INTELLIGENT SYSTEM FOR QUEEN CELL MATURITY TRACKING AND BROOD PATTERN ANALYSIS TO DETECT EARLY COLONY COLLAPSE IN HONEY BEE APICULTURE

LAT, BETINA GRACE C.[1]
College of Computer Studies
Laguna State Polytechnic
University Laguna,Philippines
latbetina@gmail.com

MONTESA, ROSEMARIE D.[2]
College of Computer Studies
Laguna State Polytechnic
University Laguna,Philippines
rosedecastromontesa@gmail.com

PINO, RENALYN N. [3]
College of Computer Studies
Laguna State Polytechnic
University Laguna,Philippines
renalynpino27@gmail.com

VILLARICA, MIA V.[4]
College of Computer Studies
Laguna State Polytechnic
University Laguna,Philippines
mia.villarica@lspu.edu.ph

a

ABSTRACT- Honeybee colony collapse remains a pressing concern in modern apiculture, often caused by queen failure, brood irregularities, and the delayed detection of reproductive instability within hives. Manual inspections are the most common method of checking, but they are labor-intensive, time-consuming, and prone to human errors, making early interventions difficult. This study offers a modern approach using deep learning and computer vision to support early detection through queen cell stage maturity detection and brood pattern anomaly analysis. The system was built following an iterative system development methodology and trained using 4,565 annotated queen cell images and 4,500 brood images from beekeeping farms and online sources. Multiple deep learning models were implemented and compared, determining the most effective method. Initially, Mask R-CNN and Faster R-CNN, were used as the baseline models and achieved moderate performance scores (overall AP ~54–57%). While Mask R-CNN performed better in generating segmentation masks for matured cells, Faster R-CNN showed stronger classification performance for failed and matured cells but struggled with predicting capped and small cells. Through employing RF-DETR, the model achieved improved results (mAP@50: 94.3%, precision: 95.0%, recall: 92.6%), emphasizing the significance of the dataset and preprocessing. The YOLOv11 Large and Medium models were also employed to detect and segment queen cell features. The Medium model slightly outperformed the Large model across five cell classes (mAP@50: 98.5% vs. 98.1%, precision: 95.6% vs. 94.7%, recall: 95.0% vs. 95.2%). Overall, YOLOv11 Medium ranked the highest for queen cell maturity classification (mAP@50 of 91.3%, precision of 97.6%, and recall of 90.4%). For brood stage analysis, the same model detected eggs, larvae, and pupae, achieving an overall mAP@50 of 87.2%, precision of 81.2%, and recall of 84.9%, though egg detection remained challenging due to small size and subtle features.

Keywords: Honeybee, Colony Collapse, Apiculture, Hive Monitoring, Queen Cell Maturity Stages, Brood Pattern Anomalies, Deep Learning, Computer Vision, YOLOv11, Convolutional Neural Networks Convolutional Neural Networks

I. INTRODUCTION

Honeybees play an important role in beekeeping activities, most importantly for their role as major pollinators of flowering plants, many of which are food crops and producing honey known as a superfood. However, low survival rate of these honey bees often leading to colony collapse within hive boxes, has been observed. This is mainly due to their high tendency to migrate, triggered by various factors such as queen failure and natural enemies. In addition, issues such as queen duplication, absence of a queen or brood anomalies, including irregular brood patterns, missing larvae or scattered eggs can indicate poor colony health and reproductive failure.

One major problem faced by beekeepers is the death or loss of the queen is of great negative impact for the well-being of a honeybee colony, beekeepers need to be aware if a queen has died in any of their hives so that appropriate remedial action can be taken (Ruvinga et al., 2023). The death or loss of the queen leads to colony decline and can be terminal if a new queen is not successfully raised or introduced. Worker bees attempt to rear a new queen by feeding selected larvae royal jelly, but this emergency queen rearing process only gives the colony one chance to recover. If the new queen fails to mate or is killed, the colony will eventually perish, typically surviving only a few months without a queen.

To address such issues and reduce manual monitoring efforts, various researchers have started exploring automated solutions that use advanced technologies to assess hive conditions and detect early signs of decline. One of these is the study titled "Automatic detection and classification of honey bee comb cells using deep learning,", where the researchers created a tool called DeepBee© to help automatically count and classify comb cells in hive images. Instead of doing everything manually, which takes a lot of time and effort, this software can detect and sort cells into seven types such as eggs, larvae, pollen, nectar, and honey. They used Circle Hough Transform and semantic segmentation, which led to a 98.7% detection rate, higher than previous methods. Out of 13 CNN models tested, MobileNet gave the best balance between speed and accuracy, with an F1-score of 94.3%.

Inspired by the success of advanced AI models, the system will use computer vision techniques like YOLO/YOLO-*seg* to detect and track queen cells in hive images. For brood pattern analysis, deep learning algorithms such as Convolutional Neural Networks (CNNs) will be applied to monitor egg-laying behavior and identify early signs of queen failure. The study will explore high-performing architectures like ResNets and EfficientNets for improved accuracy. Transformer-based architectures may also be employed to extract features and recognize brood and queen cell patterns.

Their use is expected to improve the reliability of hive health assessment and enable earlier identification of colony collapse risks.

Hence, this study explores their system in a progressive web application designed to predict queen cell hatching and detect early colony collapse. The system tracks queen cell maturity and predicts when a new queen will hatch, so beekeepers can prepare and manage their hives better. After the queen hatches, the system will analyze the brood pattern to see if the queen is laying eggs properly and identify if the brood is performing healthy. It will also log the age of new queens to keep track of their health over time. With that said, the system can spot early signs of colony decline and alert the beekeeper before things get worse, allowing for quick action to save the colony and keep it healthy and productive.

II. RESEARCH OBJECTIVES

This study aims to design, develop, and validate a progressive web application system that uses image processing and deep learning to assist beekeepers in monitoring queen cell maturity and brood patterns. The system is intended to support hive health management and help prevent colony collapse by December 2025.

The specific objectives of the study are as follows:

1. To collect and prepare an image dataset consisting of 4,565 high-quality images of queen cells and 4,500 images of broods at various stages of development and health conditions, obtained from both self-captured and external data sources. The dataset will be curated and further enhanced using image rotation, resizing, and color adjustments to improve the reliability and flexibility of the models.

2. To develop and train multiple deep learning models including YOLOs (v11 Small, Medium, Large), Mask R-CNN, Faster R-CNN, and RF-DETR for localized detection and segmentation. These models will be used to classify queen cell maturity stages (open, capped, semi-mature, mature, post-hatch and failed cells) and to identify brood anomalies that may indicate colony health status.
3. To apply predictive techniques that estimate the expected hatching time of queen cells based on visual characteristics, and to enable basic tracking of each queen's age after emergence through log entries, supporting ongoing monitoring of reproductive health.
4. To build and implement a user-friendly progressive mobile platform that integrates the selected model(s) and provides features such as early warning alerts when multiple mature queen cells or unusual brood patterns are detected. The system will be designed to be accessible and practical for beekeepers.

5. To evaluate the system's performance and acceptability by:
   - conducting quantitative testing using standard evaluation methods (accuracy, precision, recall, and F1-score) on reserved test images;
   - gathering qualitative feedback from a local beekeeper expert that the researchers' have been working with; and
   - carrying out field tests in real beekeeping environments to confirm the system's effectiveness and reliability. This final evaluation phase is scheduled for completion by December 2025.

III. LITERATURE OF THE STUDY

In the study titled "ColEval: Honeybee COLony Structure EVALuation for Field Surveys", Hernandez et al., (2020), developed ColEval, which is a simple and standardized method for evaluating honey bee colony structure in field conditions by visually estimating the coverage of different elements such as adult bees, brood (open and capped), nectar, honey, and pollen on both sides of each frame. Their method includes the use of images and correction factors to improve accuracy and minimize observer variability. In their findings, it states that adult bee counts based on visual estimation should be multiplied by a correction factor of 1.8 to better match actual bee numbers. Thus, the study highlights the importance of consistent and minimally invasive monitoring tools to assess colony structure, especially when surveying large numbers of hives.

In their study, Yıldız & Karabağ (2025) reviewed the use of deep learning, particularly, convolutional neural networks (CNNs), transfer learning, and hybrid models for image-based classification of honey bee (Apis mellifera) lineages, highlighting the importance of sophisticated image-processing techniques in distinguishing subtle morphological traits. They noted that while CNN-based methods offer high accuracy and rapid classification, challenges such as limited dataset size, labeling complexity, and environmental variability remain.

Haddaoui et al. (2024) in "A Comprehensive Review of Beekeeping Datasets for Precision Apiculture Research" conducted systematic analysis of dataset characteristics including accessibility, size, quality, and diversity using rigorous data cleaning processes and duplicate removal for biological image analysis. The study examined various datasets containing high-resolution bee imagery (ranging from 27–520 pixels) with RGB PNG images and CSV annotation files, with sizes varying from 400 to over 13,000 samples, finding significant variations in dataset quality and identifying critical gaps in standardized data collection methods for biological detection tasks. This research is relevant to the proposed study as it provides essential methodologies for dataset preparation and image preprocessing techniques for biological stage detection in beekeeping applications.

A study by, Karypidis et al. (2022) compared traditional ML techniques (BoVW with SIFT + KNN/SVM) against deep learning models (VGG16 and a custom CNN) for 2D image classification using the Belgium Traffic Sign dataset, showing that deep CNNs, particularly those enhanced with Mish activation, batch normalization, and data augmentation outperformed classical methods in both accuracy (over 90%) and generalization. Similarly, Alex et al. (2025) evaluated deep learning models for real-time bee monitoring using a dataset of 9,664 annotated images, benchmarking YOLOv5m, YOLOv5s, and YOLOv8m to assess performance in precision, recall, mAP, and inference speed. YOLOv5m achieved the highest accuracy (85.6%) but with slower inference, while YOLOv5s provided faster performance with slightly reduced accuracy, offering a practical trade-off for real-time use.

In recent years, web-based systems have become essential in apiculture, allowing beekeepers to monitor hives in real time and respond promptly to hive conditions. Alifieris et al. (2023) introduced "IOHIVE," an IoT-based system integrating hive sensors with a web platform and a wearable device. It captures real-time data (weight, temperature, humidity, sound, pressure) and enables interactive journaling, highlighting seamless sensor integration and user interaction. This supports the current study's aim to create an image-supported, web-integrated hive monitoring tool.

To achieve more feasible, accurate, and low-effort hive monitoring, artificial intelligence (AI) applications have gained traction in apiculture. Andrijević et al. (2024), in their study titled "Concept Solution of Autonomous IoT Smart Hive and Optimization of Energy Consumption Using Artificial Intelligence," introduced a fully autonomous smart hive integrating entrance video monitoring, sensor arrays, and AI-controlled ventilation powered by solar energy. Their system uses machine learning to predict weather shifts and optimize hive energy use while addressing challenges in reliable image acquisition, aligning with the present study's goal of incorporating queen cell and brood detection into intelligent hive systems.

The study of Mohseni et al. (2022) investigated the likelihood of Iranian beekeepers adopting Internet of Things (IoT) technologies through the lens of the Technology Acceptance Model (TAM) and UTAUT2. Their conceptual study aimed to identify the behavioral intentions and technological tendencies of beekeepers by examining perceived ease of use, usefulness, performance expectancy, price value, and social influence. With the Beedar device as a case study, the authors emphasized how smart product-service systems like Beedar, featuring IoT-based hive monitoring via a mobile application could enhance productivity, reliability, and ecological practices in beekeeping. The study integrated multiple variables across disciplines such as psychology, economics, and information systems, and proposed a model to guide the acceptance and diffusion of smart technologies within agriculture. This is highly relevant to the current research, as it not only addresses the same smart apicultural technology but also provides a foundational framework for assessing Iranian beekeepers' technology acceptance behavior, which can inform both product design and policy-making.

IV. METHODOLOGY

Research Design

In this study, the research design used to conduct the study is depicted. A research design is intended to provide an applicable framework for a study. The research design is a plan that organizes how the combined study elements address the research problem effectively, which guides data collection, measurement, and analysis (Thakur, 2021). Furthermore, the proponents used descriptive-experimental research methods. The descriptive design allows the researchers to obtain relevant information needed, identifying current situations and trends through observation, interviews, surveys, and case studies. Meanwhile, the experimental design involves testing the developed system in controlled settings to measure its effectiveness, performance, and accuracy based on predefined criteria.

Applied Concepts and Techniques

This study uses image processing, deep learning, and progressive web application system development to monitor hive health. Thus, some computer vision models, including CNNs and YOLOv8, were used to detect queen cell maturity and identify abnormal brood patterns from hive images. These models were trained using annotated datasets and integrated into a mobile application that provides real-time insights to beekeepers. Overall, the study applies deep learning techniques such as CNNs, ResNets, EfficientDets, and transformer-based models to support early detection of colony collapse and better hive management.

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are deep learning models designed to process and analyze visual data. As explained by Taye (2023) in the study *"Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions"*, CNNs use layers such as convolution, pooling, and activation to automatically extract and learn image features. These features help the model recognize patterns in images with minimal manual effort, making CNNs highly effective for tasks like image classification and object detection.

Relating to the current study, CNNs are used to detect and classify queen cell maturity and brood pattern stages in hive images. By applying this concept, the system can identify biological patterns that signal changes in hive health.

YOLO (You Only Look Once)

YOLO has been a leading one-stage object detection framework known for its speed and real-time performance. In the year 2025, YOLOv8 and its segmentation variant (YOLOv8‑seg) have become widely adopted for both object detection and instance segmentation tasks.

For example, Bochkovskiy et al. (2023) reported that YOLOv8‑seg achieved a mask AP of 42.8 % on COCO while maintaining extremely low latency which is clearly suitable for applications requiring real-time inference on edge devices. Its unified design allows simultaneous bounding-box detection and segmentation mask prediction, which is particularly effective for scenarios where objects have subtle morphological differences or dense clustering.

Mask R‑CNN

Mask R‑CNN has remained a dominant architecture in instance segmentation through at least 2025, with numerous enhancements. For example, Srinivas et al. (2021) introduced BoTNet a ResNet backbone with self‑attention; that, when integrated into the Mask R‑CNN framework, achieved a Mask AP of 44.4 % on COCO and surpassed traditional ResNeSt baselines, while maintaining low latency. Its dual‑branch design, combining bounding‑box detection and pixel‑wise mask prediction, is especially well‑suited for discriminating morphologically similar cell types while supporting fast inference in mobile or low‑resource environments.

Faster R-CNN

It is a kind of CNN that has evolved from R-CNN. According to the area proposal network, a Faster R-CNN selects an arbitrary area of the image as the proposal area, and trains for obtaining the equivalent type and location of a specific area in the image.

Compared with the conventional selective explore technique, the Faster R-CNN is breaking the blockage issue of massive cost in the calculation as the RPN creates equivalent proposal sites. So, the practical examination becomes feasible. Additionally, using an adaptive scale pooling layer, a Faster R-CNN can adapt to arbitrary images and adjust the whole network to enhance the accuracy of deep network recognition. A faster R-CNN model is capable of breaking the time blockage of computation, and ensures an effective prediction rate is achieved. Thus, a Faster R-CNN analysis model is presented to process the feature extraction process in the insulator and the nest for identifying the destination. A Faster R-CNN technique is comprised of 2 CNN

RF-DETR

RF-DETR (Receptive-Field DETR) a real-time object detection and instance segmentation, achieving state-of-the-art performance through 2025. According to Robinson et al. (2025), the model represents the first real-time detector to surpass 60 AP on COCO, reaching 60.1 AP at 17.2ms latency.

RF-DETR Seg integrates a lightweight but also inspired by MaskDINO. This combination allows it to perform instance-level mask prediction without much computational cost. Besides, the RF-DETR variant "Nano" has been reported by Robinson et al. (2025) to be the best among the small models in terms of accuracy, albeit, with low latency. The multi-resolution training comes in as a smart way for RF-DETR to equally share speed and accuracy with different hardware configurations without retraining, thus, being able to support its deployment in cloud servers, mobile devices, and embedded systems.

Algorithm Analysis

This study employed two different but complementary computer vision strategies to analyze colony health: instance segmentation for queen cell maturity classification and object detection for brood pattern anomaly detection. Each task was trained on its own dataset, specifically adapted to the unique visual and spatial features of its focus area.

Although the current dataset was sufficient for baseline training, the research team is actively working to expand the queen cell image set. This includes gathering rarer developmental cases, more extreme environmental conditions, and diverse honeybee subspecies to ensure better generalization across hives, seasons, and geographic locations.

Data Collection Methods

This study required real-world images of brood and queen cells to support the detection of early indicators of colony collapse. The data gathering process involved field visits, expert validation, and direct observation of bee colonies in active hives.

The actual gathering to confirm the queen as an indicator of colony collapse was guided by Mr. Lucido. The expert showed the researchers brood frames and pointed out queen cells inside an active hive. These firsthand observations helped in identifying key features such as cell size, shape, and position, especially when a queen cell is about to hatch. Mr. Lucido also helped validate the images collected and confirmed the presence of queen-related structures in the dataset. Additional data and parameter validation were provided by Mr. Lee Gaitana of Pia's Bee Farm, an ATI-accredited Learning Site for Agriculture in Batangas. Mr. Gaitana, referred by the UPLB Bee Program, has over ten years of beekeeping experience. The expert confirmed the visual features used in the dataset and shared consistent patterns observed across different colonies, which strengthened the credibility of the annotations.

The researchers captured images manually using a mobile phone camera in natural lighting. Each image was reviewed with expert assistance and labeled accordingly, focusing on the visual traits of queen cells and brood cells. Discrepancies were resolved through follow-up discussions with both beekeepers.

Table 1. Data Sources Utilized

Data Type	Purpose
Queen Cell Images	Detect and classify maturity stages
Brood Pattern Images	Analyze brood consistency and egg-laying health

Table 2. Parameters Measured (Queen Cell Maturity)

Cell Stage	Description
Open Queen Cell	Elongated, open-ended; larva is seen (3-5 days old)
Capped Queen Cell	Partially sealed cell; transition stage (4-6 days old
Semi-Mature Cell	Uniform color (5-8 days old)
Matured Queen Cell	Conical tip dark; dotted lines on conical tip evident; ready to hatch
Failed Cell	Dead cell; failed process

Table 3. Brood Pattern For Early Colony Collapse Evaluation

Brood Pattern	Description
Egg Presence	No eggs = queen is absent or failing
Larvae Presence	No larvae = queen hasn't laid in days = early sign of collapse
Pupae	Sudden drop = brood cycle disrupted = potential collapse starting

Data Model Generation

To develop a reliable and robust deep learning model capable of identifying queen cell maturity stages and detecting anomalies in brood patterns, it was necessary to curate a well-structured and diverse dataset. This involved not only collecting sufficient images but also applying a comprehensive set of preprocessing and augmentation techniques. These steps were designed to improve model generalization, reduce bias as well as simulate real-world variations in bee colony environments. The following processes describe how the dataset was handled, from initial splitting to augmentation and fine-tuning for better learning outcomes.

Data Splitting

The complete dataset was partitioned into three subsets using a stratified split of 70% training, 20% validation, and 10% testing. This allocation was chosen to give the model ample exposure during training while preserving separate subsets for validating performance during the learning process and evaluating final outcomes on unseen data. Stratification allowed each subset to retain a proportionate distribution of brood and queen cell classes, which is important for accurate classification and performance consistency across different data stages.

Preprocessing Techniques

The total dataset target is 5,000 high-quality images, strategically divided between queen cell and brood analysis tasks. The exact split was based on the complexity and class diversity of each task. Queen cells required more representation due to multiple maturity stages, so the researchers allocated more images to that set. Prior to training each dataset underwent preprocessing and were independently enhanced through augmentation and used with different deep learning models to optimize detection and classification performance. The goal of preprocessing was to clean and standardize the images so that the model could interpret the inputs more effectively.

- **Resizing**: All images were resized to **640×640 pixels**, a standard resolution for many computer vision models. This resizes harmonized image dimensions across the dataset, allowing consistent input into the neural network without causing distortions or cropping essential features.
- **Aspect Ratio Padding (Black Fit Edges)**: To preserve the original shape and orientation of key visual features such as capped cells or queen cups, black padding was applied when the source image did not conform to a square format. This technique prevented the stretching or warping of cell shapes and retained spatial integrity.
- **Normalization**: Image pixel values were converted from their original 0–255 range to a normalized 0–1 scale. This step helped optimize model convergence and stabilized the learning process by reducing numerical disparities in the input features.

Data Augmentation

In agricultural image analysis, achieving optimal model performance remains a key objective, particularly in addressing visual recognition tasks such as classification, segmentation, detection, and localization despite persistent challenges posed by biological variability and unstructured field environments (Olaniyi et al., 2022). Likewise, data augmentation was applied to the training images to artificially increase the variety of input conditions the model could learn from. These transformations mimicked common scenarios in real hive environments.

Horizontal Flip: A 50% chance of flipping images along the vertical axis was applied. This allowed the model to recognize brood and queen cell patterns even when their orientation differed.

- **Rotation (±10°)**: Slight rotations were randomly applied within a ±10-degree range. These rotations account for the natural tilt of hive frames or handheld camera misalignments during data collection.
- **Brightness Adjustment (±20%)**: Brightness was randomly increased or decreased by up to 20%, simulating different lighting conditions such as morning light, shadows, or artificial illumination inside bee boxes.
- **Noise (σ = 0.5%)**: Low-level noise was added to simulate minor visual interference such as dust, reflections, or camera blur. This helped the model become less sensitive to slight imperfections in input images.

Feature Extraction

Following the completion of preprocessing and augmentation, the feature extraction stage marks a significant part of the deep learning pipeline wherein the model begins to interpret and encode significant visual information from the input images. This is accomplished through a series of convolutional operations that progressively detect and abstract image characteristics, ranging from low level cues such as edges, textures, and color transitions, to higher level patterns associated with biological structures present in the hive environment.

In the queen cell classification, the YOLO-seg/Mask R CNN model is equipped to isolate region specific features and delineate morphological differences essential for determining cell maturity, particularly across open, capped, and mature developmental stages. For brood pattern detection, the YOLOv8 architecture extracts spatially distributed features from different image scales, enabling it to distinguish between dense brood patches, scattered cells, and irregular formations that may suggest early colony stress.

Metrics

To evaluate the performance of the models trained on both the queen cell and brood anomaly datasets, a set of standard metrics commonly used in object detection and classification tasks was employed. Precision reflects the percentage of correctly predicted positive instances out of all positive predictions, while recall measures the proportion of actual positives correctly identified by the model. The F1-score, which is the harmonic mean of precision and recall, was used to provide a balanced measure that considers both false positives and false negatives.

In addition to these, mean Average Precision (mAP) was computed at different Intersection over Union (IoU) thresholds, with mAP@0.5 and mAP@0.5:0.95 serving as primary indicators of object detection performance. mAP@0.5 measures the average precision when the predicted bounding box overlaps with the ground truth by at least 50%, while mAP@0.5:0.95 averages precision across multiple thresholds, with a more comprehensive and stringent evaluation of localization accuracy.

Other supporting metrics included inference speed (measured in frames per second) and model size in memory, both of which are critical for assessing the system's potential for real-time or edge deployment in field conditions. A minimum performance threshold of 85% was targeted across core metrics, particularly for mAP and F1-score, to guarantee that the model meets deployment standards. Continuous evaluation during training and validation phases was conducted to monitor overfitting, performance degradation, or class imbalance. These metrics collectively guided model tuning, selection, and validation throughout the experiment lifecycle.

System Development Methodology

In this study, the researchers employed an iterative development approach, repeating cycles of model design, training, evaluation, and refinement.

According to Guerriero et al. (2023), the DNN Assessment and Improvement Cycle (DAIC) framework exemplifies such an iterative life-cycle approach for deep neural networks, where operational data and validation feedback inform incremental retraining to improve accuracy over time. Through following this approach, the researchers were able to iteratively refine models such as YOLOs, CNN-based models, and Transformer-based models such as RF-DETR, adjusting parameters and correcting issues based on validation results.

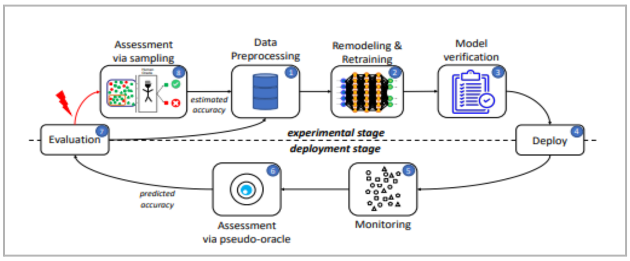

Figure 8. Example of the iterative approach using the DNN Assessment and Improvement Cycle (DAIC). Adapted from Guerriero et al. (2023)

For the modeling phase, different deep learning algorithms were used, including YOLOv8, CNN, ResNet, and MobileNet. Each model was trained and tested using Google Colab with GPU support to speed up the process. The researchers adjusted hyperparameters such as learning rate, batch size, and number of epochs based on how the models performed in early runs.

The process was repeated multiple times to improve accuracy and reduce errors. The use of model checkpoints and logs helped in monitoring training progress and identifying overfitting or underfitting issues. This cycle of training and evaluation continued until the results were stable and acceptable.

Software Tools Used

The researchers utilized a combination of open-source and cloud-based tools to support the development and training of the deep learning model.

- **Roboflow**

  The researchers used Roboflow to handle dataset preparation. This tool was essential for image annotation, where brood and queen cells were labeled manually.
- **Google Colab**

  The researchers conducted the training and evaluation phase in Google Colab. It provided free access to GPUs, allowed easy integration with Python and libraries such as PyTorch, which were crucial in the modeling phase.
- **Python**

  Python is used for writing scripts and for handling all backend tasks such as preprocessing data, training and evaluating the YOLOv8 model, and visualizing results. It also allowed integration with Google Colab notebooks and interaction with the trained model for prediction and testing.
- **OpenCV**

  The researchers employed OpenCV for image preprocessing tasks. It was used to resize images, apply color conversions (such as BGR to grayscale or HSV), and augment data through operations such as rotation and flipping.
- **PyTorch**

  PyTorch is the primary deep learning framework behind YOLOv8. It enabled the researchers to implement custom training logic, manage model weights, and monitor loss during training. PyTorch's dynamic computation graph also helped in debugging and understanding model behavior during different stages.
- **NumPy**

  It assisted in image array manipulation and data transformation tasks, making the integration between OpenCV, PyTorch, and other tools more efficient.
- **Visual Studio Code**

  Visual Studio Code was used as the integrated development environment (IDE) during the coding and debugging stages of the project. It allowed the researchers to edit Python scripts, manage dependencies, and integrate version control systems like Git.

System Architecture

The system architecture of the system is structured to support a complete pipeline for visual analysis of honeybee colonies, encompassing data acquisition, preprocessing, model training, evaluation, and eventual deployment.

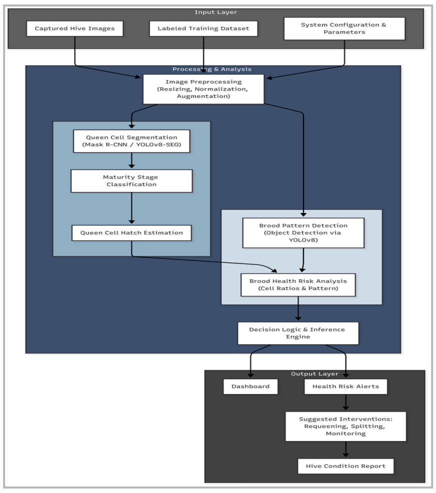

Figure 9. System Architecture Flow Diagram

The framework provides a detailed representation of the system's architecture showing how various components interact from data input to processing and ultimately to output generation. The framework is composed of three main components: the Input Layer, the Processing Pipeline, and the Output Layer. These components transform raw hive image data into meaningful diagnostic outputs that support decision-making for early colony collapse detection.

The Input Layer represents the primary sources of data used by the system. This includes two major categories: (1) pre-collected datasets and (2) real-time images taken directly from actual bee farms. Public datasets are gathered from open-source repositories and research databases that contain labeled images of brood frames and queen cells. In parallel, field data is collected from Lucido Bee Farm in Rizal, Laguna, where actual images of brood frames and queen cell structures are captured under natural conditions. These images undergo a strict quality control and curation process to ensure the visibility, clarity, and relevance of features such as capped brood, eggs, larvae, and queen cells.

Following data collection, annotation is performed using Roboflow, where each region of interest is labeled according to specific classes such as open queen cell**,** capped queen cell**,** mature queen cell**,** failed queen cell**,** and brood elements like egg (eggs not abnormal)**,** larva (larvae not abnormal**)**, and pupa (pupae not abnormal) and other related attributes. This process transforms raw images into structured datasets suitable for machine learning model training. Additionally, this layer includes the mobile-based interface infrastructure, which facilitates user interaction with the system. Through this interface, users mainly beekeepers can upload hive images, browse past uploads, and initiate analysis workflows without needing technical expertise, making the system accessible even to smallholder or backyard beekeepers.

The Processing Pipeline is the central component of the system, where artificial intelligence and computer vision techniques are applied to perform detection, classification, and interpretation of queen cells and brood patterns. At the start of this pipeline, the uploaded images undergo a series of preprocessing steps including resizing, normalization, noise reduction, and augmentation to improve model robustness and generalization. This study utilizes object detection for efficient identification and counting of brood and queen cells, and complements it with instance segmentation in specific modules to analyze cell morphology and highlight potential anomalies, offering a dual-layered approach to early colony collapse detection.

After preprocessing, the images are routed into two parallel processing modules. The first module utilizes a YOLOv8 instance segmentation model trained specifically to detect and classify queen cells into their respective maturity stages. The second module involves a convolutional neural network (CNN)-based detection model that analyzes the brood pattern by identifying healthy brood cell types and locating anomalies. It assesses parameters such as uniformity, density, centrality, and presence of patchy or scattered brood areas indicators often linked to failing queens or early colony collapse.

Beyond individual cell classification, logic layers are applied to synthesize findings from both modules. For example, the system correlates a high count of mature queen cells with simultaneous presence of irregular brood patterns to trigger an early warning. These logic-driven evaluations are integrated to deliver contextually relevant data, making deep learning not just a detector but also a basic recommendation system.

The Output Layer represents the end-point of the system's workflow, where findings are translated into user-friendly outputs and recommendations. This includes both visual and textual outputs. Visually, the system displays annotated versions of the uploaded hive images, showing bounding boxes or segmentation masks with labels such as "Mature Queen Cell" or "Abnormal Brood"overlaid on the original frame. This gives the beekeeper with a direct, intuitive view of detected hive elements. Quantitatively, the system calculates metrics such as the percentage of healthy versus abnormal brood cells and the number of detected queen cells per maturity category. These values are aggregated to produce a *Hive Health Score,* which acts as a simplified metric of colony condition. If the calculated score indicates potential risk such as abnormal brood exceeding a certain threshold or mature queen cells being abnormally numerous the system generates a flag and provides short management tips. These may include suggesting requeening, enhancing nutrition, isolating queen cells, or increasing inspection frequency.

The outputs aim to serve not just as reports but as early warning indicators**,** for beekeepers to act before symptoms of colony collapse become irreversible. The clear, science-based feedback loop promotes proactive hive management and reduces dependency on experience-based guesswork. Simplifying complex biological observations into structured, AI-supported analysis, this framework empowers smallholder beekeepers with accessible, real-time, and interpretable data. Ultimately, this also supports sustainability, bee health preservation, and reduced colony loss through technology-driven precision apiculture.

The model training was carried out using GPU-accelerated environments with optimized parameters including early stopping, learning rate scheduling, and data balancing. Performance metrics such as precision, recall, mAP@0.5, and F1-score were monitored closely across validation folds. Inference speed and memory efficiency were also evaluated to determine deployment feasibility on constrained devices. The system is being optimized for future deployment through conversion to lightweight formats like TensorRT or ONNX for edge inferencing. This is consistent with recent advances in AI-assisted agriculture, where deep learning systems are deployed on low-power devices in the field for real-time feedback El Akrouchi et al. (2025). Overall, the system architecture ensures both scientific rigor and practical scalability for real-world beekeeping applications.

Software Testing

- **Unit testing** was conducted to verify the integrity of each data handling and model inference module. These tests ensured that individual scripts such as those responsible for image loading, annotation parsing, or result rendering functioned as expected under diverse input formats and resolutions.
- **Integration testing** validated the interaction across modules. For example, the connection between the augmented dataset loader, training loop, and evaluation metrics was tested under various batch sizes and training epochs. This phase was essential in identifying mismatches such as label misalignment and inconsistent bounding box scaling, especially when integrating both segmentation and detection models (Charisis & Argyropoulos, 2024)
- **Model testing and evaluation** included quantitative validation on unseen data splits, guided by industry-standard metrics. Class-wise accuracy, confusion matrices, and mAP curves were analyzed to detect recurring misclassifications. To make the system operate under practical constraints, stress tests were executed on large datasets such as brood and with reduced memory conditions. These tests simulated low-resource edge deployment environments and helped identify memory bottlenecks and inference lags.

- **User-based testing** was also introduced to capture real-world effectiveness. Ten beekeepers participated in reviewing the output annotations and provided qualitative feedback on usability, correctness, and clarity. Feedback from this phase informed interface revisions and guided the prioritization of post-processing features.

- **Automated testing scripts** were developed to continuously evaluate training progress, detect anomalies, and log training artifacts. This continuous testing mechanism ensures model reproducibility and supports iterative improvements during expansion of the dataset.

V. RESULTS AND DISCUSSION

Research Objective 1: This objective aims to collect and prepare an image dataset for queen cell maturity classification and brood stage detection. The images were collected from an actual bee farm and sourced available online, then these were enhanced through resizing, rotation, and augmentation to improve data quality.

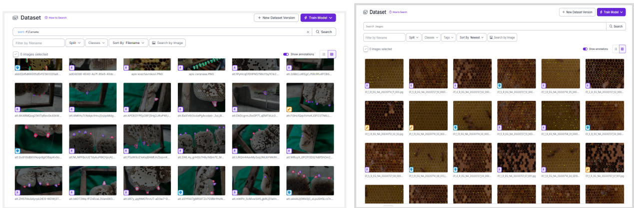

Figure 12. Collected Queen Cell Brood Image Datasets

Thus, the result shows from various model training that the datasets were sufficient and diverse, which allows the model to learn the different maturity stages of queen cells and the condition of brood. Hence, this implies that data augmentation the model's robustness and reduced overfitting, which ensure the reliability of training results.

Research Objective 2: This objective focuses on developing and training multiple deep learning models including YOLOs, Mask R-CNN, Faster R-CNN, and RF-DETR for localized detection and segmentation. Through this, the system classifies queen cell maturity stages (open, capped, semi-mature, mature, post-hatch and failed cells) and detect brood anomalies that may indicate colony health status.

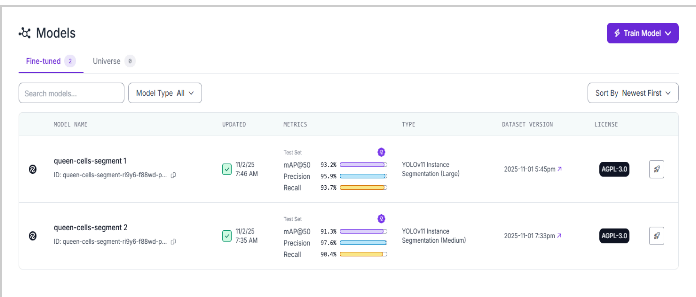

Figure 27. Fine-tuning with 4565 images (Final Counts)

Among the tested models, it shows that YOLOv11-based models showed the best results, wherein it showed the best balance of accuracy and speed, which is suitable for real-time mobile use. Nonetheless, YOLOv11 Medium performed the best, which slightly outperformed the YOLOv11 Large. Therefore, the researchers decided to use the YOLOv11 Medium model weight, as it proved to be the best fit for the system.

Research Objective 3: This objective applies predictive analytics to estimate the expected hatching time of queen cells based on visual characteristics, and to enable basic tracking of each queen's age after emergence through log entries, supporting ongoing monitoring of reproductive health.

The predictions were consistent with known biological timelines of queen development, as per discussed by Mr. Lee Gaitana, a beekeeper expert from Batangas. It states that queen cells undergo distinct morphological changes from the larval stage to pupation, which serve as reliable indicators of imminent queen emergence. Through analyzing these stages, the system can estimate hatching times and allow beekeepers to monitor queen development more precisely. Tracking queen age helps identify aging or underperforming queens, which is critical in preventing colony decline and improving hive management decisions. Therefore, tracking queen age helped identify aging or underperforming queens.

Research Objective 4: To build and deploy a user-friendly Progressive Web Application (PWA) that was accessible on both mobile and desktop such as Android, and iOS devices. The system was developed to help beekeepers easily monitor queen cell maturity and detect brood analysis and their counts using the models created in this study.

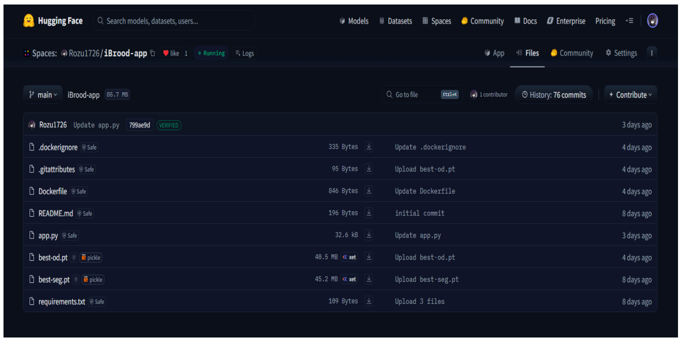

Figure 48: Hugging Face for Model Hosting

The backend used FastAPI to manage communication between the system and deep learning models. Hence, there were two models integrated, one is for queen cell maturity classification, and the other is for brood pattern analysis to detect colony health issues.

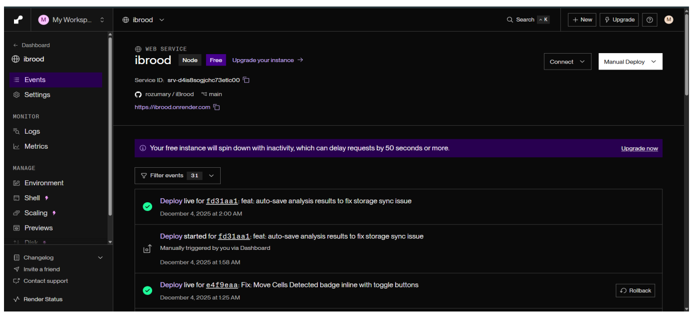

Figure 49: Render Deployment for Public Hosting

Consequently, the Render platform hosted the system which made it stay online and be available all the time via a web browser without the need for installation.

The system was interchangeable and suitable for field operation as it could also be reached on mobile devices via any browser. One of the reasons to set up the system under a .tech domain for one year minimum was to facilitate rapid testing, controlled access, and continuous refinement. Majorly, during this deployment phase access was exclusive to beekeepers, researchers, and students which allowed the focused evaluation, usability assessment, and integration of enhancements based on structured feedback.

Moreover, the system supports real-time upload and analysis of the images, which allows the beekeepers to upload hive frame images, and receive instant results generated by the model. The system also included early warning alerts that notified users when it detected multiple mature queen cells or abnormal brood patterns that can indicate the possible loss counts of brood development which can be an early sign of colony collapse. These features helped beekeepers act quickly and make better hive management decisions.

Research Objective 5: This objective focuses to evaluate the system's performance and acceptability by conducting quantitative testing using standard evaluation methods (accuracy, precision, recall, and F1-score) on reserved test images. Gathering qualitative feedback from local beekeepers that the researchers' have been working with and carrying out field tests in real beekeeping environments to confirm the system's effectiveness and reliability.

Table 1: Expert Rating for Evaluation Questionnaire

Overall All Assessment	
Functionality	4.8
Usability	5
Performance	5
Reliability	5
Overall user satisfaction	5

The five features are functionality, usability, performance, reliability, and overall user satisfaction were rated using a 5-point scale. The evaluation results indicate a consistently high assessment across all criteria, with most features receiving a rating of 5, and one feature receiving a rating of 4. Hence, considering that the evaluation was conducted by a single expert, the mean values are equivalent to the individual scores assigned for each criterion, and the computation of standard deviation is not applicable. However, the models achieved high accuracy and reliability, while the consistent high ratings reflect the expert's strong agreement that the system performs effectively, reliably, and meets the intended functional and usability requirements.

VI. SUMMARY OF CONCLUSIONS AND RECOMMENDATIONS

Summary

This study aimed to develop a progressive web application accessible through both mobile and desktop/laptop devices to help beekeepers detect early signs of colony collapse. The system analyzes queen cell maturity and brood patterns using image processing, machine learning, and deep learning to address the problem of late or inaccurate hive inspections, which often lead to queen failure, brood irregularities, and colony decline. To build its image-based monitoring features, the study prepared a curated dataset of 4,565 queen cell images and 4,500 brood images representing various developmental stages and health conditions. These images, sourced from both self-captured apiary photos from Mr. Lucido's bee farm and publicly available datasets online, were carefully annotated and enhanced through data augmentation to ensure reliability under real-world conditions.

Using this dataset, the researchers trained several models, including YOLOv11-Small, Medium, YOLOv11-Large, Mask R-CNN, Fast R-CNN, and RF-DETR to classify queen cell stages and detect brood abnormalities that may indicate colony stress. Thus, the YOLOv11-Medium model showed strong performance, particularly in detecting larva and pupa, while egg detection remained more challenging likely due to their smaller size and less distinct features.

Hence, through a systematic process of data gathering, preprocessing, model training, system development, field testing, and expert evaluation, the study revealed that artificial intelligence can reliably interpret hive images and offer practical insights for beekeepers. The findings confirm that automated detection of queen cell maturity and brood irregularities is both feasible and useful in real-world apiculture.

Conclusion

This study provided important insights into the development and use of an intelligent monitoring system for queen cell maturity tracking and brood pattern analysis in beekeeping. By applying machine learning, deep learning, and image processing, the system addressed major challenges faced by beekeepers, such as detection of queen failure, detection of queen cell hatching, brood irregularities, and early signs of colony collapse. The models performed well in identifying queen cell developmental stages and brood health conditions, supported by a well-prepared dataset and effective data augmentation. Its accessible, dual-platform (mobile and desktop/laptop) design also made the system practical for beginner beekeepers. These findings highlight the system's potential as a valuable tool for improving hive management and supporting timely interventions in real-world apiculture.

The study reached the following conclusions based on its findings:

1. The integration of machine learning and computer vision significantly improved the accuracy and speed of detecting queen cell stages and brood irregularities, offering beekeepers a more objective and reliable way to assess hive conditions compared to manual inspection.
2. The application of advanced models, particularly YOLOv11-Medium, YOLOv11-Large, Mask R-CNN, Fast R-CNN, and RF-DETR proved effective in recognizing key developmental stages, hence, YOLOv11-Medium showed the best performing in detecting larva and pupa. Thus, the system can provide early warnings before colony collapse occurs.
3. The system performed best in detecting pupa and larva, while egg detection remained the weakest, likely due to their smaller size and less distinct visual features. Among all classes, pupa achieved the highest and most stable F1 scores, followed by larva, while eggs consistently showed noticeably lower F1 scores. Hence, the system is highly

reliable for pupa and larva detection but still limited when identifying eggs.
4. The use of data augmentation and careful dataset preparation enhanced model performance and generalizability, which allows the system to handle real-world variations in lighting, angles, and image quality common in apiaries.
5. The developed application demonstrated that automated hive assessment is feasible and practical, helping beekeepers reduce manual monitoring effort, make quicker decisions, and maintain healthier and more productive colonies.

Recommendations

Based on the conclusions presented above, the researchers offers the following recommendations:

1. Future researchers should keep adding more queen cell and brood images from different sources and conditions. This will help the models become more reliable and work better in real beekeeping settings.
2. It is recommended to improve the techniques that estimate queen hatching time and the tracking of each queen's age. Adding more samples and continuous updates can make these predictions more accurate.
3. Since egg detection remains the most challenging due to their smaller size and less distinct features, future work should focus on improving model performance for eggs. Techniques such as higher-resolution imaging, enhanced preprocessing, or specialized detection approaches may help.
4. Future work should include more testing using accuracy, precision, recall, and F1-score, as well as gathering feedback from more beekeepers and experts. More field testing in real hives will help confirm the system's reliability and effectiveness.

LITERATURE CITED

Alex, A. J., Barnes, C. M., Machado, P., Ihianle, I., Markó, G., Bencsik, M., & Bird, J. J. (2025). Enhancing pollinator conservation: Monitoring of bees through object recognition. Computers and Electronics in Agriculture, 228, 109665. https://doi.org/10.1016/j.compag.2024.109665

Alifieris, C., Chamaidi, T., Malisova, K., Mamalis, D., Nomikos, E., Rigakis, C., Vlachogiannis, E., & Stavrakis, M. (2023b). IOHIVE: Architecture and infrastructure of an IOT system for beehive monitoring and an interactive journaling wearable device for beekeepers. In *Lecture notes in computer science* (pp. 133–149). https://doi.org/10.1007/978-3-031-37117-2_11

Alves, T. S., Pinto, M. A., Ventura, P., Neves, C. J., Biron, D. G., Junior, A. C., De Paula Filho, P. L., & Rodrigues, P. J. (2020). Automatic detection and classification of honey bee comb cells using deep learning. Computers and Electronics in Agriculture, 170, 105244. https://doi.org/10.1016/j.compag.2020.105244

Andrijević, N., Herceg, D., Maričić, S., Radivojević, V., & Jocić, G. (2024). Concept solution of autonomous IoT smart hive and optimization of energy consumption using artificial intelligence. Journal of Process Management and New Technologies, 12(1–2), Article 49567. https://doi.org/10.5937/jpmnt12-49567

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection (No. arXiv:2004.10934). arXiv. https://doi.org/10.48550/arXiv.2004.10934

Charisis, C., & Argyropoulos, D. (2024). Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges. Smart Agricultural Technology, 8, 100448. https://doi.org/10.1016/j.atech.2024.100448

El Akrouchi, M., Mhada, M., Gracia, D. R., Hawkesford, M. J., & Gérard, B. (2025). Optimizing Mask R-CNN for enhanced quinoa panicle detection and segmentation in precision agriculture. Frontiers in Plant Science, 16. https://doi.org/10.3389/fpls.2025.1472688

Guerriero, A., Pietrantuono, R., & Russo, S. (2023). Iterative assessment and improvement of DNN operational accuracy. arXiv. https://doi.org/10.48550/arXiv.2303.01295

Haddaoui, S., Khlifa, N., Chikhi, S., Varastehpour, S., & Adjailia, F. (2024). A Comprehensive Review of Beekeeping Datasets for Precision Apiculture Research. 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT), 2043–2048. https://doi.org/10.1109/CoDIT62066.2024.10708308

Hernandez, J., Maisonnasse, A., Cousin, M., Beri, C., Quintrec, C. L., Bouetard, A., Castex, D., Decante, D., Servel, E., Buchwalder, G., Brunet, F., Feschet-Destrella, E., De Bellescize, K., Kairo, G., Frontero, L., Pédehontaa-Hiaa, M., Buisson, R., Pouderoux, T., Aebi, A., & Kretzschmar, A. (2020). ColEVaL: Honeybee COLony Structure EVALuation for field surveys. Insects, 11(1), 41. https://doi.org/10.3390/insects11010041

Karypidis, E., Mouslech, S. G., Skoulariki, K., & Gazis, A. (2022). Comparison Analysis of Traditional Machine Learning and Deep Learning Techniques for Data and Image Classification. WSEAS TRANSACTIONS ON MATHEMATICS, 21, 122–130. https://doi.org/10.37394/23206.2022.21.19

Mohseni, H. M., Nameghi,E. N. M., & Khalaj, M. (2022). Investigating the tendency of customers to adopt Internet of Things devices based on the development of technology acceptance model (Case Study: beekeeping industry). DOI: 10.30845/ijbss.v13n5p4

Olaniyi, E., Chen, D., Lu, Y., & Huang, Y. (2022, April 10). Generative Adversarial Networks for image augmentation in Agriculture: A Systematic review. arXiv.org. https://arxiv.org/abs/2204.04707

Robinson, I., Robicheaux, P., Popov, M., Ramanan, D., & Peri, N. (2025, November 12). *RF-DETR: Neural Architecture Search for Real-Time Detection Transformers*. arXiv.org. https://arxiv.org/abs/2511.09554

Ruvinga, D., et al. (2023). Queen bee presence detection using acoustic signals and deep learning methods. Journal of Insect Science, 24(2), 105–117.

Srinivas, A., Lin, T., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021, January 27). Bottleneck transformers for visual recognition. arXiv.org. https://arxiv.org/abs/2101.11605?fbclid=IwY2xjawL0oVJleHRuA2FlbQIxMABicmlkETFra1NnZ0R5SXBNcHdmNkNlAR4uVxIFhL8KIrw2IBbSirNb6NkBVDk1JzbsY36a0WqYIGG5ycIBahU_rkdRKw_aem_8dyKUqD_0AQEzCEq98jt4A

Taye, M. M. (2023). Theoretical understanding of convolutional neural network: concepts, architectures, applications, future directions. Computation, 11(3), 52. https://doi.org/10.3390/computation11030052

Thakur, H. K. (2021). Research Methodology in Social Science. Research Gate, 175. https://www.researchgate.net/publication/353430802_Research_Design

Yıldız, B. İ., & Karabağ, K. (2025). Deep Learning Approaches for Image-Based Classification of Honey Bee (Apis mellifera) Lineages. Türkiye Tarımsal Araştırmalar Dergisi, 12(2), 224–230. https://doi.org/10.19159/tutad.1696120