

BeeGuardAI : Embedded system for detecting Asian hornets and monitoring bee activity using AI

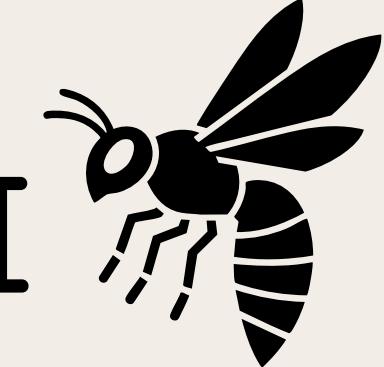


2025/2026





Team BeeGuardAI



YANN DOUZE

YASSER BOUHAI

YANELLE BEKKAR

GHOZLENE HANAFI

HADRIEL RATIARISON

AMINE NAIT SI AHMED

THEME: Embedded system for detecting Asian
hornets and monitoring bee activity using AI

PROBLEM STATEMENT



CONTEXT:

BEE COLONIES ARE ESSENTIAL FOR POLLINATION BUT ARE THREATENED BY THE ASIAN HORNET (*VESPA VELUTINA*).

CURRENT CHALLENGES:

- REPEATED HORNET ATTACKS ON HIVES
- PROGRESSIVE WEAKENING OF COLONIES
- DIFFICULTY IN CONTINUOUS MONITORING
- LATE DETECTION OF THREATS





SOLUTION

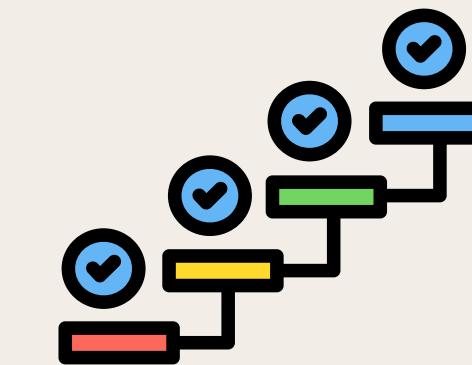


HOW WE SOLVE THE PROBLEM:

BEEGUARDAI IS AN AUTONOMOUS EMBEDDED SYSTEM COMBINING COMPUTER VISION AND AUDIO ANALYSIS POWERED BY TINYML TO MONITOR BEEHIVES 24/7. THE SYSTEM DETECTS ASIAN HORNETS, TRACKS BEE TRAFFIC, AND ANALYZES COLONY HEALTH THROUGH ACOUSTIC PATTERNS, PROVIDING REAL-TIME ALERTS TO BEEKEEPERS.

HOW IT HELPS SOLVE THE PROBLEM:

- EARLY DETECTION OF HORNET ATTACKS BEFORE COLONY DAMAGE
- CONTINUOUS MONITORING WITHOUT MANUAL INTERVENTION
- DATA-DRIVEN INSIGHTS FOR PROACTIVE COLONY MANAGEMENT
- REMOTE ACCESS VIA WEB DASHBOARD FOR MULTIPLE HIVES



IMPACT METRICS:

- HORNET DETECTION ACCURACY (TARGET: >90%)
- BEE COUNTING PRECISION (IN/OUT TRAFFIC PER HOUR)
- COLONY HEALTH STATUS CLASSIFICATION ACCURACY
- RESPONSE TIME FROM THREAT DETECTION TO ALERT (<5 MINUTES)
- SYSTEM UPTIME AND BATTERY AUTONOMY





SOLUTION

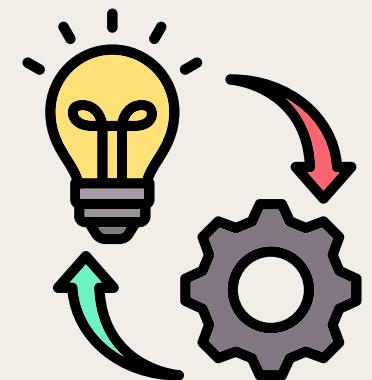
TECHNOLOGY STACK:

- HARDWARE: ESP32-CAM, ARDUINO NICLA VISION, MICROPHONES
- AI FRAMEWORK: EDGE IMPULSE (TINYML OPTIMIZATION)
- MODELS: YOLO-BASED OBJECT DETECTION, CNN/RNN FOR AUDIO CLASSIFICATION
- COMMUNICATION: LORAWAN (LOW POWER, LONG RANGE)
- BACKEND: INFLUXDB, NODE-RED / GRAFANA
- REASON: TINYML ENABLES ON-DEVICE INFERENCE WITH MINIMAL POWER CONSUMPTION; LORAWAN PROVIDES RELIABLE CONNECTIVITY IN REMOTE LOCATIONS



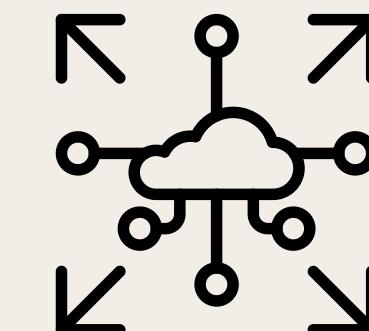
IMPLEMENTATION & EFFECTIVENESS:

- EASY DEPLOYMENT: MODULAR DESIGN ATTACHABLE TO ANY HIVE
- LOW COST: OPEN-SOURCE SOFTWARE + AFFORDABLE HARDWARE (<€150/UNIT)
- PROVEN TECHNOLOGY: TINYML SUCCESSFULLY USED IN WILDLIFE MONITORING



SCALABILITY/USABILITY:

- SCALABLE TO NETWORKS OF 100+ CONNECTED HIVES
- MULTI-USER DASHBOARD FOR APICULTURE COOPERATIVES
- ADAPTABLE TO OTHER PEST DETECTION SCENARIOS



THREAT RESPONSE SYSTEMS

AUTOMATED ELIMINATION SOLUTIONS

ONCE ASIAN HORNETS ARE DETECTED BY THE AI SYSTEM, TWO RESPONSE MECHANISMS CAN BE ACTIVATED TO PROTECT THE COLONY:

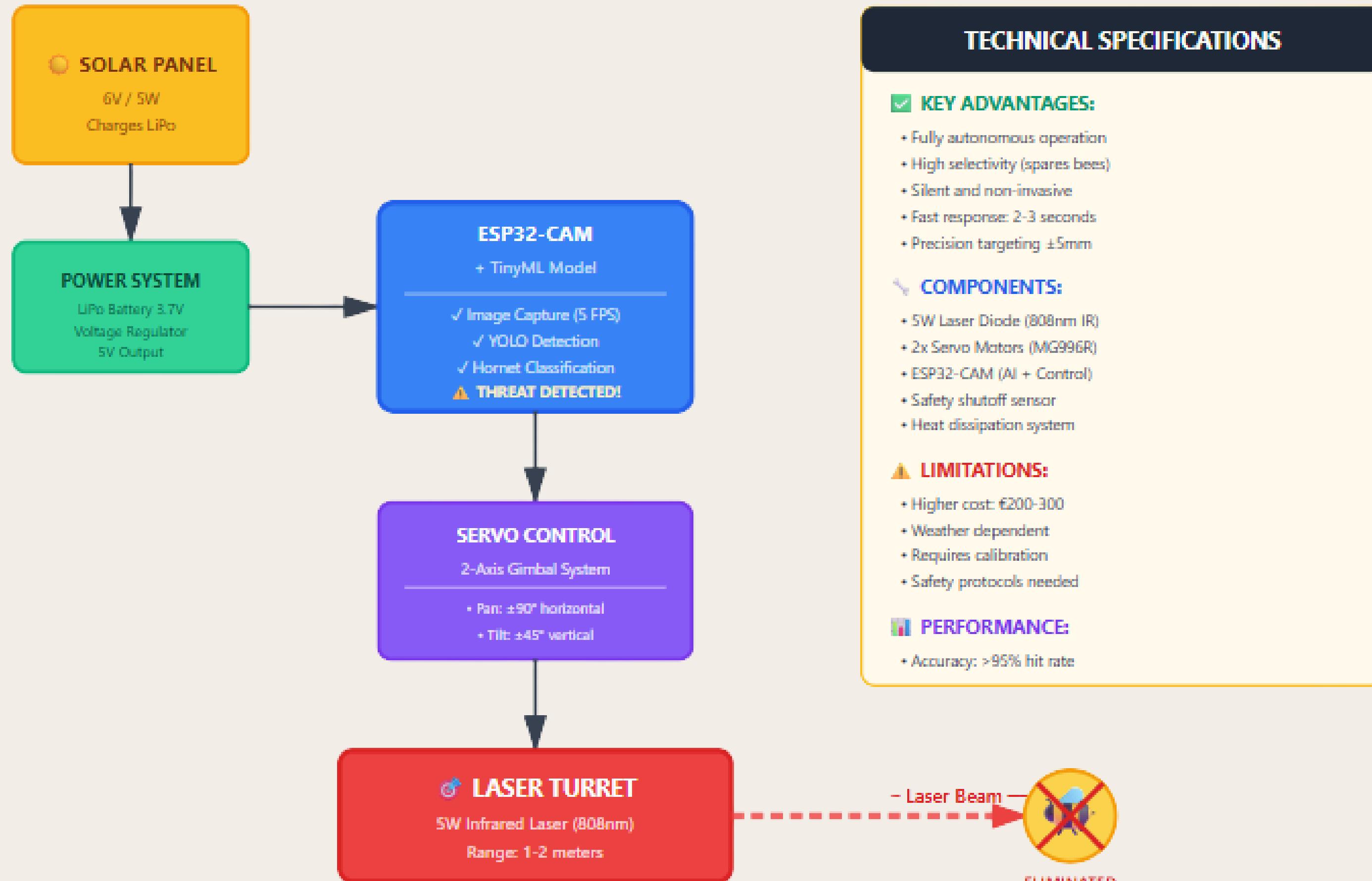
SOLUTION 1: LASER-BASED ELIMINATION

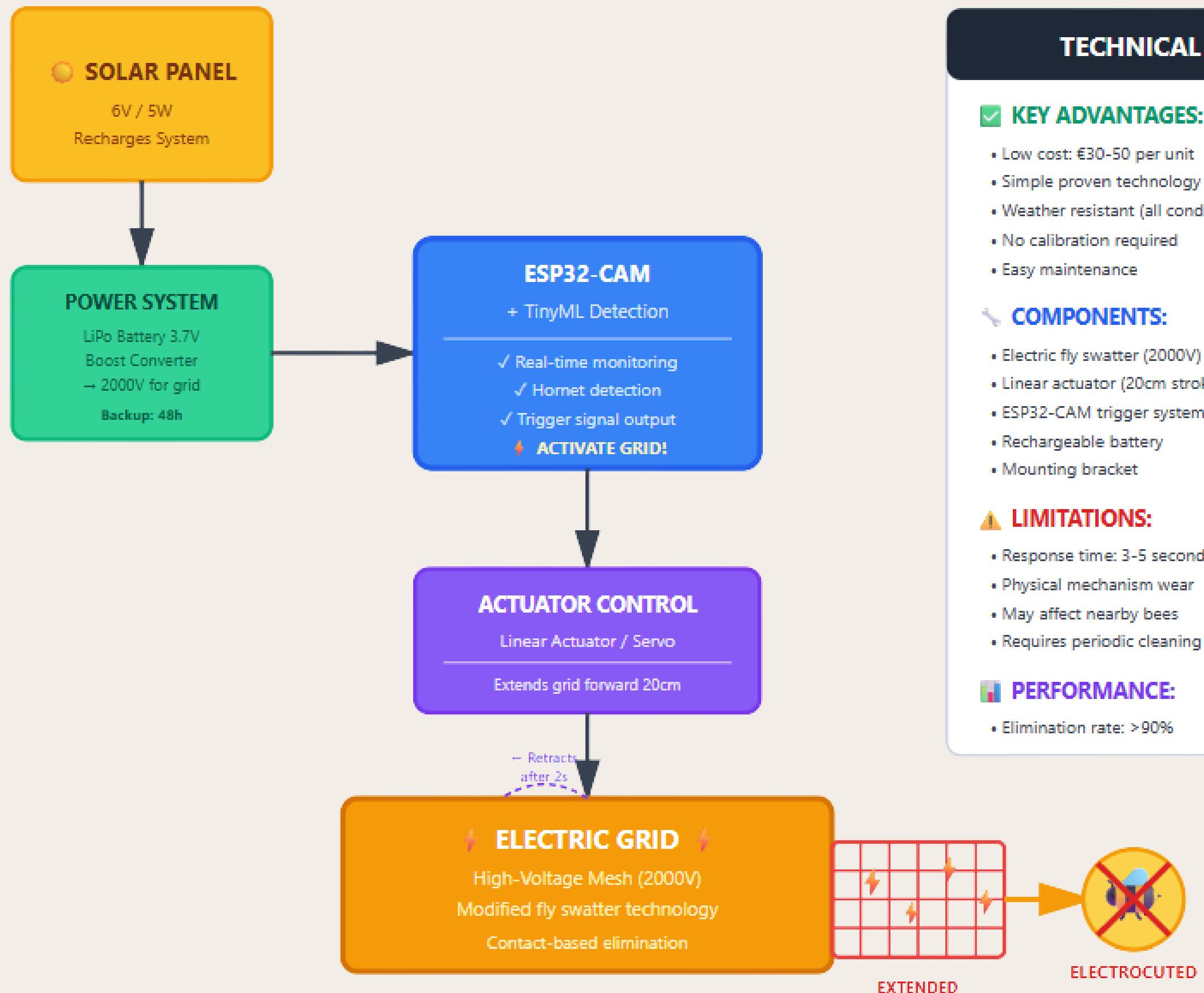
- HIGH-PRECISION TARGETING SYSTEM
- AUTONOMOUS OPERATION
- IDEAL FOR PROFESSIONAL APIARIES

SOLUTION 2: AUTOMATED ELECTRIC RACKET

- CONTACT-BASED ELIMINATION
- COST-EFFECTIVE SOLUTION
- SUITABLE FOR HOBBYIST BEEKEEPERS

BOTH SYSTEMS INTEGRATE SEAMLESSLY WITH THE BEEGUARDAI DETECTION PLATFORM AND PROVIDE REAL-TIME ALERTS VIA THE DASHBOARD.





TECHNICAL SPECIFICATIONS

KEY ADVANTAGES:

- Low cost: €30-50 per unit
- Simple proven technology
- Weather resistant (all conditions)
- No calibration required
- Easy maintenance

COMPONENTS:

- Electric fly swatter (2000V)
- Linear actuator (20cm stroke)
- ESP32-CAM trigger system
- Rechargeable battery
- Mounting bracket

LIMITATIONS:

- Response time: 3-5 seconds
- Physical mechanism wear
- May affect nearby bees
- Requires periodic cleaning

PERFORMANCE:

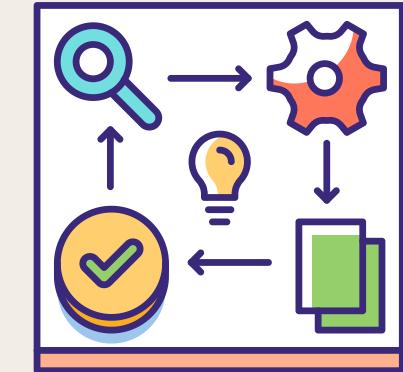
- Elimination rate: >90%

METHODOLOGY



CONCEPTS & PRINCIPLES:

- EDGE COMPUTING: AI INFERENCE ON-DEVICE TO MINIMIZE LATENCY AND BANDWIDTH
- MULTI-MODAL SENSING: COMBINING VISION + AUDIO FOR ROBUST DETECTION
- LOW-POWER IOT: LORAWAN FOR ENERGY-EFFICIENT LONG-RANGE COMMUNICATION
- EVENT-DRIVEN ARCHITECTURE: SYSTEM ACTIVATES ON MOTION/SOUND TRIGGERS



SYSTEM COMPONENTS:

1. DATA ACQUISITION LAYER

- ESP32-CAM (320X240PX IMAGES, 5 FPS)
- PDM/I2S MICROPHONES (16KHZ SAMPLING)
- ENVIRONMENTAL SENSORS (TEMPERATURE, HUMIDITY)

2. PROCESSING LAYER

- VISION PIPELINE: IMAGE PREPROCESSING → YOLO DETECTION → CLASSIFICATION
- AUDIO PIPELINE: MFCC FEATURE EXTRACTION → CNN CLASSIFIER → HEALTH STATUS
- TINYML MODELS (<200KB) OPTIMIZED FOR ESP32



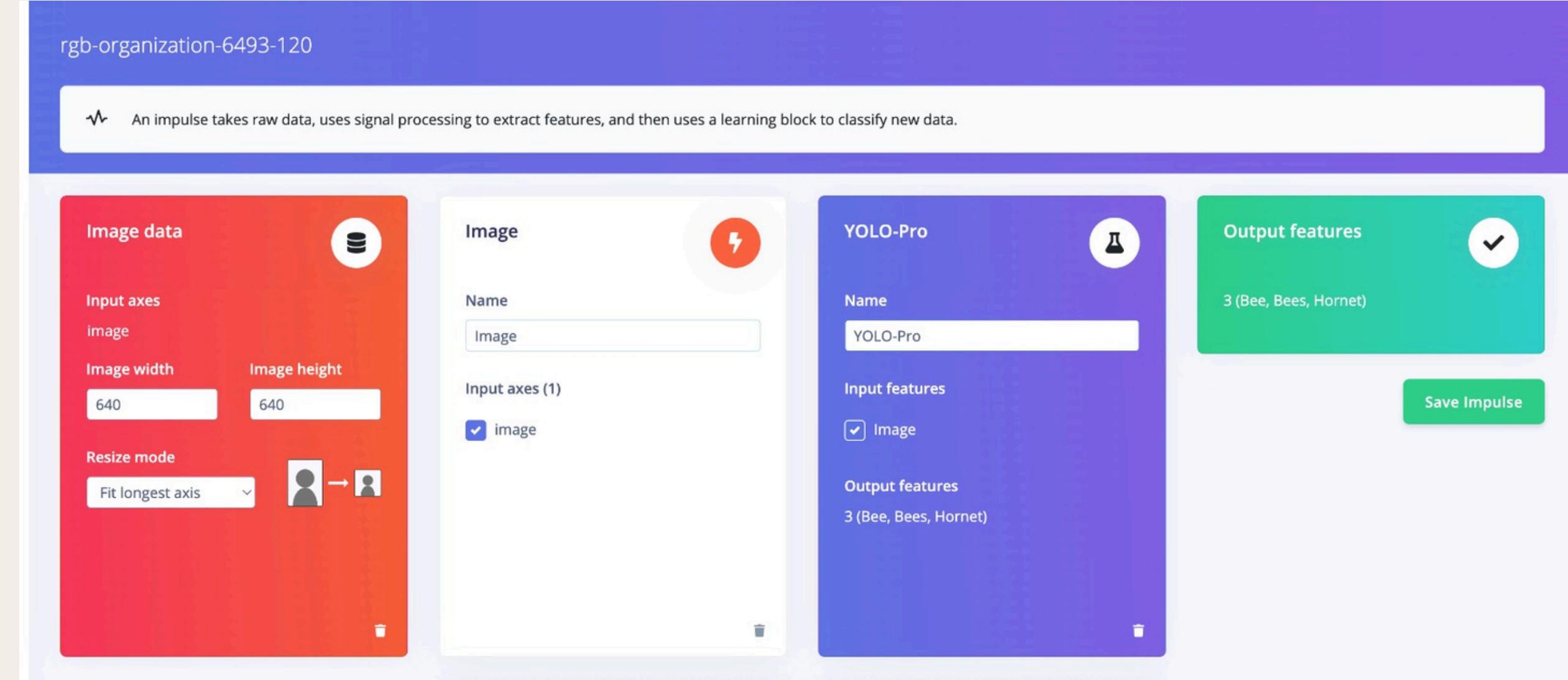
3. COMMUNICATION LAYER

- LORAWAN GATEWAY (868 MHZ EU BAND)
- PAYLOAD: DETECTION COUNTS + TIMESTAMPPS (12 BYTES/MESSAGE)
- TRANSMISSION FREQUENCY: EVERY 15 MINUTES OR ON ALERT

4. APPLICATION LAYER

- TIME-SERIES DATABASE (INFLUXDB)
- DASHBOARD (GRAFANA/NODE-RED) WITH REAL-TIME CHARTS
- ALERT SYSTEM (EMAIL/SMS NOTIFICATIONS)

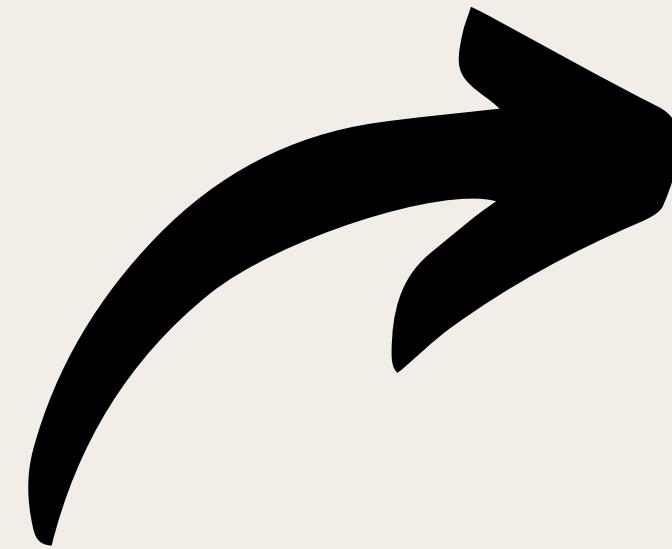
Edge Impulse Pipeline Configuration



PIPELINE OVERVIEW:

- **Image Input:** 640x640px, resized to fit longest axis
- **Feature Extraction:** Image processing block for object detection
- **Model:** YOLO-Pro neural network
- **Output:** 3 classes (Bee, Bees, Hornet)

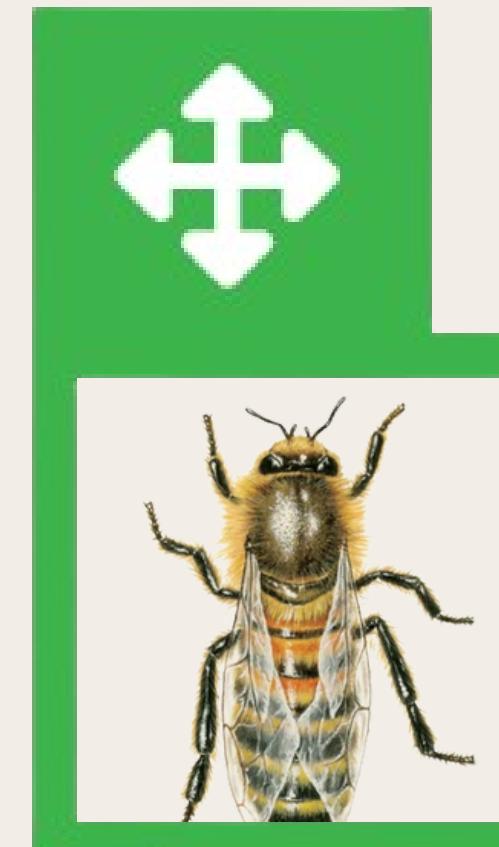
AI labeling



Hornet

Labeling with
bounding boxes

2 classes



Bees

AI labeling

Labeling steps :



Creation of
bounding boxes
(OWL-Vit object
detector)



Boxes
re-labeling
(GPT-4o)

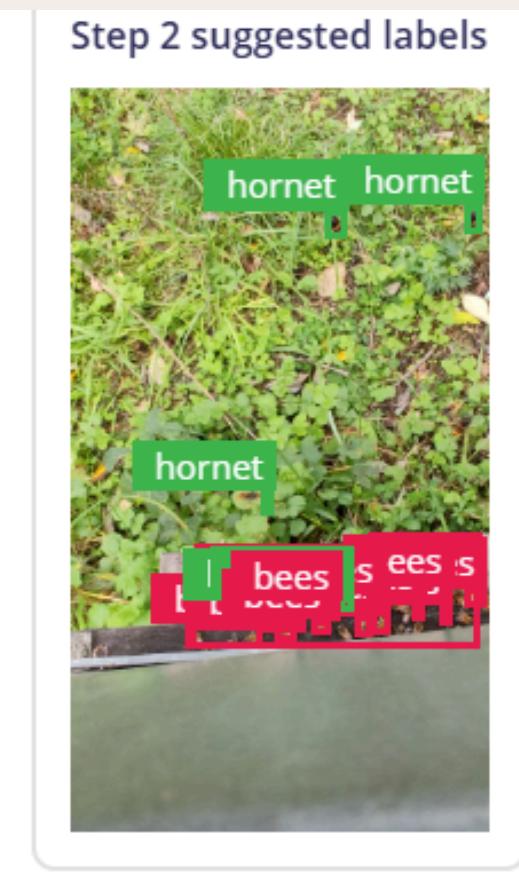
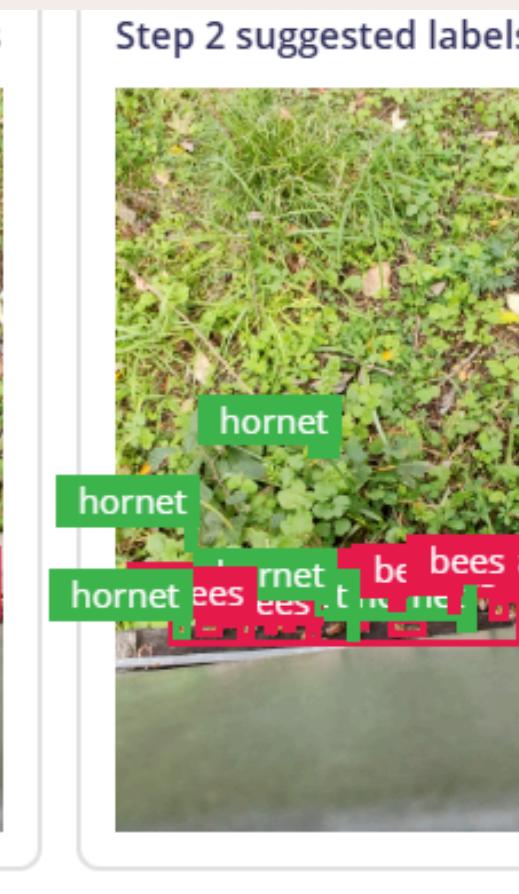
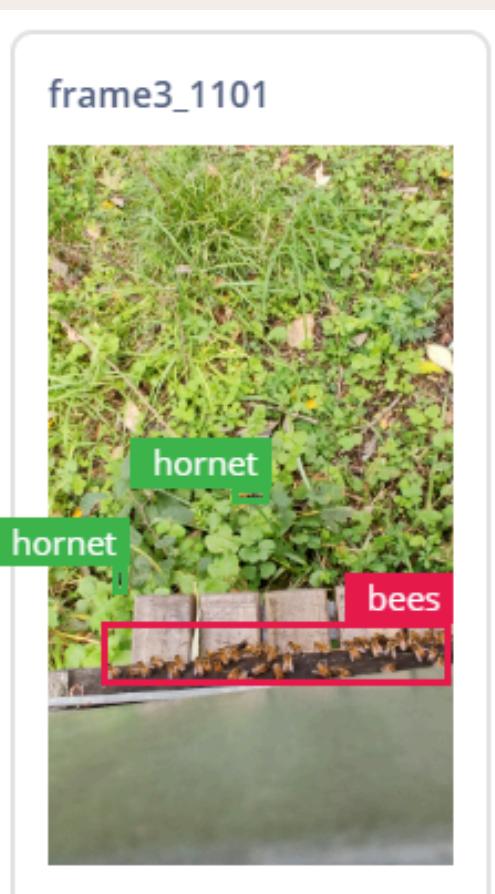
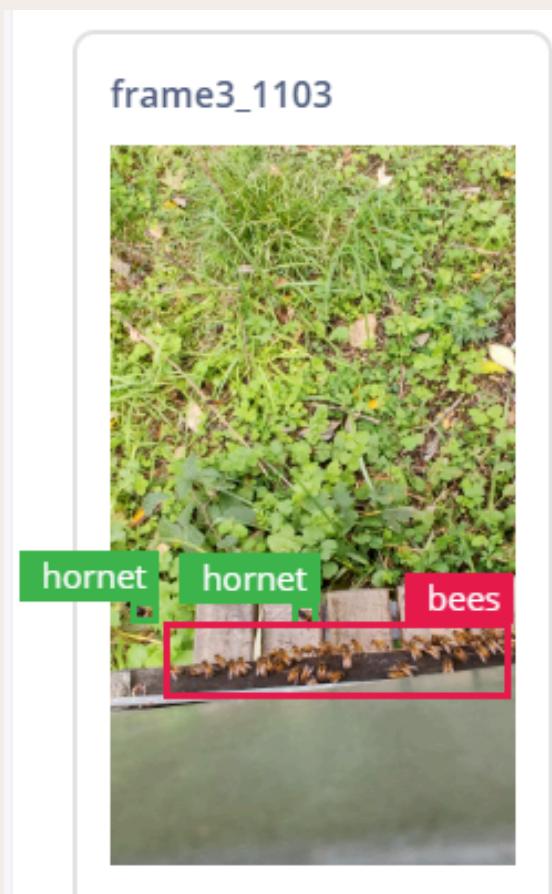
AI labeling

Comparison :

Ground truth
(manual labeling)

VS

Labeling using AI



Analysis & Visualization

YOLO-Pro Training Configuration

YOLO-Pro model training parameters: 100 cycles, 0.001 learning rate, GPU, with pretrained weights. The model achieves a precision score of 62.7% on the validation set with an mAP of 0.12

The screenshot displays the YOLO-Pro Training Configuration interface. On the left, the 'YOLO-Pro settings' panel shows training parameters: 'Training processor' (GPU), 'Number of training cycles' (100), 'Learning rate' (0.001), 'Model size' (pico (682K)), and 'Use pretrained weights' (checked). Below this is the 'Neural network architecture' section, which includes an 'Input layer (1,228,800 features)' diagram and a 'YOLO-Pro DEVELOPER PREVIEW' button. A dashed box labeled 'Choose a different model' is overlaid on this section. At the bottom is an 'Output layer (3 classes)' diagram. On the right, the 'Training output' panel shows 'Model version: Quantized (int8)'. The 'Last training performance (validation set)' section displays a precision score of 62.7%. The 'Metrics (validation set)' table provides detailed performance metrics:

Metric	Value
mAP	0.12
mAP@[IoU=50]	0.53
mAP@[IoU=75]	0.00
mAP@[area=small]	0.09
mAP@[area=medium]	0.16
mAP@[area=large]	0.18
Recall@[max_detections=1]	0.11
Recall@[max_detections=10]	0.16
Recall@[max_detections=100]	0.18
Recall@[area=small]	0.15
Recall@[area=medium]	0.30
Recall@[area=large]	0.21



YOLO-Pro Test Results

Test data

Classify all

Model testing output

(0)

Results

Model version: Unoptimized (float32)

ACCURACY 50.00%

Metrics for YOLO-Pro

Metric	Value
mAP	0.18
mAP@[IoU=50]	0.58
mAP@[IoU=75]	0.06
mAP@[area=small]	0.34
mAP@[area=medium]	0.14
mAP@[area=large]	0.23
Recall@[max_detections=1]	0.18
Recall@[max_detections=10]	0.22
Recall@[max_detections=100]	0.22
Recall@[area=small]	0.38
Recall@[area=medium]	0.14
Recall@[area=large]	0.26

Feature explorer

Validation results on the test set showing 50.00% accuracy with a mAP of 0.18. Detailed metrics display performance per class and object size.

Model Comparison with Different Latencies

Experiments EON Tuner Ghozlene Hanafi

Experiments Delete (1) Retrain (1) Test (1) + Create new impulse

<input type="checkbox"/> NAME	INPUT	DSP BLOCKS	LEARN BLOCKS	F32_V_ACC	F32_T_ACC	I8_V_ACC	I8_T_ACC	F32_LATENCY	F32_RAM	F32_FLASH	I8_LATENCY	
<input type="checkbox"/> Image detection model 2	160x160	Image	FOMO MobileNetV2 0.35	56.8%	8.5%	51.3%	-	3 ms.	1.1M	112.7K	622 ms.	:
<input checked="" type="checkbox"/> rgb-organization-6493-120	640x640	Image	YOLO-Pro	70.6%	50.0%	62.7%	-	X	-	-	29361 ms.	:
<input type="checkbox"/> Impulse #17	96x96	Image	FOMO MobileNetV2 0.35	54.3%	3.3%	51.6%	-	3 ms.	435.2K	112.7K	225 ms.	:
<input type="checkbox"/> model 3	96x96	Image	FOMO MobileNetV2 0.1	49.0%	8.9%	44.3%	-	3 ms.	399.7K	76.8K	212 ms.	:
<input type="checkbox"/> model 4 (run3)	320x320	Image	YOLO-Pro	65.9%	34.4%	55.7%	-	3 ms.	399.7K	2.3M	8065 ms.	:
<input type="checkbox"/> Model 5 (YOLO 96)	96x96	Image	YOLO-Pro	28.2%	44.4%	26.9%	-	3 ms.	836.3K	27.4M	6108 ms.	:
<input type="checkbox"/> Impulse model 6	160x160	Image	FOMO MobileNetV2 0.35	52.8%	5.6%	50.3%	-	3 ms.	1.1M	112.7K	621 ms.	:

FOMO Model

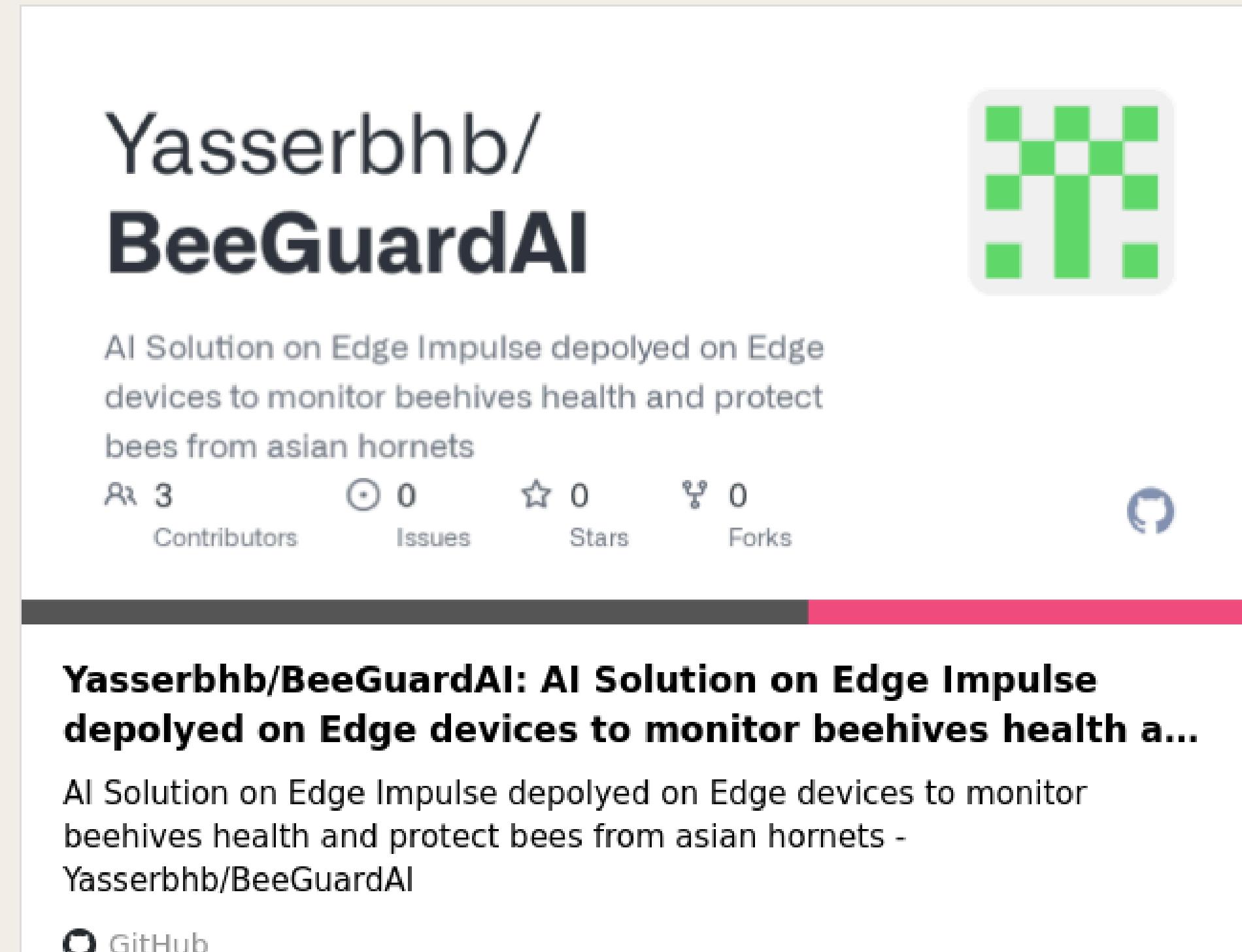
The screenshot shows the FOMO Model configuration and training output interface. On the left, the 'Neural Network settings' section includes 'Training settings' (Number of training cycles: 80, Use learned optimizer: checked, Learning rate: 0.001, Training processor: GPU), 'Data augmentation' (checked), and 'Advanced training settings'. Below these are sections for 'Neural network architecture' (Input layer: 76,800 features) and 'Output layer' (3 classes). In the center, the 'Training output' section displays the 'Model' (F1 SCORE: 50.3%), 'Last training performance (validation set)', 'Confusion matrix (validation set)' (with rows for BACKGROUND, BEE, BEES, HORNET and columns for BACKGROUND, BEE, BEES, HORNET), 'Metrics (validation set)' (Precision: 0.54, Recall: 0.47, F1 Score: 0.50), and 'On-device performance' (Engine: EON™ Compiler (RAM optimized), INFERENCING TIME: 618 ms., PEAK RAM USAGE: 321.1K, FLASH USAGE: 81.4K). A sidebar on the right shows a preview of the model's output.

FOMO (Faster Objects, More Objects) model configuration and results:

- F1-Score: 50.3%
- Confusion matrix showing classification of 3 classes (background, bees, hornet)
- Optimal on-device performance: 618ms inference, 321.1K peak RAM, 81.4K flash
- Much better suited for embedded systems than YOLO-Pro

Attachments

**Yasserbhb/
BeeGuardAI**



AI Solution on Edge Impulse depolyed on Edge devices to monitor beehives health and protect bees from asian hornets

3 Contributors 0 Issues 0 Stars 0 Forks

Yasserbhb/BeeGuardAI: AI Solution on Edge Impulse deployed on Edge devices to monitor beehives health a...

AI Solution on Edge Impulse depolyed on Edge devices to monitor beehives health and protect bees from asian hornets - Yasserbhb/BeeGuardAI

 GitHub



THANKS