOpen. DOI: 10.5772/intechopen.100633.

[5] Rasmussen, J., Jacobsen, L., Bergmann, T., & Bunk, K. (2023). How much is enough in watering plants? State-of-the-art in irriga-tion control: Advances, challenges, and opportunities with respect to precision irrigation. Frontiers in Sustainable Food Systems; 7: 77. DOI: 10.3389/fsufs.2023.00077.

[6] Hunt, E.R., Hively, W.D., Fujikawa, S.J., Linden, D.S., Daughtry, C.S.T., McCarty, G.W. (2018). Acquisition of NIR-Green-Blue Digital Photographs from Unmanned Aircraft for Crop Monitoring. Remote Sensing; 8(1): 111. DOI: 10.3390/rs8010111.

[7] IntechOpen. (2023). Recent Advances in Irrigation and Drainage. DOI: 10.5772/intechopen.100759.

[8] Lillesand, T.M., Kiefer, R.W., Chipman, J.W. (2015). Remote Sensing and Image Interpretation (7th ed.). Wiley, Hoboken.

[9] Smith, R.J., Baillie, J.N. (2009). Precision irrigation: challenges and opportunities. Agricultural Water Management; 96(6): 835-839. DOI: 10.1016/j.agwat.2009.02.011.

[10] Zhang, C., Kovacs, J.M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. Precision Agriculture; 13(6): 693-712. DOI: 10.1007/s11119-012-9274-5.

[11] DroneKit-Python (https://github.com/dronekit/dronekit-python/)

[12] Mission planner software (https://ardupilot.org/planner/docs/mission-planner-overview.html)

[13] Gonzalez, R.C., Woods, R.E. (2002). Digital Image Processing (2nd ed.). Prentice Hall, Upper Saddle River.

[14] Hartley, R., Zisserman, A. (2004). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge University Press, Cambridge.

[15] Schott, J.R. (2007). Remote Sensing: The Image Chain Approach (2nd ed.). Oxford University Press, New York.

[16] Gitelson, A.A., Kaufman, Y.J., Stark, R., Rundquist, D. (2002). Novel algorithms for remote estimation of vegetation fraction. Remote Sensing of Environment; 80(1): 76-87. DOI: 10.1016/S0034-4257(01)00289-9.

[17] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature; 521(7553): 436-444. DOI: 10.1038/nature14539.

[18] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems; 25: 1097-1105. DOI: 10.1145/3065386.

[19] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265-283). DOI: 10.5555/3026877.3026899.

[20] TensorFlow. (n.d.). Retrieved from https://www.tensorflow.org/

[21] Pan, S.J., Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering; 22(10): 1345-1359. DOI: 10.1109/TKDE.2009.191.

[22] Yosinski, J., Clune, J., Bengio, Y., Lipson, H. (2014). How transferable are features in deep neural networks? Advances in Neural Information Processing Systems; 27: 3320-3328. DOI: 10.5555/2969033.2969197.

[23] Shorten, C., Khoshgoftaar, T.M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data; 6(1): 1-48. DOI: 10.1186/s40537-019-0197-0.

[24] Chollet, F. (2015). Keras: Deep Learning for humans. Retrieved from https://keras.io/

[25] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research; 12: 2825-2830.