

University of Arkansas, Fayetteville

ScholarWorks@UARK

Computer Science and Computer Engineering
Undergraduate Honors Theses

Computer Science and Computer Engineering

5-2023

Smart-Insect Monitoring System Integration and Interaction via AI Cloud Deployment and GPT

Ahmed Moustafa

Follow this and additional works at: <https://scholarworks.uark.edu/csceuht>



Part of the [Agricultural Education Commons](#), and the [Computational Engineering Commons](#)

Citation

Moustafa, A. (2023). Smart-Insect Monitoring System Integration and Interaction via AI Cloud Deployment and GPT. *Computer Science and Computer Engineering Undergraduate Honors Theses* Retrieved from <https://scholarworks.uark.edu/csceuht/127>

This Thesis is brought to you for free and open access by the Computer Science and Computer Engineering at ScholarWorks@UARK. It has been accepted for inclusion in Computer Science and Computer Engineering Undergraduate Honors Theses by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu.

Smart-Insect Monitoring System Integration and Interaction via
AI Cloud Deployment and GPT

Smart-Insect Monitoring System Integration and Interaction via
AI Cloud Deployment and GPT

A thesis submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Science

By

Ahmed Moustafa
University of Arkansas
Bachelor of Science in Computer Science, 2023

May 2023
University of Arkansas

This thesis is approved for recommendation to the Graduate Council

Prof. Khoa Luu
Thesis Director

Prof. Alexander Nelson
Committee member

Prof. John Gauch
Committee member

Abstract

The Insect Detection Server was developed to explore the deployment and integration of an Artificial Intelligence model for Computer Vision in the context of insect detection. The model was developed to accurately identify insects from images taken by camera systems installed on farms. The goal is to integrate the model into an easily accessible, cloud-based application that allows farmers to analyze automatically uploaded images containing groups of insects found on their farms. The application returns the bounding boxes and the detected classes of insects whenever an image is captured on-site, enabling farmers to take appropriate actions to address the issue of the insects' presence. To extend the capabilities of the application, the server is linked to a GPT-3.5 API. This will allow users to ask questions about the bugs detected on their farms, creating an "online expert"-like feature. Python, C++, and Computer Vision libraries were used for the detection model, while the OpenAI API was used for GPT-3.5's integration. By combining these technologies, farmers can more effectively and efficiently manage pests on their farms than current alternatives. This Generative Pre-trained Transformer (GPT) aspect of the project can be leveraged to enable the emulation of agricultural experts for users/farmers. The large language model (LLM) neural network can be fine-tuned using prompt engineering to generate natural language responses to user queries. This will enable farmers to get expert advice and guidance on pest management without having to consult with a human expert. The integration of GPT-3.5 API will also allow the application to provide personalized recommendations based on each farm's specific needs and circumstances. This added feature will give the farmers a more comprehensive and tailored approach to pest management, further increasing the efficiency and effectiveness of their pest control strategies. The significance of this research lies in the development of a practical and accessible tool for farmers to manage pests on their farms. Using Computer Vision and Artificial Intelligence, farmers can quickly and accurately identify insects, leading to more efficient and effective pest management. This could help farmers reduce the use of pesticides and other forms of pest management, leading

to improved crop yields and reduced environmental impacts. The potential benefits of this technology extend beyond the agricultural industry, as the techniques used in this research can be applied to a wide range of computer vision and user-facing data analytic applications. For example, the developed techniques could be applied to other fields, such as surveillance, security, and medical imaging.

THESIS DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed_____

Ahmed Moustafa

Refused_____

Ahmed Moustafa

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to several individuals and institutions who have played a crucial role in the completion of this thesis. First and foremost, I would like to thank Dr. Khoa Luu, my honors research mentor, for his invaluable guidance, support, and mentorship throughout the entire research process. Dr. Luu's expertise, insights, and constructive feedback have been instrumental in shaping this work and in helping me develop as a researcher. I would also like to extend my thanks to Thanh-Dat Truong, the graduate student and fellow researcher in the Computer Vision and Image Understanding (CVIU) lab who generously shared his time and expertise to help with the model and project as a whole. Dat's contributions were essential to the success of this research, and I am very grateful for his support. I am also indebted to the University of Arkansas for providing the resources and opportunities necessary to facilitate this Undergraduate Honors Research. The university's commitment to fostering undergraduate research has been instrumental in my academic and professional development, and I am honored to have been a part of this program. Finally, I would like to thank my thesis defense committee members, Dr. Khoa Luu, Dr. John Gauch, and Dr. Alexander Nelson, for their time and expertise.

TABLE OF CONTENTS

Abstract	ii
Acknowledgements	v
Table of Contents	vi
List of Figures	viii
1 Introduction	1
2 Related Works	6
2.1 Automated Farm Management	7
2.2 Detection and Classification	8
2.3 Support Tools	9
3 Background	11
3.1 Transformers	11
3.2 Computer Vision Model	13
3.2.1 Deep Learning	13
3.2.2 Insect Detection and Classification Models	14
3.3 Generative Pretrained Transformer (GPT)	15
3.3.1 Prompt Engineering	16
4 Implementation	18
4.1 Backend	19
4.1.1 Database	19
4.1.2 API	21
4.2 Frontend	26
4.3 Artificial Intelligence Chat Feature	28
5 Conclusion	32
6 Publications	34
Bibliography	35
A Broader Impacts of Research	40
A.1 Applicability of Research Methods to Other Problems	40
A.2 Impact of Research Results on U.S. and Global Society	41
A.3 Impact of Research Results on the Environment	41

B	Identification of All Software Used in Research and Thesis/Dissertation Generation	43
C	Plagiarism Check	44

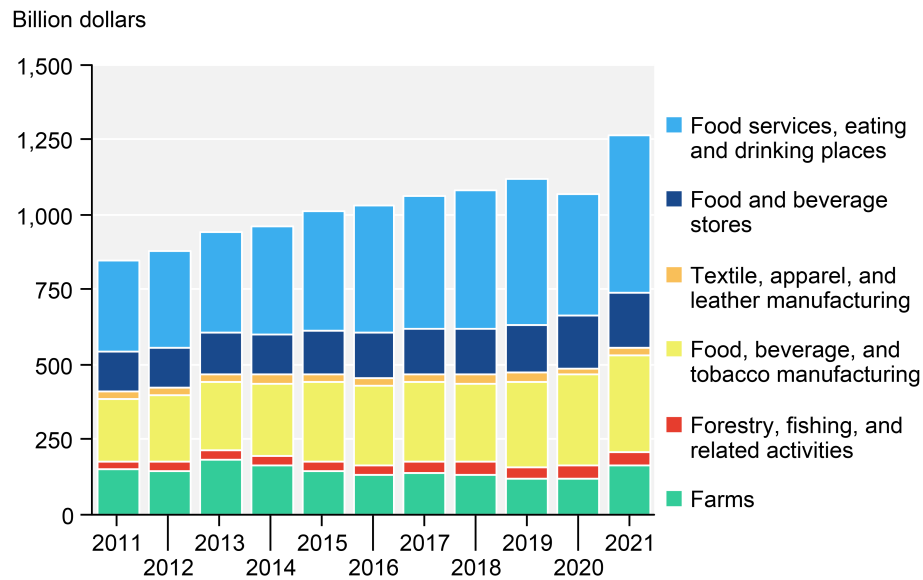
LIST OF FIGURES

Figure 1.1:	Chart displaying the value added to U.S. GDP by Agriculture and related industries, 2011-21. Farms alone exceed \$150 billion annually.	1
Figure 1.2:	Chart displaying the rise and fall of pesticide usage in the U.S. with a peak in 1981, 1960-2008.	2
Figure 1.3:	Bed Bug Algorithm: small-scale Integrated Pest Management (IPM) plan.	4
Figure 2.1:	Hexacopter spraying pesticides in Tamil Nadu Agricultural University Farm rice fields in Tamil Nadu, India, during October 2019.	7
Figure 3.1:	The Transformer - model architecture [1].	12
Figure 3.2:	The Vision Transformer - model architecture [2].	14
Figure 3.3:	Bug system transformer diagram.	15
Figure 3.4:	Chart of No. of Parameters for multiple LLMs [3].	16
Figure 4.1:	TypeScript interface structures for the Detection and Detection Instance object used in the JSON payload and database schema.	20
Figure 4.2:	Cloud FireStore Database data showcasing the detections collection and its child document structure. The document structure matches the interfaces from Fig 4.1.	21
Figure 4.3:	JSON response example for a GET request from the frontend, returning all the detection data for a specific day.	22
Figure 4.4:	JSON payload example for a POST request from the on-site machine.	23
Figure 4.5:	Function for handling a GET request to fetch the latest detection data from the database.	24
Figure 4.6:	Function for saving new detections to the database.	25
Figure 4.7:	The frontend user interface showcasing the pie chart with tooltip functionality and the image carousel for individual detections.	27
Figure 4.8:	The chat modal UI with an example conversation about pest-related questions.	29
Figure 4.9:	Example conversation where the AI assistant exhibits narrowed scope.	30
Figure 4.10:	Example messages parameter of payload for an API call to OpenAI's GPT-3.5 model with specified expertise in agriculture and insects.	31

1 Introduction

The U.S. agricultural industry has long been a cornerstone of the country's economy, contributing over a trillion dollars in value-added to the GDP annually. Specifically, farms generate approximately \$164.7 billion annually, as illustrated in Fig 1.1 [4, 5]. One of the significant challenges farmers face is managing pests and crops, with the average cost being approximately “34% of a farmer's variable crop production costs” [6]. While pests can cause economic loss and harm to farmland, there are also beneficial insect species that save growers millions of dollars annually [7].

Value added to U.S. GDP by agriculture and related industries, 2011–21

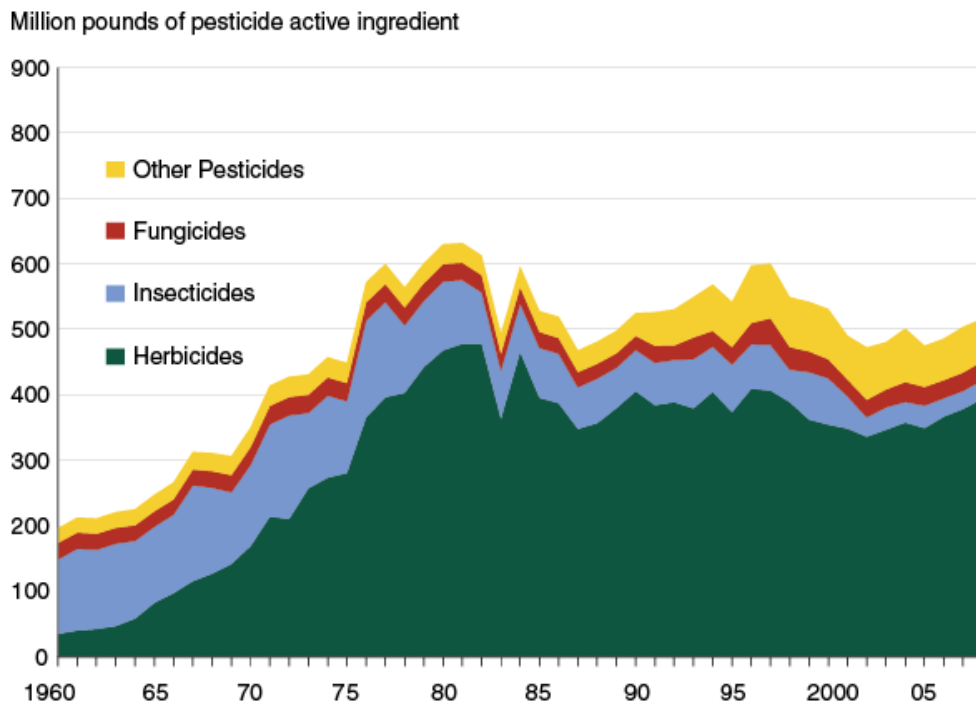


Note: GDP = Gross domestic product.
Source: USDA, Economic Research Service using data from U.S. Department of Commerce, Bureau of Economic Analysis, Value Added by Industry, as of December 22, 2022.

Figure 1.1: Chart displaying the value added to U.S. GDP by Agriculture and related industries, 2011-21. Farms alone exceed \$150 billion annually.

To account for rising number of pests and invasive species over the past century, farmers have increased their pesticide use to match, as seen in Fig 1.2 [8]. Besides increased costs, the consequences of this rise in pesticide usage include unintentional environmental and human harm due to runoff and unintended targets [9]. Another risk related to classic pesticide usage is the development of insecticide resistance through surviving insect populations and mutations [10]. With this many risks associated with traditional pesticide use, adding more quantity can lead to diminishing returns on pesticide efficacy and even negative returns in extreme cases.

Pesticide use in U.S. agriculture peaked in 1981 (21 selected crops, 1960 -2008)



Source: USDA, Economic Research Service using USDA, National Agricultural Statistics Service and proprietary data.

Figure 1.2: Chart displaying the rise and fall of pesticide usage in the U.S. with a peak in 1981, 1960-2008.

To combat the rise in pesticides we see in Fig 1.2 due to classic pest control methodologies, farmers began planting new pest-resistant strains of crops and implemented Integrated Pest Management (IPM). IPM is a pest control strategy that prioritizes long-term prevention through environmentally considerate methods. These methods include manipulating the pests' habitat, replacing possibly outdated practices by the farmer, planting resistant crop varieties, etc. Though there are many different IPM practices, all IPM programs contain the same six general components [11]:

1. Pest identification
2. Monitoring and assessing pest numbers and damage
3. Guidelines for when management action is needed
4. Preventing pest problems
5. Using a combination of biological, cultural, physical/mechanical and chemical management tools
6. After action is taken, assessing the effect of pest management

IPM is becoming increasingly adopted among farmers to reduce costs and yield losses. A concise version of an IPM is illustrated in Fig 1.3 [12]. The limitation of many IPMs is their labor constraints, as the identification and monitoring of pests require consistent manual sampling, inspection, and expertise regarding insects [13]. A novel system combining modern hardware and software has been developed to address these challenges to create a consistent and efficient pest identification and monitoring method.

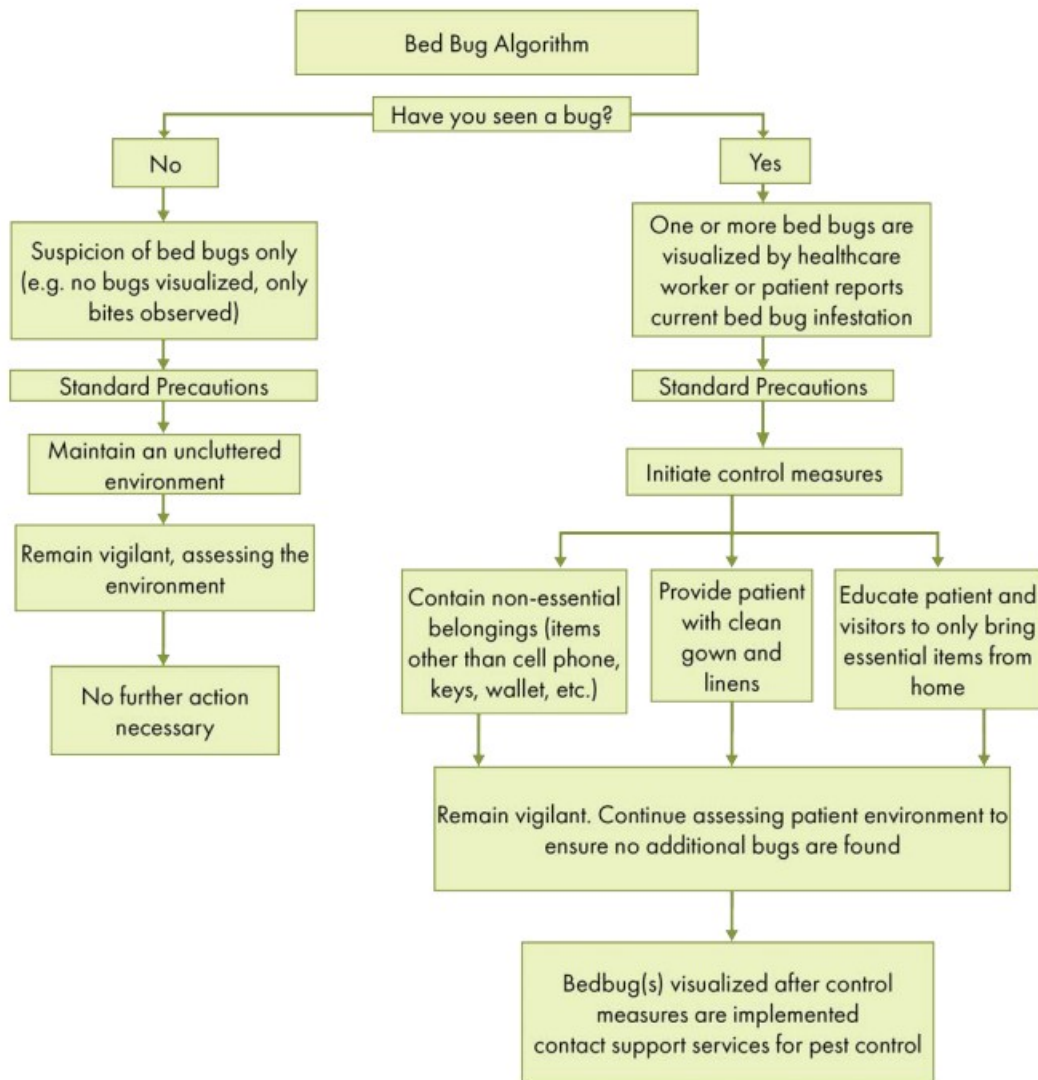


Figure 1.3: Bed Bug Algorithm: small-scale Integrated Pest Management (IPM) plan.

Integrating an on-site camera system running an insect detection and classification model with a user-facing application can revolutionize farmers' management of pests. By leveraging Artificial Intelligence, farmers can take informed action and track pest numbers in real-time, thereby improving the sustainability of IPM plans and increasing the efficiency of pest management. Furthermore, the

system is complemented by the integration of an additional AI model that is used to emulate an agriculture expert: OpenAI's GPT-3.5. This feature provides farmers with personalized recommendations based on their unique farm conditions, allowing for faster and more effective pest control strategies. This innovative approach can potentially reduce the use of harmful pesticides and improve crop yields by a significant amount.

2 Related Works

Currently, a majority of farmers and insect researchers manually sample insects regularly. Since many insects and mites can be tiny, the samplers must carry tools like digital cameras and 10x magnifying glasses. This monitoring procedure will include traps, beating counts, fecal pellet collection, and more [14]. Though these procedures are successful, maintaining the sampling schedule is troublesome since the researcher/farmer must use different techniques and traps for various insects. For example, to detect and monitor populations of alfalfa weevils, armyworms, and leafhoppers, the recommended guidelines are to do a sweep-net sample twice a week or more [15]. Moreover, manual sampling can be prone to errors and biases, resulting in inaccurate insect population estimates. Researchers or farmers may miss small insects or misidentify them, leading to incorrect population counts. Additionally, the accuracy of the estimates can be affected by the sampling frequency, location, and timing, as well as the experience of the sampler. These manual sampling limitations can result in delays and errors in the Integrated Pest Management plan.

The Agricultural industry is known for its innovation as farming has improved throughout history for the sake of efficiency. Original manual labor was replaced with machines that increased harvest yields and decreased time spent on many farming processes, including planting, maintenance, and harvesting. Recent advancements in robotic agricultural technology have resulted in the creation of automated harvesting systems that use Deep Learning, sensor-map fields, etc., to replace some of the many manual aspects of agriculture [16]. Although this principle of automation and technological advancement is exceptionally efficient, it often needs to account for issues like pest control which may be external to the base processes like seeding, growing, and harvesting of arable land.

2.1 Automated Farm Management

Automated farm management systems have expanded their scope to encompass pest monitoring and control, increasing efficiency and accuracy. For instance, Zhang et al. [17] developed a system that uses deep learning to identify and count insects on crops within a greenhouse. The system utilizes cameras to capture images and processes them with a convolutional neural network, providing real-time pest data to farmers. This method has the advantage of offering continuous monitoring but may be limited by factors such as camera resolution and lighting conditions. Similarly, an automated pest detection system for crops using unmanned aerial vehicles (UAVs) equipped with cameras and a machine learning-based classification algorithm was proposed, as shown in Fig 2.1 [18, 19]. This approach enables large-scale monitoring and rapid response to pest infestations but may be affected by environmental factors such as wind and weather.



Figure 2.1: Hexacopter spraying pesticides in Tamil Nadu Agricultural University Farm rice fields in Tamil Nadu, India, during October 2019.

Additionally, “Internet of Things” (IoT) based systems have been employed to improve pest management. Qazi et al. [20] mention examples of IoT-based smart agriculture systems that integrate pest monitoring with other farm management tasks, such as soil and climate monitoring. By consolidating multiple aspects of farm management, these systems enable an increase in informed decision-making and rapid interventions. However, implementing IoT systems may require significant upfront investment and technical expertise, which may be a barrier for small-scale farmers.

Recent advancements in remote sensing technology have also been employed in pest detection and monitoring. There is an example where remote sensing was used to detect insects in agriculture, focusing on the spectral characteristics of insect sounds [21]. This approach can help identify and localize insects in the field but might be limited by the complexity of the acoustic environment and the need for extensive insect sound libraries.

2.2 Detection and Classification

Recent advances in AI and computer vision have facilitated the development of more sophisticated detection and classification methods for insect monitoring. There are examples of efficient deep learning-based approaches being used for detecting and classifying insects through convolutional neural networks (CNNs) [22]. This method accurately recognizes various insect species, including pests and beneficial insects. However, the performance of such models may be limited by the quality and diversity of the training data, which could lead to misclassifications.

Thenmozhi et al. [23] developed an automatic recognition system for insect pests using image processing techniques and machine learning classifiers. This approach achieved high recognition rates but required manual preprocessing of images for feature extraction, which could be time-consuming and labor-intensive.

2.3 Support Tools

The emergence of AI-driven support tools has provided farmers and researchers with valuable resources for making educated choices in pest control. Navarro-Hellin et al. [24] created a decision support system that integrates pest monitoring information, meteorological data, and crop growth simulations to deliver customized pesticide application guidance. This system encourages eco-friendly pest management by reducing pesticide use and optimizing crop production.

A pest risk prediction system based on deep learning and remote sensing data was proposed by Chen et al. [25]. The system predicts the likelihood and severity of pest outbreaks, allowing farmers to implement preventative actions and reduce the consequences of pest infestations. Nevertheless, the precision of these models may be affected by the quality and promptness of input data and the intricacy of the factors that contribute to pest behavior.

Mimicking expert knowledge and decision-making has been an area of interest in research and development across various sectors. Expert systems have been created to help users with tasks that would usually necessitate the skills of a professional, such as medical diagnoses, financial planning, and legal counsel. These systems often employ AI, machine learning, and natural language processing to enable expert-like decision-making and deliver tailored recommendations [26, 27].

Expert emulation has been applied to various areas, including pest management in agricultural settings. Gonzalez-Andujar [28], for example, designed an expert system to estimate the risk of insect pests in olive crops. This system uses specialized knowledge to offer personalized pest management advice based on user-provided information, like plant species and geographic location. Similarly, an expert system for assessing pest risk in organic agriculture was created that assists farmers in recognizing potential hazards and executing suitable pest management tactics [29].

In addition to conventional expert systems, chatbots have been utilized to

replicate expert guidance and offer support in different industries. For instance, Agil-Ifdillah et al. [30] created a chatbot to help farmers identify crop diseases and pests while providing expert suggestions on prevention and treatment methods. This approach enables farmers to access expert knowledge in a more accessible and user-friendly way, lowering the obstacles to adopting sophisticated pest management techniques.

3 Background

Artificial Intelligence has advanced at a remarkable rate recently, with Deep Learning models achieving state-of-the-art performance in various tasks. These models, which include Transformers, Convolutional Neural Networks (CNNs), and Generative Pretrained Transformers (GPTs), have been applied to a wide range of applications, including Natural Language Processing (NLP) and Computer Vision. This chapter provides an overview of the critical concepts and architectures underpinning these models and their applications in the on-site insect detection and classification system. The chapter also discusses prompt engineering, a technique used to optimize the performance of language models like GPT.

3.1 Transformers

Transformers are a type of Deep Learning Neural Network architecture [1]. They have become the foundation for many state-of-the-art NLP and Computer Vision models. The primary value of the new transformer architecture is its “self-attention mechanism,” which enables selection across multiple inputs based on importance when processing each element. This mechanism enables transformers to effectively capture long-range dependencies and relationships in the input data.

Transformers have been widely used for machine translation, language modeling, and text classification. The Transformer architecture has an encoder and decoder component, each composed of multiple layers of self-attention and feed-forward neural networks. The encoder takes in the input sequence for processing, and the decoder creates the output sequence. The generic model architecture of a Transformer developed by Google is shown in Fig 3.1.

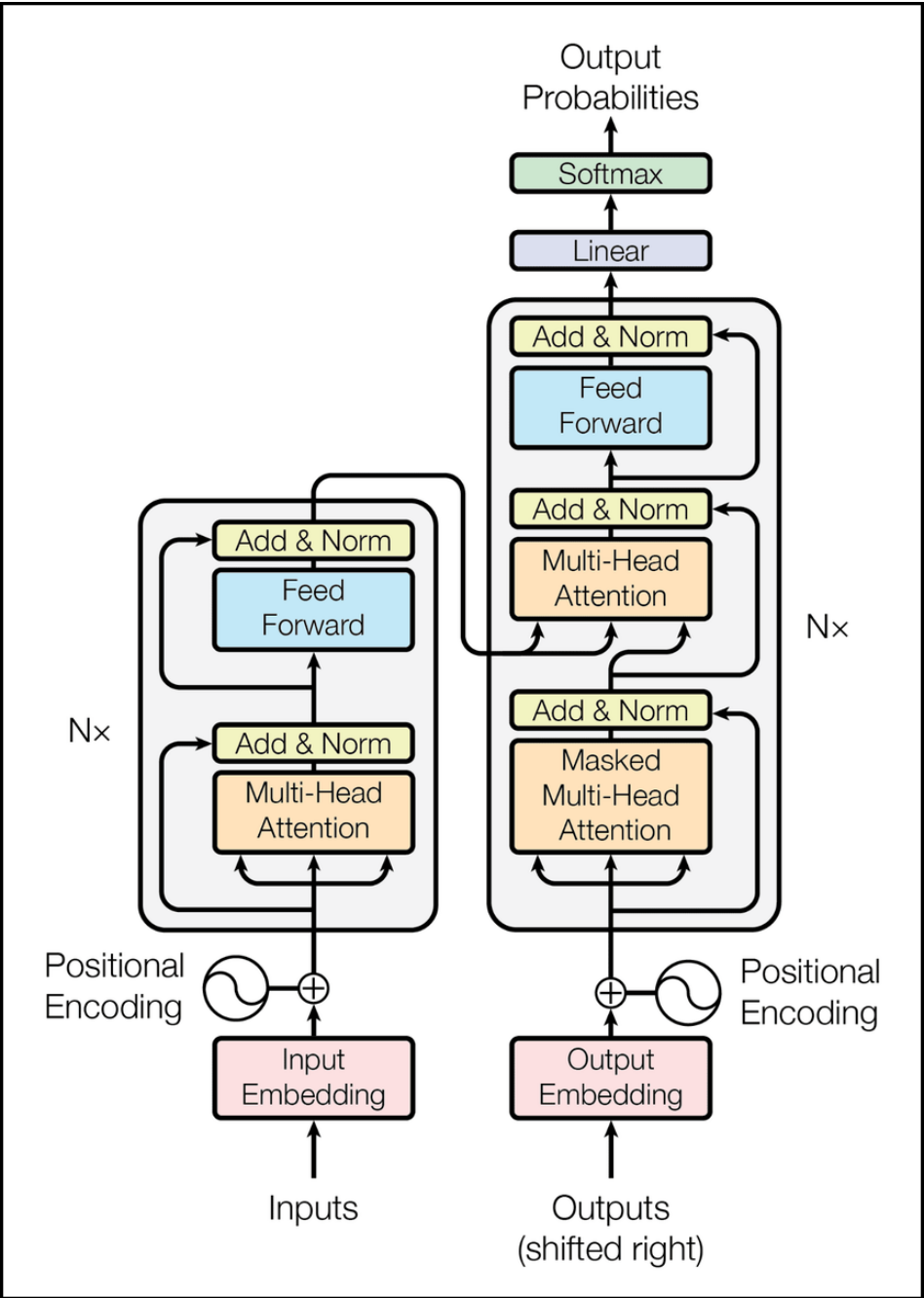


Figure 3.1: The Transformer - model architecture [1].

3.2 Computer Vision Model

The on-site insect detection and classification system leverages the power of deep learning models to achieve accurate and efficient results. By building upon previous research in the field of Computer Vision, the system employs a combination of detection and classification models to identify and categorize insects in images captured by the camera. The use of transformers in the computer vision domain allows the system to capture complex patterns and relationships in the image data, leading to high levels of accuracy and reliability. The following sections cover the details of the deep learning models and techniques used in the system.

3.2.1 Deep Learning

Within the field of Machine Learning exists its subset, Deep Learning, which involves the training of Neural Networks with many layers to perform tasks such as image and speech recognition [31]. A specific type of Deep Learning model used in Computer Vision is Convolutional Neural Networks (CNNs). CNNs consist of convolutional layers that automatically learn spatial rankings of features from the input data.

The ImageNet classification challenge has been a driving force behind developing Deep Learning models for Computer Vision. Notably, the AlexNet architecture [32] achieved a breakthrough in the ImageNet challenge using deep CNNs. YOLO (You Only Look Once) is another popular deep learning model for real-time object detection and classification [33]. What makes YOLO more efficient than traditional methods is its ability to process the entire inputted image in one go rather than multiple possessions. YOLO employs a single CNN that divides the input image into a grid and predicts each grid cell's bounding boxes and class probabilities. The model then thresholds these predictions to produce the final object detections.

Vision transformers [2] are a recent development that applies transformer architecture to computer vision tasks. They divide an image into fixed-size patches

and process them as a sequence using self-attention mechanisms. The model architecture of a Vision Transformer developed by Google is shown in Fig 3.2.

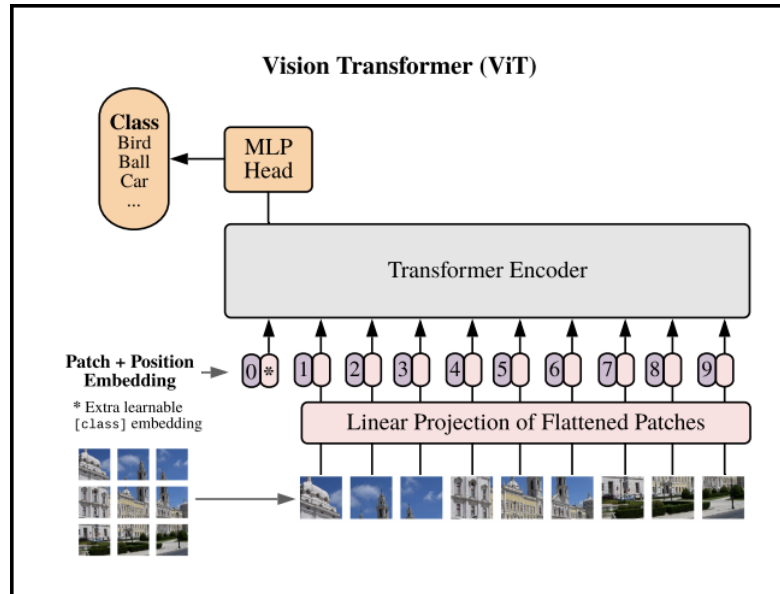


Figure 3.2: The Vision Transformer - model architecture [2].

3.2.2 Insect Detection and Classification Models

The on-site system that is responsible for sending insect data and running the Machine Learning models uses advanced AI detection and classification models to locate objects in the image returned from the camera and assign classes to those detections based on an insect dataset [34].

The dataset used combines manually sourced images of bee flies, the IP102 dataset [35], and the EU Moths dataset [36]. The 2,205 image dataset was manually annotated for the detection model to work. The final detection model was implemented using a Detection Transformer (DETR) as opposed to yolov5 due to its higher accuracy [37]. Similarly, the final classification model was implemented using a Vision Transformer instead of a CNN to improve accuracy in recognizing minuscule differences between insect species. Using both models, the automated

system can produce annotated images and extract data from them, as shown in Fig 3.3.

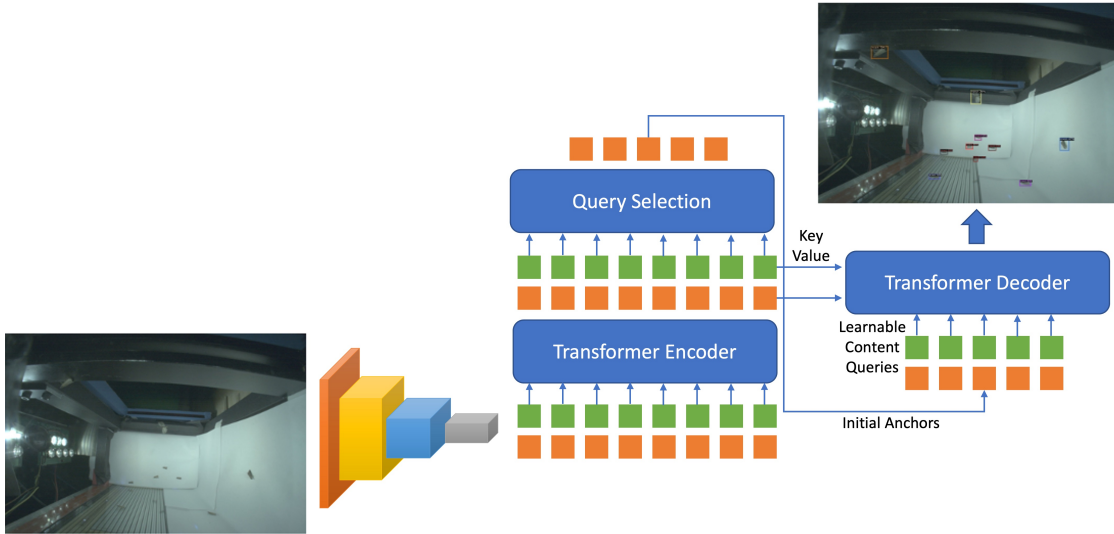


Figure 3.3: Bug system transformer diagram.

3.3 Generative Pretrained Transformer (GPT)

The Generative Pretrained Transformer (GPT) is a groundbreaking large language model (LLM) based on the Transformer architecture that leverages unsupervised learning for natural language understanding tasks. The underlying principle of GPT is a two-step process that involves unsupervised pre-training followed by supervised fine-tuning. Initially, the model is trained on a large unannotated text corpus to learn a general language representation. Subsequently, it is fine-tuned on specific, smaller labeled datasets for targeted tasks [38].

The GPT model employs a variety of masked language modeling called “masked multi-head self-attention,” which allows it to scale to longer sequences and capture long-range dependencies more effectively than traditional recurrent neural networks (RNNs). The Transformer architecture at the core of GPT enables the model to process input tokens in parallel, resulting in improved computational

efficiency and scalability compared to RNNs and LSTMs.

A key aspect of GPT's success is its ability to achieve state-of-the-art performance across various natural language understanding tasks, including sentiment analysis, natural language inference, and question answering. By leveraging a generative language model for unsupervised pre-training and fine-tuning for specific tasks, GPT outperforms previous models by a significant margin [38].

As GPT has evolved through different versions, such as GPT-3.5 and GPT-4, it has experienced significant improvements in model size, architecture, parameters, and performance. Each new version incorporates advancements in training techniques, data sources, and architectural refinements that enhance language understanding capabilities. Fig 3.4 shows the enormous gap in the number of parameters between GPT-4 and previous models.

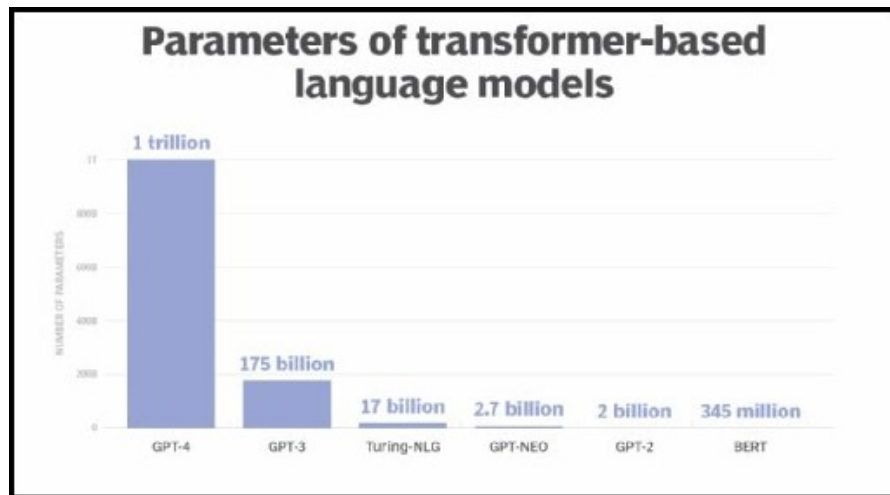


Figure 3.4: Chart of No. of Parameters for multiple LLMs [3].

3.3.1 Prompt Engineering

Prompt engineering is the development and refinement of input prompts to guide the behavior of language models. It involves crafting the input text to effectively communicate the desired task to the model and elicit the desired

response. Prompt engineering is important because the quality of the prompt can significantly impact the model's performance.

The choice of examples, the phrasing of the task, and other elements of the prompt can influence the model's understanding and output. For instance, in few-shot learning, the prompt may include a few examples of the task to be performed, followed by the specific instance on which the model should generate a response. Prompt engineering is still an active area of research, with development being done to create more effective prompts and evaluate model performance in various tasks for comparison [39, 40].

4 Implementation

This chapter covers the implementation process behind the user-facing application that combines the insect detection and classification model with an emulated agriculture and insect expert using GPT. The implementation process was divided into three major components: the backend, the frontend, and the AI Chat feature.

The backend component is responsible for the application's core functionality, including the integration of the insect detection and classification model with the database management system. The backend was developed using TypeScript and ExpressJS framework, which provided a solid foundation for building scalable and robust web applications. For database management, we've opted for Firebase, a highly-regarded open-source solution by Google known for its dependability and data integrity.

The frontend component is responsible for the visual representation and interaction of the application with the user. It includes the development of the user interface, which was designed to be intuitive and user-friendly, allowing farmers to access the features they need easily. The frontend was implemented using modern web technologies and packages such as HTML, CSS, TypeScript, Recharts, and more, with the React framework being used as the primary tool for development.

The AI Chat feature is the most innovative component of the application. It integrates OpenAI's GPT-3.5 model to provide personalized responses and recommendations based on the unique input from each farmer. This feature was developed using Typescript and the OpenAI API. Implementing this feature required extensive testing of various prompts to ensure its effectiveness as an emulator of agricultural expertise.

4.1 Backend

The backend for the Insect Detection Server project is primarily responsible for managing the database, which stores detections and related data from the on-site machines, and maintaining the Application Programming Interface (API). The project's API is a service between multiple applications that essentially exposes routes to the frontend to read from the database and exposes routes for the on-site machines to write data to the database.

4.1.1 Database

The original design of this work had several goals in mind, which included creating an interface for farmers to access their pest-related data and connecting on-site machines running computer vision models responsible for insect detection and classification. Since there were no requirements for complicated queries on the stored data, a NoSQL approach with high scalability was preferred, and Google Firebase's Firestore Database met those demands better when compared to an alternative such as MySQL [41]. The importance of designing the database schema lies in matching the payloads of the requests we send from other applications through our API. As such, we chose to use a simplified JSON payload that contains the date of the detection, the image with the bounding boxes returned from the on-site machine, and the array of detection objects, including all of the insect detections/classifications from the model. As shown in Fig 4.1, the structure of each detection within the detections array includes the date it was captured and the name and confidence score returned by classification.

```
interface Detection {
  confident_score: number;
  date: Date;
  name: string;
}
interface DetectionInstance {
  date: Date;
  detections: Detection[];
  image: string;
}
```

Figure 4.1: TypeScript interface structures for the Detection and Detection Instance object used in the JSON payload and database schema.

Within the Firestore Database, the types match those of the structures from Fig 4.1. We store the image as a string rather than using a separate object storage service because we can convert our consistent 2K resolution images retrieved from the camera on the farm into Base64 strings, which are relatively easy to store and retrieve at smaller image sizes. Another concept specific to Firebase is “collections,” which are simply containers for “documents,” the unit of storage in Firebase [42]. The main benefit of these collections is that they provide easy user-based data storage, as each user can have a unique collection of data of a similar structure. The structure of each detection document within our detections collection can be seen in Fig 4.2.

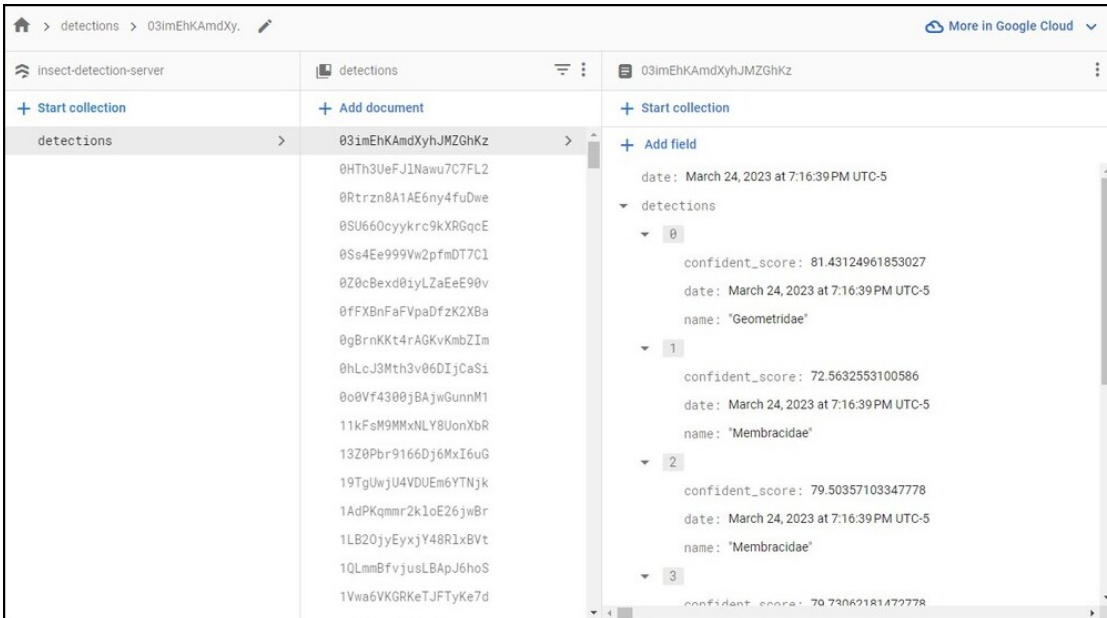


Figure 4.2: Cloud Firestore Database data showcasing the detections collection and its child document structure. The document structure matches the interfaces from Fig 4.1.

4.1.2 API

The API serves as an interface between the frontend, on-site machines, and the database, allowing them to exchange data. It provides necessary routes to access and manipulate the data stored in Firestore. The API was developed in TypeScript using ExpressJS, a popular and powerful web application framework, and Firebase Functions, allowing us to run serverless functions responding to HTTP requests.

To implement the API, we defined the routes corresponding to the various actions the frontend and on-site machines can perform. The primary routes include fetching all detections for a specific day, fetching the latest detection data, and posting new detection data. The JSON response for a typical GET request from the frontend for a particular day's data is shown in Fig 4.3.

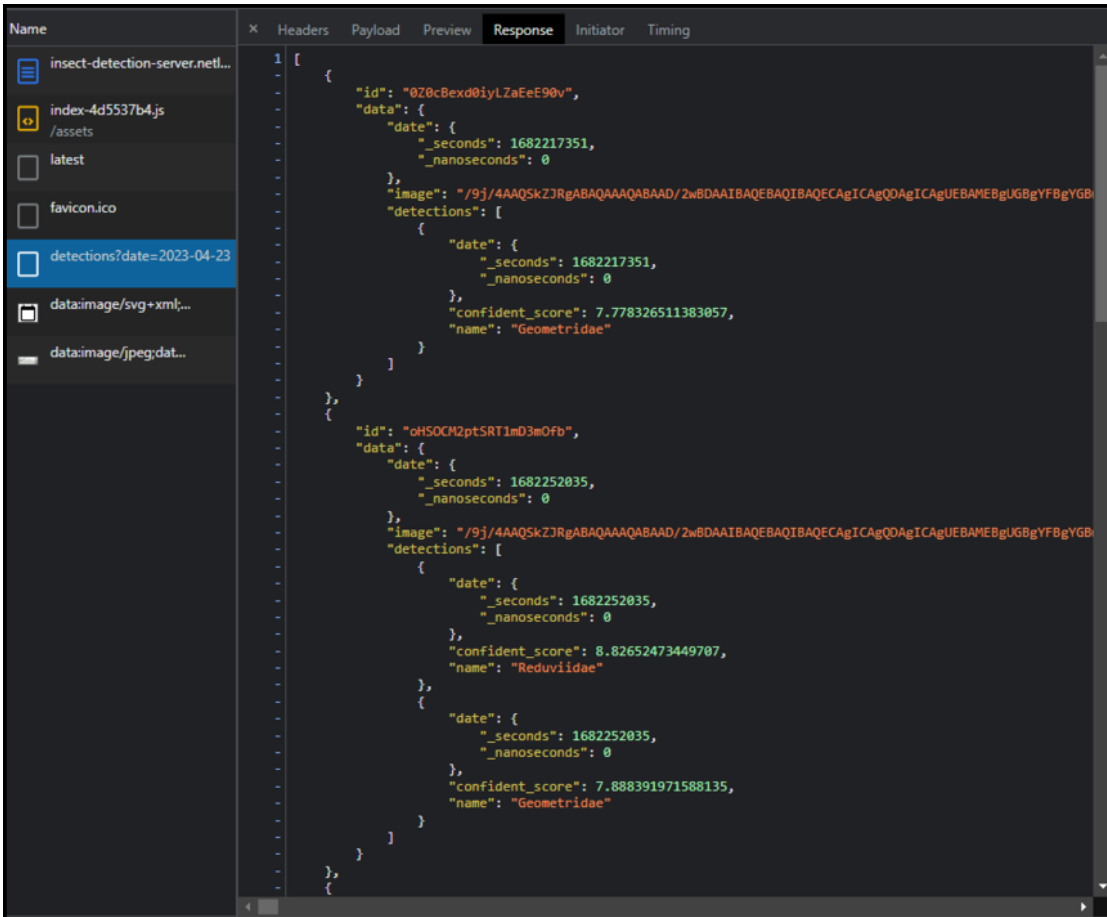


Figure 4.3: JSON response example for a GET request from the frontend, returning all the detection data for a specific day.

The on-site machines use the POST route to send new detection data to the database. These machines are responsible for running the insect detection and classification models, and they send the resulting data as a JSON payload. The response of the GET request is very similar to the payload of the POST request, as they contain the same data. A key difference is the format of dates due to their storage as milliseconds-since-epoch in the database. In Fig 4.4 we can see the POST request payload that matches up with the second detection document in the response from Fig 4.3.


```
{
  "date": "Apr-23-2023-01:00:00",
  "detections": [
    {
      "confident_score": 88.2652473449707,
      "date": "Apr-23-2023-01:00:00",
      "name": "Reduviidae"
    },
    {
      "confident_score": 78.88391971588135,
      "date": "Apr-23-2023-01:00:00",
      "name": "Geometridae"
    }
  ],
  "image": "/9j/4AAQSkZJRgABAQAAQABAAQ/
2wBDAAIBAQEBAQIQAQECAGICAgQDAgICAgUEBAMEBgUGBgYFBgYGBwkIBgcJBWYGCAsICQoKCgoKBggLDAsKDAKACgr/
2wBDAQICAgICAgUDAwUKBwYHCgoKCoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgo/
8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/
8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV
1hZWMkNkZWNnGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3
+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/
```

Figure 4.4: JSON payload example for a POST request from the on-site machine.

To handle these requests, we use Firebase Functions to define serverless functions that interact with the FireStore database. For example, we use a Firebase Function to fetch the latest detection data such that we can query the FireStore collection and return the most recent detection document. The sample Firebase Functions for handling the latest detection data request and saving detection data from the on-site machines are shown in Fig 4.5 and Fig 4.6, respectively.

```
// get latest detection
app.get("/latest", async (req, res) => {
  res.setHeader("Access-Control-Allow-Origin", "*");

  try {
    const latestQuerySnapshot = await db
      .collection(dataCollection)
      .orderBy("date", "desc")
      .limit(1)
      .get();

    const latest = latestQuerySnapshot.docs.map((doc) => ({
      id: doc.id,
      data: doc.data(),
    }));

    res.status(200).json(latest);
  } catch (error) {
    res.status(500).send(error);
  }
});
```

Figure 4.5: Function for handling a GET request to fetch the latest detection data from the database.

```

// Create new detection
app.post("/detections", async (req, res) => {
  try {
    const detectionInstance: DetectionInstance = {
      date: new Date(req.body["date"]),
      detections: req.body["detections"].map((detection: Detection) => ({
        ...detection,
        date: new Date(detection.date),
      })),
      image: req.body["image"],
    };

    const newDoc = await db.collection(dataCollection).add(detectionInstance);
    res.status(201).send(`Created a new detection instance: ${newDoc.id}`);
  } catch (error) {
    res.status(400).send("Format Incorrect or Image String too long!!!");
  }
});

```

Figure 4.6: Function for saving new detections to the database.

Once the API and Firebase Functions were implemented, they could then be deployed to Firebase Hosting. This process involves bundling the API code and configuration files into a single package and uploading it to the Firebase servers. The deployment process is straightforward and ensures the API is accessible and scalable, handling many requests as needed. Ultimately, the API implementation facilitates seamless communication between the frontend, on-site machines, and the FireStore database. It enables efficient data retrieval and storage, ensuring a smooth user experience and accurate tracking of insect detection data over time.

4.2 Frontend

The frontend component was implemented using the React framework and TypeScript. React is a widely-used library for building user interfaces, while TypeScript adds static types to JavaScript, ensuring better code quality and maintainability. The frontend is deployed on Netlify, which automatically builds and deploys the application from the GitHub repository.

The primary goal of the frontend design was to provide an intuitive and user-friendly interface for farmers to monitor insect detections and classifications daily and review individual detection instances. To achieve this goal, we created a UI with a header row with a date picker in the center and an open chat button in the top right. The page's main content consists of a pie chart for the total insect detection statistics for the selected day, followed by images of detections throughout the day, which users can click left and right through. The pie chart is implemented using Recharts, a popular charting library for React.

Figure 4.7 presents the website UI with the pie chart at the top and the image with detections returned from the on-site machine at the bottom. The pie chart is tooltip-enabled, allowing users to see the average confidence percentage and the insect's name when they hover over a chart section.

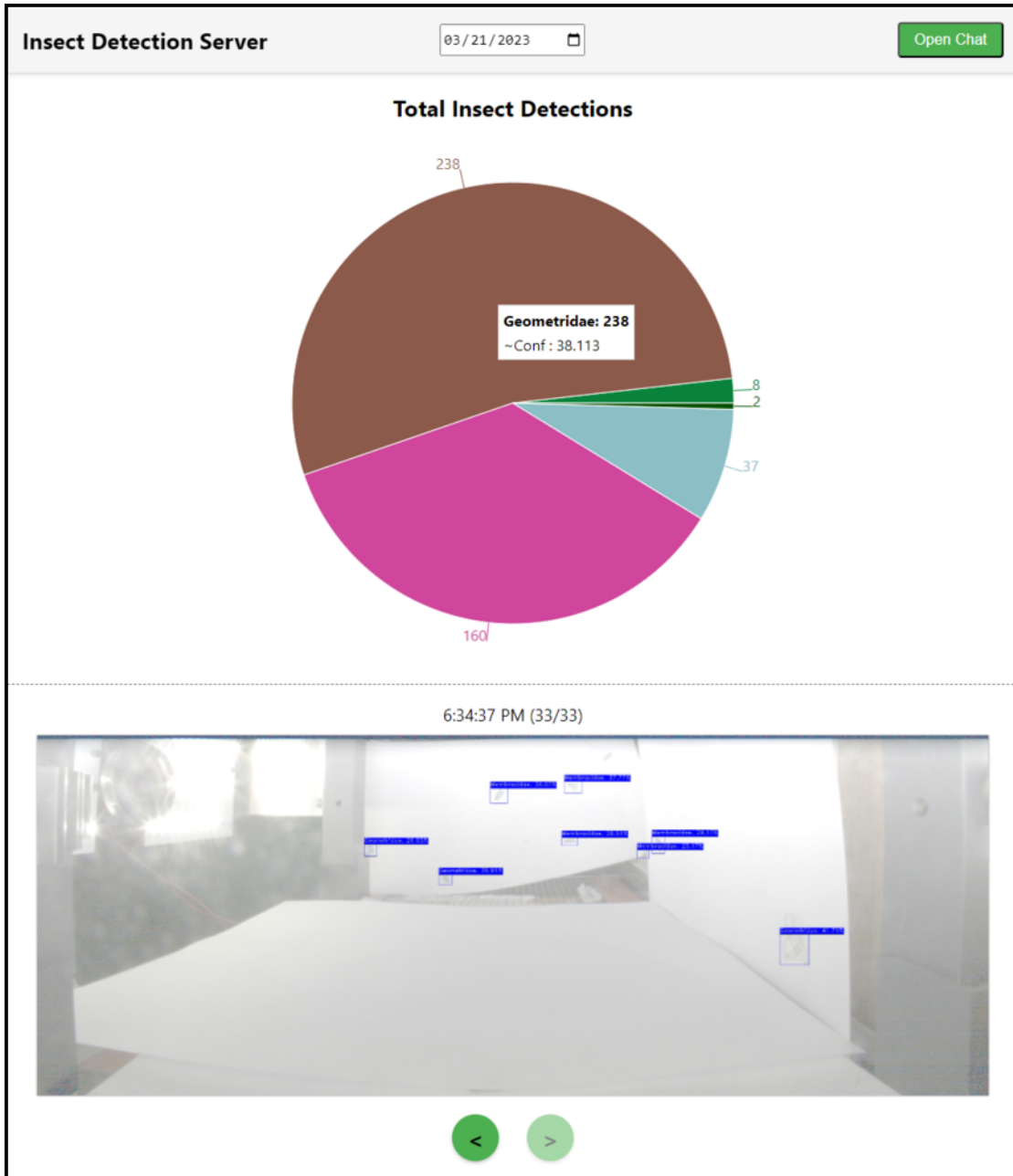


Figure 4.7: The frontend user interface showcasing the pie chart with tooltip functionality and the image carousel for individual detections.

4.3 Artificial Intelligence Chat Feature

The Artificial Intelligence chat feature is designed to provide farmers with personalized responses and recommendations based on their specific questions related to pests. To implement this feature, we integrated OpenAI’s GPT-3.5 model using the OpenAI API and TypeScript. The decision to use GPT-3.5 over OpenAI’s recently released GPT-4 was due to the superior speed and simplicity of GPT-3.5. However, due to its advanced reasoning, integrating GPT-4 could result in more advanced discussions on the implications of the insects’ presence. To maintain the previous message history and enable the model to generate coherent and accurate responses, exchanged messages are saved during the conversation.

The chat feature is accessible via a chat modal UI, as shown in Figure 4.8, where users can converse with the AI about their pest concerns. The AI assistant can address various pest-related questions, from identification and potential impact to research and case studies and offering specific advice on control and management strategies tailored to the user’s farm and crop conditions. By leveraging the advanced capabilities of GPT-3.5, the AI chat feature can provide detailed and accurate information, empowering farmers to make more informed decisions regarding pest management on their farms.

A limitation of the GPT model used, and large language models as a whole, is hallucinations. Hallucinations in AI are the generation of outputs that could be valid when they are factually incorrect. These phenomena can be explained by the limitations in the data the model was trained on. The issue with these hallucinations is their potential to mislead users since they can cause improper decision-making and a breakdown in trust in the model’s other outputs [43]. We can see an example of GPT “hallucination” in frames 1 and 2 of Fig 4.8 as most of the research articles referenced in its output do not exist. There are efforts to resolve these cases of hallucinations by implementing a validation step on GPT outputs. Still, the most effective response is transparency and disclosure of the model being used and its possible inaccuracies [44].

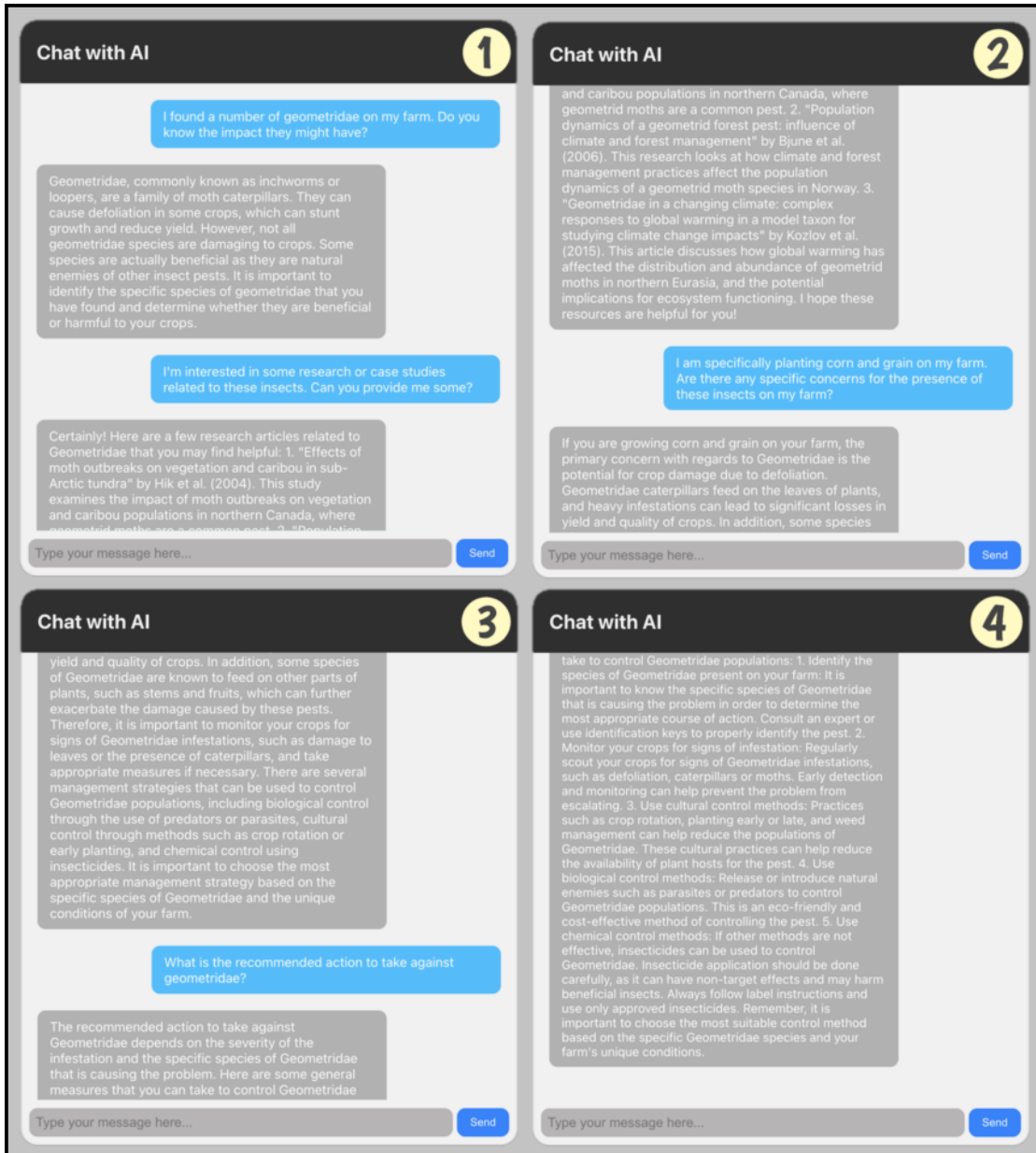


Figure 4.8: The chat modal UI with an example conversation about pest-related questions.

We employed prompt engineering techniques to guide the GPT-3.5 model in generating appropriate responses to ensure that the AI chat feature provides

relevant and accurate information. In the API call, we specified the expertise in agriculture and insects as a system prompt so that the generated responses are appropriate and informed by agricultural knowledge. The importance of using a well-tested system prompt is to narrow the scope of the model's responses and promote only relevant conversations that are most beneficial to the typical user. We can see an example of the model's narrowed scope and promotion of relevant questions in Fig 4.9. Figure 4.10 shows the example array of messages, a parameter in the API call's payload, including the system prompt.

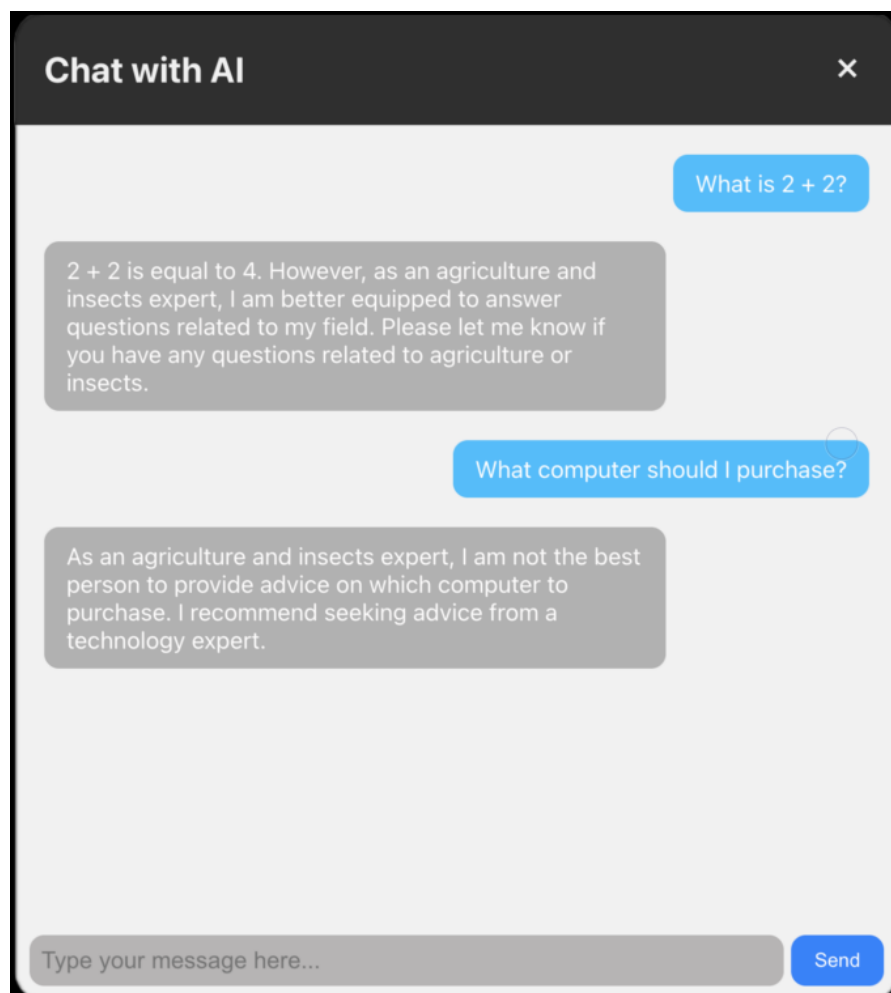


Figure 4.9: Example conversation where the AI assistant exhibits narrowed scope.


```
▼ (9) [{"role": "system", "content": "You are an expert in Agriculture and Insects. Please answer user questions"}, {"role": "user", "content": "I found a number of geometridae on my farm. Do you know the impact they might have on my crops?"}, {"role": "assistant", "content": "Geometridae, commonly known as inchworms or loopers, can be both beneficial and harmful to crops. Some species are pests, while others are used in biological control."}, {"role": "user", "content": "I'm interested in some research or case studies related to these insects. Can you provide any?"}, {"role": "assistant", "content": "Certainly! Here are a few research articles related to geometridae in agriculture. I hope these resources are helpful to you."}, {"role": "user", "content": "I am specifically planting corn and grain on my farm. How can I manage the presence of these insects?"}, {"role": "assistant", "content": "If you are growing corn and grain on your farm, there are several strategies you can use to manage geometridae and other pests. These include crop rotation, using resistant varieties, and applying targeted pesticides."}, {"role": "user", "content": "What is the recommended action to take against geometridae?"}, {"role": "assistant", "content": "The recommended action to take against geometridae species and your crops is to identify the specific species and use targeted control methods. This may include using insecticides, biological control, or cultural practices."}]\nlength: 9\n[[Prototype]]: Array(0)
```

Figure 4.10: Example messages parameter of payload for an API call to OpenAI's GPT-3.5 model with specified expertise in agriculture and insects.

Implementing the AI chat feature required extensive testing and refining of prompts to ensure its effectiveness as an emulator of agricultural expertise. The result is an innovative and valuable addition to the application that helps farmers get quick and accurate information about their pest-related concerns, ultimately contributing to better decision-making in pest management.

5 Conclusion

The Insect Detection Server project demonstrates the potential of combining Artificial Intelligence, Computer Vision, and advanced language models to create a practical and accessible solution for pest management in the agricultural industry. By integrating the insect detection and classification model with OpenAI's GPT-3.5 model, the application provides farmers with a comprehensive tool that identifies insects on their farms and offers expert advice and personalized recommendations on pest management strategies.

The implementation of this project involved developing the backend, frontend, and AI chat feature, resulting in a user-friendly interface that allows farmers to monitor insect detections and engage in conversations about their pest concerns. The integration of GPT-3.5 API enables the AI chat feature to provide accurate and contextually relevant responses to user queries, emulating the expertise of a human agricultural specialist.

One of the primary benefits of this system is the potential for reducing pesticide use and hastening farmers' Integrated Pest Management plans, leading to improved crop yields and reduced environmental impacts. Furthermore, the techniques used in this research can be applied to a wide range of computer vision and user-facing data analytic applications, extending their potential implications beyond the agricultural industry.

Although the current implementation utilizes GPT-3.5, future work could explore integrating the more advanced GPT-4 model for even more advanced discussions on the implications of insect presence and potential pest management strategies. Additionally, further research could be conducted to refine the prompt engineering techniques and ensure the AI chat feature continues to provide relevant and accurate information in response to evolving agricultural practices and pest management strategies.

In conclusion, the Insect Detection Server project showcases the innovative application of Artificial Intelligence and Computer Vision technologies in addressing real-world problems farmers face. By providing a user-friendly and accessible tool for insect detection and expert advice, this project contributes to the ongoing efforts to improve the efficiency and sustainability of agricultural practices.

6 Publications

CONDA: Continual Unsupervised Domain Adaptation Learning in Semantic Scene Understanding (WACV 2023 - Submitted).

Bibliography

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.
- [2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: <https://arxiv.org/abs/2010.11929>
- [3] S. M. Kerner, “What is a large language model (llm)? – techtarger definition,” Apr 2023. [Online]. Available: <https://www.techtarger.com/whatis/definition/large-language-model-LLM>
- [4] S. Zahniser and K. Kassel, “What is agriculture’s share of the overall u.s. economy?” Jan 2023. [Online]. Available: <https://www.ers.usda.gov/data-products/chart-gallery/gallery/chart-detail/?chartId=58270>
- [5] A. DePietro, “2021 u.s. gdp by industry,” Nov 2022. [Online]. Available: <https://www.forbes.com/sites/andrewdepietro/2021/08/23/2021-us-gdp-by-industry/?sh=625ed1336970>
- [6] Lrichmond, “Department of agriculture: 2007 resolutions,” 2007. [Online]. Available: <https://www.nj.gov/agriculture/conventions/2007/pestmanagement.html>
- [7] A. E. M.-P. State, “Good vs. bad in battle of the bugs,” Nov 2010. [Online]. Available: <https://www.futurity.org/good-vs-bad-in-battle-of-the-bugs/>
- [8] R. Nehring, J. Fernandez-Cornejo, A. Vialou, A. Martin, S. J. Wechsler, and C. Osteen, “Pesticide use peaked in 1981, then trended downward, driven by technological innovations and other factors.” [Online]. Available: <https://www.ers.usda.gov/amber-waves/2014/june/pesticide-use-peaked-in-1981-then-trended-downward-driven-by-technological-innovations-and-other-factors/>

- [9] W. Aktar, D. Sengupta, and A. Chowdhury, “Impact of pesticides use in agriculture: Their benefits and hazards,” *Interdisciplinary Toxicology*, vol. 2, no. 1, p. 1–12, 2009.
- [10] L. Gut, P. McManus, R. Isaacs, and A. Schilder, “How pesticide resistance develops.” [Online]. Available: https://www.canr.msu.edu/grapes/integrated_pest_management/how-pesticide-resistance-develops
- [11] “What is integrated pest management (ipm)?” [Online]. Available: <https://ipm.ucanr.edu/what-is-ipm/>
- [12] J. Hardy, M. K. Lesher, and B. Olson, “Introduction to integrated pest management,” Jul 2021. [Online]. Available: <https://www.epa.gov/ipm/introduction-integrated-pest-management>
- [13] G. Zehnder, “Overview of monitoring and identification techniques for insect pests,” Feb 2009. [Online]. Available: <https://eorganic.org/node/2721>
- [14] K. Pinkston, “Sampling and monitoring for pests and their natural enemies.” [Online]. Available: <http://hortipm.tamu.edu/ipmguide/ento/chapters/sampling.html>
- [15] R. F. Long, P. B. Goodell, R. A. Baldwin, C. A. Frate, L. D. Godfrey, O. B. Orloff, W. M. Canevari, R. M. Davis, E. T. Natwick, D. H. Putnam, and et al., “Sampling with a sweep net,” Jan 2017. [Online]. Available: <https://ipm.ucanr.edu/agriculture/alfalfa/sampling-with-a-sweep-net/>
- [16] C. Magee, “The evolution of agricultural technology,” Jul 2020. [Online]. Available: <https://www.innovationnewsnetwork.com/the-evolution-of-agricultural-technology/6039/>
- [17] X. Zhang, J. Bu, and X. Wang, “Automatic pest identification system based on deep learning and machine vision in the greenhouse,” *SSRN*, Dec 2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4314554
- [18] K. S. Subramanian, S. Pazhanivelan, G. Srinivasan, R. Sathi, and N. Sathiah, “Drones in insect pest management,” *Frontiers in Agronomy*, vol. 3, 2021. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fagro.2021.640885>
- [19] S. Azfar, A. Nadeem, K. Ahsan, A. Mehmood, H. Almoamari, and S. S. Alqahtany, “Iot-based cotton plant pest detection and smart-response

- system,” *Applied Sciences*, vol. 13, no. 3, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/3/1851>
- [20] S. Qazi, B. A. Khawaja, and Q. U. Farooq, “Iot-equipped and ai-enabled next generation smart agriculture: A critical review, current challenges and future trends,” *IEEE Access*, vol. 10, pp. 21 219–21 235, 2022.
- [21] R. W. Mankin, C. W. McCoy, K. L. Flanders, J. Brandhorst, R. L. Crocker, and J. P. Shapiro, “Methods for acoustic detection of insect pests in soil.” [Online]. Available: <https://www.ars.usda.gov/ARSUserFiles/3559/publications/agroacoust3pp3-8.html>
- [22] M. Tannous, C. Stefanini, and D. Romano, “A deep-learning-based detection approach for the identification of insect species of economic importance,” *Insects*, vol. 14, no. 2, 2023. [Online]. Available: <https://www.mdpi.com/2075-4450/14/2/148>
- [23] K. Thenmozhi and U. Srinivasulu Reddy, “Crop pest classification based on deep convolutional neural network and transfer learning,” *Computers and Electronics in Agriculture*, vol. 164, p. 104906, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169919310695>
- [24] H. Navarro-Hellín, J. M. del Rincon, R. Domingo-Miguel, F. Soto-Valles, and R. Torres-Sánchez, “A decision support system for managing irrigation in agriculture,” *Computers and Electronics in Agriculture*, vol. 124, pp. 121–131, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016816991630117X>
- [25] C. Chen, Y. Liang, L. Zhou, X. Tang, and M. Dai, “An automatic inspection system for pest detection in granaries using yolov4,” *Computers and Electronics in Agriculture*, vol. 201, p. 107302, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169922006147>
- [26] J. Durkin, *Expert systems: Design and development*. Prentice Hall Professional Technical Reference, 1994.
- [27] E. Charniak and D. V. McDermott, *Introduction to artificial intelligence*. Addison-Wesley Publ. Co., 1988.
- [28] J. Gonzalez-Andujar, “Expert system for pests, diseases and weeds identification in olive crops,” *Expert Systems with Applications*, vol. 36, pp. 3278–3283, 03 2009.

- [29] J. McKinion and H. Lemmon, “Expert systems for agriculture,” *Computers and Electronics in Agriculture*, vol. 1, no. 1, pp. 31–40, 1985. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0168169985900043>
- [30] “Dr tania: An indonesian ai chatbot helps farmers identify and treat crop disease,” Jun 2022. [Online]. Available: <https://en.reset.org/indonesian-ai-powered-app-helps-farmers-identify-crop-disease-05252020/>
- [31] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436–444, 2015.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [34] P. Helton, K. Luu, and A. Dowling, “Artificial intelligence system for automatic imaging, quantification, and identification of arthropods in leaf litter and pitfall samples,” *Inquiry: The University of Arkansas Undergraduate Research Journal*, vol. 21, no. 1, 2022.
- [35] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, “Ip102: A large-scale benchmark dataset for insect pest recognition,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8779–8788.
- [36] D. Korsch, P. Bodesheim, and J. Denzler, “Deep learning pipeline for automated visual moth monitoring: Insect localization and species classification,” in *51. Jahrestagung der Gesellschaft für Informatik, INFORMATIK 2021 - Computer Science & Sustainability, Berlin, Germany, 27. September - 1. Oktober, 2021*, ser. LNI, G. für Informatik, Ed., vol. P-314. Gesellschaft für Informatik, Bonn, 2021, pp. 443–460. [Online]. Available: <https://doi.org/10.18420/informatik2021-036>
- [37] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, “Dn-detr: Accelerate detr training by introducing query denoising,” 2022.

- [38] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training.” [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
- [39] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, “Evaluating large language models trained on code,” *CoRR*, vol. abs/2107.03374, 2021. [Online]. Available: <https://arxiv.org/abs/2107.03374>
- [40] Y. Liu, S. Shen, and M. Lapata, “Noisy self-knowledge distillation for text summarization,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 692–703. [Online]. Available: <https://aclanthology.org/2021.naacl-main.56>
- [41] J. Clark, “Firebase database vs. mysql,” Jul 2022. [Online]. Available: <https://blog.back4app.com/firebase-database-vs-mysql/>
- [42] “Cloud firestore data model — firebase.” [Online]. Available: <https://firebase.google.com/docs/firestore/data-model>
- [43] A. Gungor, “Chatgpt: What are hallucinations and why are they a problem for ai systems,” Mar 2023. [Online]. Available: <https://bernardmarr.com/chatgpt-what-are-hallucinations-and-why-are-they-a-problem-for-ai-systems/>
- [44] H. Alkaissi and S. I. McFarlane, “Artificial hallucinations in chatgpt: Implications in scientific writing,” *Cureus*, 2023.

A Broader Impacts of Research

A.1 Applicability of Research Methods to Other Problems

The innovative techniques and technology utilized in the creation of the Insect Detection Server hold great potential for application in various fields beyond agriculture. The incorporation of Computer Vision and AI for insect identification and classification, along with GPT-3.5's ability to simulate agricultural expertise, can be tailored to tackle various issues in different industries.

1. **Medical Imaging:** In diagnostic scans, the Computer Vision methods employed for insect detection can be applied to medical imaging to identify and categorize medical conditions, such as tumors or lesions
2. **Environmental Conservation:** The technology for wildlife monitoring and conservation initiatives can be modified. Utilizing computer vision to identify and track endangered species in their natural habitats allows conservationists to gather essential data on population dynamics and devise effective conservation strategies.
3. **Security and Surveillance:** The computer vision model can be employed in security and surveillance systems to detect and recognize suspicious activities or individuals, thus enhancing security measures in public spaces and private properties.
4. **Customer Service:** The implementation of GPT-3.5 as an “online expert” can be expanded to other areas, including customer service and technical support. Companies can utilize AI chatbots to offer instant and personalized customer assistance, boosting customer satisfaction and reducing response times.

A.2 Impact of Research Results on U.S. and Global Society

The Insect Detection Server has the potential to generate substantial positive impacts on both U.S. and global society, particularly within the agricultural sector.

1. **Improved Pest Management:** The capacity to accurately identify and classify insects on farms enables farmers to implement targeted and timely pest control measures. This can result in more effective pest management, decreased crop damage, and heightened agricultural productivity.
2. **Reduced Pesticide Use:** By supplying farmers with precise information about pest presence, the technology can help minimize the indiscriminate application of pesticides. This can reduce the environmental and health hazards associated with pesticide exposure.
3. **Enhanced Food Security:** Better pest management and decreased crop losses contribute to improved food security by increasing the availability and affordability of food. This is especially crucial in regions where agriculture is a primary source of livelihood, and food security is a significant concern.
4. **Access to Expertise:** The integration of GPT-3.5 as an “online expert” offers farmers expert advice and guidance on pest management. This democratizes access to agricultural knowledge, particularly for small-scale farmers who may not have access to traditional agricultural extension services.

A.3 Impact of Research Results on the Environment

The implementation of the Insect Detection Server can produce several positive environmental impacts:

1. **Reduction in Pesticide Pollution:** By facilitating targeted pest management, the technology can decrease pesticide usage, reducing pesticide runoff into

water bodies and soil contamination. This helps safeguard aquatic and terrestrial ecosystems from the detrimental effects of pesticides.

2. **Biodiversity Conservation:** By curbing the indiscriminate application of pesticides, the technology can help protect non-target species, including beneficial insects such as pollinators, from pesticide exposure. This supports the conservation of biodiversity in agricultural environments.
3. **Sustainable Agriculture:** The technology promotes sustainable agriculture practices by providing farmers with data-driven insights for pest management. This empowers farmers to make informed decisions that balance agricultural productivity with environmental responsibility.

In conclusion, the Insect Detection Server holds the potential to promote more sustainable and eco-friendly agricultural practices. By harnessing AI and computer vision technologies, the research results can lead to enhanced pest management, reduced environmental impacts, and agricultural sustainability.

B Identification of All Software Used in Research and Thesis/Dissertation Generation

Machine #1:

Owner: Ahmed Moustafa

Software

Ubuntu 18.0.4

TypeScript

Open source web-development libraries

Google Firebase

OpenAI API

Netlify

Machine #2:

Included CPU on the prototype bug system

Software

Ubuntu 16.04

Required scripts for system operation

C Plagiarism Check

This thesis was created by Ahmed Moustafa. I attest that all work found in this document is my own. Any examples of repetition in other works is incidental and not plagiarized material. _____