Development of a mechanism for protecting beehives against Asian hornets based on image processing by supervised learning

Tessier Raphael, Goalard Rémi, Relandeau Antonin, Dupuy Pauline, Pinel Axel, Etienne Sylvain, Raecke Jonathan, directed by : Gwendoline Lecorre, Elodie Chanthery[1]

[a]*INSA Institut National des Sciences Appliquées de Toulouse, 135 avenue de rangueil, 31400 Toulouse,*

Abstract

Asian hornets have been a scourge for beekeepers in France since 2005. While Asian bees found a way of protecting themselves from this predator, french ones are being decimated. A few beekeepers tried to develop a system to protect their bees from asian hornets, but they all present issues. Some are killing all species, not only asian hornets, while others require a permanent connection to a power source. We decided to create an automated system to kill hornets threatening beehives, with two major constraints : identify asian hornets so they are the only ones that are killed, and create an embedded system, working with a battery. Those objectives resulted in a 3 steps system : detection of a danger near the hive; hornet recognition based on supervised IA; killing the hornet using a swatter. The frequency analysis result not to be effective, with only 53% danger recognition accuracy. With respect to the supervised hornet recognition, the results were more conclusive, with around 80% accuracy. Finally when the swatter is connected to the camera, it whips if an Asian hornet is detected.

Keywords: Beehive protection, Asian hornets, Embedded AI, Computer vision, Video Pre-processing, Multi-Layer Neural Network, Supervised learning, Hornet behaviour recognition

1. Introduction

Asian hornet, also called *Vespa Velutina* was found in France for the first time in 2005. This hornet species is a fearsome predator for European bees, which has not been able to create a way to defend itself. The *Vespa Velutina* population increased exponentially since their first apparition in Europe, resulting in a an important threat for local bees and bee-haves. This problem concerns beekeepers, who see their swarms decimated, but also all the population living in these territories, because bees play a major role in pollination and the maintenance of ecosystems. At this stage, beekeepers have not found any effective and ecological solution to protect their bees. Only 1% of the bugs captured by passive trap are Asian hornet [1], and actives traps are often based on substances that attract hornets as well as other insects, which endangers biodiversity. A French patent filed in 2019 describes a mechanism to locate a hornet via a sensor at the entrance of a hive, and to enclose it in a box of which it would be a prisoner [2]. Nevertheless, this system has a main defect, it requires a fairly large amount of energy, and requires to be permanently plugged in.

The objectives of this paper are therefore the development of a solution to eliminate Asian hornets present near hives, by creating an embedded system based on image processing by supervised learning. Those objectives resulted in a 3 steps system : First we use a frequency analysis of the beehive to detect the presence of danger. Then we take pictures of the area in front of the hive, and try to identify an hornet using a supervised IA. Last, if an Asian hornet was recognized in front of the hive, a swatter powered by a motor kills the hornet.

2. Literature review

Before starting to develop our beehive protection system, we looked for different projects or studies having similarities with the one we wanted to implement. This focused on two aspects. On the one hand the different solutions already existing, and on the other hand the options we could explore to solve our problem.

2.1. Using Artificial Intelligence for video detection: How does that work?

2.1.1. Two methods considered : Supervised Learning and Non-Supervised Learning

First, pattern recognition in video with supervised learning. The purpose of pattern recognition is to categorize some input data into different available classes. There are two main types of patterns that can be detected in videos: Spatial patterns, which deals with image identification and temporal patterns such as trajectory, speed and other motion analysis [3].

These two methods are often based on supervised learning, a subset of machine learning, which use labelled dataset to train algorithms in order to predict outcome of unlabelled data.There exist multiple models of supervised learning : interpretive, like decision tree and random forest; or not-interpretive, as neural network and support-vector machine [4].

Learning based on spatial analysis provides an automatic construction of the class models by the algorithm. It can be trained with non-sequenced images. However, the train model will be sensitive to variation in the appearance between samples, like the size, the colours impacted by the external luminosity, the background [5]. Temporal analysis avoids this appearance problems but can only be applied to in motion object, that could be a difficulty since the Vespa Velutina is less mobile than Apis Mellifera [5], [6].

The system could also quickly reach a very high size involving lots of operations, that could become a problem for real time detection [7], cf 1.2.

Second, unsupervised learning for anomalies detection and clustering. Unsupervised learning use unlabelled data to detect similarity and difference on the data. That is used in video analysis for clustering, which consist of regrouping data by similarity and making non-identified classes [8]. For this algorithm we need data sets with both Apis Mellifera and Vespa Velutina, but we do not have to label this data. However, we do not have a direct control on the created classes. Clustering is more often use in order to classify unknown data in an unspecified number of classes [9]. As we are looking for specific classification, this solution is not really a good one.

Another way to use unsupervised learning is the detection of anomalies. The purpose of this method is to identify abnormal data among a set of normal data. Anomalies detection algorithm are only trained with normal data and tested with both normal and unusual data [10]. In our case, the train set will only be composed of bees, and the test set of bees and hornets. That could be an advantage if our dataset is really unbalanced. Outlier detection can be done either with motion or appearance but cannot identify the type of the anomalies [10]. Therefore, as we need to identify specifically Asian hornets and no other anomalies like non-aggressive insects, outlier detection is not a good solution for our project.

After studying this two solutions and found some research in relation to our project [4]–[6], that always use supervised learning, we chose to focus our search on this method and the ways to applied it on an embedded system.

2.2. Embedded Artificial Intelligence

Computer vision and more globally Artificial Intelligence is very computation heavy. As it is achieved in real time it requires a lot of processing power and thus a lot of power. The trend in recent years has been focused on accuracy using deeper neural networks and cutting edge Graphics Processing Units [11]. The consideration for energetic efficiency and limited resources is very low. However, for our project to be relevant, we cannot use expensive hardware and execute energy consuming calculations on the spot. That is why it must be running on embedded systems if our application aims to monitor the hive for long periods of time on battery power only. While offloading computation to the cloud or a nearby systems is also a solution, it is more time-consuming to implement with no net improvement over embedded systems. Furthermore, sufficient network coverage for remote work and communication cannot always be guaranteed and is not as cost effective as FPGAs or General-Purpose systems [7].

2.2.1. Existing hardware solutions

Hardware implementation can be done on multiple platforms. Basic algorithms and neural networks for Computer Vision and Artificial Intelligence can easily be supported by general purpose systems such as Computation Processing Unit (CPU) or Graphical Processing Unit (GPU). Low-power versions of CPU and GPU are available for the purpose of embedded computer vision(e.g. Nvidia TX2) [12]. Recently new processing units specific for neural network application has been developed with embedded artificial intelligence in mind. This category includes Google's Tensor Processing Unit and Intel's Nervana Neural Network Processor. However, GPUs and GPUs come with drawbacks, the most noticeable being high latency.

Odrika Iqbal and his team chose Field Programmable Gate Arrays (FPGA) for the low latency, the programmability and the on-board memory because "Considering latency as our metric for evaluation, FPGAs have been known to come out ahead of both CPUs and GPUs for imaging applications” [13].Object Tracking implemented on an FPGA running at 75MHz of clock frequency, it is possible to achieve 26 milliseconds of latency. That corresponds to 38 frames per seconds of real time video processing [13].

Application Specific Integrated Circuit can also be mentioned as it is the pinnacle of hardware implementation for Artificial Intelligence. However these types of integrated circuits are the most expensive solution and requires a lot of development time due to their high complexity [12]. It is not a preferred solution as large volume is not required in the use case.

2.2.2. Software optimisations

According to Wang Ying “ Low Power Computer Vision needs the support of both the software and the hardware communities” [11]. Indeed, software optimisation plays a huge role in decreasing computing time, latency and power consumption.

The first optimisation worth looking at is Parameter Quantization and pruning. It mostly consists of reducing the size of the parameters in memory. Memory access represent a considerate amount of time, reducing the storage space of the parameters helps computing time but reduce accuracy. According to the paper “A Survey of Methods for Low-Power Deep Learning and Computer Vision”, accuracy of the neural network is not significantly affected by the shrinking of storage space when using 8bit, 6bit or 4bits numbers [7].In some cases, single bit coefficient may be used to further decrease access time, however the neural network must have a lot more layers to maintain the same level of accuracy. The main drawback to this method is the cost of training, the neural network needs to be trained multiple times with different combinations of storage optimisation.

The second software optimisation is Knowledge Transfer. The key principle of the method is that a smaller neural network can be trained to mimic a more complex neural network. It requires a very optimised training for the teacher algorithm and the construction of a similar but smaller student neural network [7]. The loss Knowledge Transfer accuracy is not significant, however the training process and the neural network design takes muwh more time. Moreover, the input and output layers cannot be different for the master and the student. A variant of this method is creating smaller networks to mimic one feature of the master network and combining several students to achieve the desired output [7]. This is known as Knowledge Distillation.

Such methods can be implemented easily on an STM32 microcontroller. They stand under the general-purpose systems category as a microcontroller is a small CPU with embedded Random Access Memory and Flash memory. Neural Network Approximators are simple enough to be executed from a STM32F401RE with low deviation [14]. An STM32F401RE can handle neural networks and multi-layers perceptron with a number of neurons ranging from 15 to 4000 [14]. An STM32 micro controller with a quicker clock frequency and more memory should be able to run neural networks with more layers.

2.2.3. *Final thoughts on embedded Artificial Intelligence*

Embedded artificial intelligence and computer vision implies lots of constraints. Hardware selection is crucial because it heavily influences the choice of algorithms and parameters of an application. While CPUs and GPUs are cheaper, they are the slower solution and limit the numbers of neurons, layers and parallel processing done in a fixed amount of time. FPGAs are quicker and almost hardwired to the application. The main drawback is design time as FPGA are more difficult to program efficiently than the general-purpose equivalent. With fixed hardware, computing time and energy efficiency still needs to be fine-tuned with software optimisation. Parameter quantisation brings down the computing time by a large factor without sacrificing too much accuracy and knowledge transfers enable power savvy neural networks for a same system

2.3. *Supervised learning applied to our project*

2.3.1. *Assembling a dataset: a crucial step in our process*

The purpose of a database for supervised learning is to provide the machine learning algorithm with a set of labelled examples that can be used to train the model and make predictions about new, unseen data. Supervised learning algorithms require labelled data to learn the relationship between the input features and the target variable [15]. The database of labelled examples serves as the training data for the model, allowing it to learn and make predictions based on the patterns and relationships it discovers in the data.

We can also test our model by using a testing set. Of course, this set must contain only data which has not been used in the training of the IA.

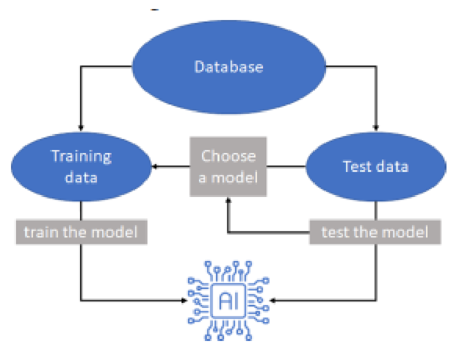

Figure 1: Supervised learning and database[1]

A well-designed and curated database of labelled examples is critical for the success of our supervised learning model. The quality and diversity of the labelled data in the database will directly impact the accuracy and robustness of the model's predictions. If the database contains high-quality, diverse, and representative examples, the machine learning model will be better able to generalize to new data and make accurate predictions.

The choice of the data collected depends on the model we want to build. For example, if we want a model that can recognize bees by their look (colour, size, shape, . . .), we can use pictures. However, if the model is based on the hive activity (noise, trajectory of bee, number of bees outside the hive, behaviour, ...), pictures are not enough. We will probably need videos and sound recording too. To be sure to get relevant data, the best solution is still to accumulate as much data as possible.

The creation of a database is composed of

several steps [15]:
Data Collection: The first step is to collect a large and diverse set of data on bees and hornets. We need as much data as possible if we want to construct a correct model.
Data Preprocessing: Once the data is collected, it is important to pre-process it to ensure that it is in a suitable format for machine learning. This may involve resizing the images, converting them to grayscale, normalizing the pixel values, and splitting the data into training, validation, and test sets.
Data Labelling: The next step is to label the data. This involves manually annotating the images to bring out the bees and the hornet on the picture.
Model Development: The next step is to develop a machine learning model that can effectively classify bees based on the images. This may involve training a deep neural network or using a pre-trained model and fine-tuning it for the specific task.
Model Evaluation: Finally, it is important to evaluate the model to determine its performance and identify any areas for improvement.

2.3.2. Collecting data

For this project, depending on the method chosen, we need data on bees, specifically the apis mellifera specie and on Asian hornets (vespa velutina). There are many ways to collect/assemble data. Using online image databases, such as Flickr [16] or Wikimedia [17] commons, can be a useful source of bee data. While this approach does not provide as much information as field observations or crowdsourcing, it can still provide a large number of images that can be used to train machine learning models. One problem with online databases is the accuracy of the results. Indeed, the search for ”Asian hornet” returns hornets in general, and even other insects such as bumblebees, bees... Our search is too precise for online databases. It is also difficult to isolate a species of bees.
Another approach is to use crowdsourcing platforms, where many people can contribute to the data collection effort. This can be done by creating a platform where users can upload images of bees they have observed and provide relevant information about the species and location. It was the case for research in Washington state where civilian sent pictures of Asian hornet to fight against its invasion [18]. Because of a lot of wrong report, it was quite difficult for the scientific to use the data given by the volunteers. For our project, this solution is not relevant.

Participating in bee monitoring programs or citizen science projects can also provide a wealth of data on bees. These programs typically involve trained volunteers collecting data on bee populations and their habitats, which can then be used to develop a database for machine learning.

Another solution, and perhaps the best, are field observations. This involves visiting different environments where these insects are found and capturing images of them as they interact with their environment. This is very relevant because, depending on the method we use, we need to be able to study the activity of a hive, the trajectory of the bees, all things that simple photography does not allow. Some research about AI and apiculture have already used this method [6].

Gathering data using fields observations on bees seems relatively easier than on hornets. Indeed, bees cohabit with humans and are even domesticated. We can, for example, contact beekeepers to place a camera near the hive to film the activity of the insects.

About the Asian hornets, we don't need access to a nest. We need hornet photos or beehive attack videos. We can capture hornets using already verified methods. In research [19], scientific found a way to attract hornet using the female sex pheromone. Their method is highly dependent on the weather but can be a research track for our project.

2.3.3. Image and video pre-processing

The pre-processing of the images in the database is necessary. It allows us to have usable and good quality images in order to train our artificial intelligence correctly. There are many image pre-processing techniques: cropping, scaling, colour processing, blurring...

The images in the database are often unsuitable for processing because there are many constraints on collecting images of bees.
Firstly, there are a large number of bees which are all similar, yet their appearance changes enormously depending on what they are doing, their orientation, the position of the sun, etc.
Secondly, the bees are very small (between 1.2 cm and 0.6 cm) and are found in a 3-dimensional environment, which makes their detection even more complicated.
In addition, the background also plays a role in the recognition of the bees, so to reduce errors, it may be interesting to include a panel.
Finally, the flight speed of bees can be as high as 8 m/s, so if we want to have an accurate representation of their trajectory, we need an average image acquisition frequency of 24 Hz [5].

In other research [4], scientists have noted that shadows can interfere with bee detection. When counting the number of bees in an image, their shadow can also be considered, which would double the number of bees in the image. The solution they found is to detect the blue saturation of the image. The natural light is bluer than the bees which are yellow and brown. It is therefore easy to distinguish the bees from their shadows with a simple colour filter.

Artificial intelligence research on agriculture [6] uses trajectory detection using two IR cameras that provide a 3-D image. These cameras allow the acquisition of typical trajectories of bees and hornets. To do this, there must be data processing so that external objects do not distort the 3D trajectories.
First, there is a valid flight depth margin implemented by a binary mask.
Second, the "flying insects which are tracked using motion-based multiple objects tracking algorithm based on Kalman Filter with state vector [*u, v, d*], where '*u*' and '*v*' are image coordinates and '*d*' is the depth value".
Finally, each filmed trajectory is assigned to an insect species. In this manner, the researchers were able to identify the "short-lived and simple trajectory of Apis mellifera indicates the directed movements of honeybees toward or away from the beehive", which is very different from "the trajectory of Vespa Velutina is longer and more complex due to the roaming and hovering of this species in front of the beehive".

2.3.4. Labelling the dataset with the processed images

Once we have prepared our database, we need to label it. In his thesis, Guillaume Chiron proposes to label hornets and bees using the triple blind method. Indeed, the two insects can sometimes not be well distinguished by humans. The triple-blind method allows to "limit the biases linked to human errors of appreciation" because each image is annotated by three different people. This requires three times more work, but the accuracy of the data is much better [5].

2.3.5. *Existing solution for hornets recognition in videos*

There are a few solutions for supervised Artificial Intelligence recognition of Asian hornets and we are going to describe them in this part. To achieve the recognition of Asian hornets among honeybees, Red Green Blue (RGB) camera is often used with an InfraRed sensor. The IR sensor is used to provide depth to the simple 2-Dimension RGB frames, and it also allows the trajectory processing of the bees [5]. These two devices allow a more precise classification. It is mandatory to elaborate a dataset with good data to classify each RGB frame. Some of the prior scientific research [5] used the help of people online to identify bees, by developing a quick app. People had to surround bees in an ellipse frame by frame in order to draw their trajectory and distinguish them from the background. To avoid human mistakes, they used what they call a triple blind annotation, it means that three different persons classified the same picture or video fragment, and the data was considered correct if at least two of them agreed in following the bee.
Another study was supervised by an entomologist to prevent mismatches in the dataset [6].

To label the sets they mostly used the RGB frames since the IR content is insufficient for human perception. Capturing small and fast-moving insects like bees or hornets also causes photographic issues based on the technologies of the cameras. For instance, the frames can suffer of the rolling shutter effect, leading to a distortion of the straight lines in the RGB frame. Therefore, 3D trajectory data provided by the IR sensor has been used to compute 3D velocity components and predict the position offset of the insect [6]. All datasets were built during two consecutive summers, and considering different situations of light, weather, temperature, humidity and global behaviour of the beehive during the various sessions of recording [5], [6]. They divided the dataset into 3 parts: 70% for training the algorithm, 15% for validation and 15% for the testing data. They used a neural network with the method of the stochastic gradient, but they only used the IR data to do so. Then they also did a 3D-trajectory recognition of the insect species and they used both predictions to make a final one.

The bee/hornet recognition algorithm achieved 92.7% precision for bee recognition, and 93.5% precision for the Asian hornet, only based on the IR imagery. When they added the 3D trajectory, they achieve a precision rate of 98.2% for the bee recognition and 94.5% for the Asian hornet [6].

2.4. *Supervised learning applied to our project*

Based on the already existing studies on Asian hornets recognition with artificial intelligence, we surely know we would be able to implement a supervised model on an embedded system to identify a not moving hornet in front of a hive. However, and despite the very high precision level reached, having one or more

cameras filming the entry of our hive 24hours a day would cause some energy issues. Indeed, the use of an embedded system, therefore not connected to the electrical network, causes the need to create a solution that consumes little electricity. So here is our new problem: can we detect the possible presence of a hornet without going through visual recognition with AI, then activate the camera and check if it is indeed a pest?
We found a few ideas based on bees/hive or Asian hornet characteristics.

2.4.1. Frequency analysis of the hive

The first idea would be to check at bees or hive natural behaviour, and look for a modification/abnormal one to deduce the possible presence of a hornet-like hazard.

Bees are a subject on which a large number of studies have been carried out, including some on the ways they use to communicate. It has been shown that, like many animals, they use the frequency of the vibrations they produce with their wings in order to convey messages or indicate the activities they are performing [4]. Despite the information that the behaviour of each individual can teach us, it is very complicated for us to try isolating the frequency emitted by one particular bee, even more considering it is within the hive.

Nevertheless, a study published on January 5, 2022 by Hadjur Hugo took up this study characteristic, but extending the system to the entire hive [4]. By studying the audio and frequency responses of the beehive, researchers showed that the frequency of the hive is always around a few hundred Hertz, except in the presence of danger. In this case, the frequency can vary between 0 and several thousand Hertz, ”showing a dominant frequency of 6 kHz and harmonics reaching 16 kHz” [4].

Knowing this information, we could decide to set up a frequency sensor permanently listening to the beehive, and thus indicate the presence of a danger. If so, the camera could take over and carry out a visual recognition of intruders if some are threatening the hive. Moreover, if the danger turns out not to be Asian hornets, it is possible to imagine an alert system sent to the beekeeper, so that he can go and check what is the actual problem.

The advantage of this method would be to use two sensors in a complementary way, with the first one (frequency) consuming much less energy than non-stop video surveillance.

2.4.2. Thermal and audio sensing

The second direction in which it would be possible to look would be, not to be interested in the behaviour of bees in the face of the threat hornets represent, but rather to detect hornets presence directly, without necessarily going through video surveillance.

Among all the studies done on hornets, these are two characteristics that caught our attention.At first, and in a similar way to what has been seen on bees, studies have been carried out on the frequency of hornets wings while flying. Depending on the study, they found frequency between 85Hz and 115 Hz. The first one [20], made on “Vespa Crabro” hornets, found out “The mean frequency amounted to 97+/-13 Hz at 25°C and 85+/-11 Hz at 35°C with no significant differences between both temperatures. Wing-beat frequency of flying queens at 30°C ranged around 84+/-7 Hz”. Meanwhile, in the second study, done on 4 different types of hornet, they found frequency around 110 Hz or 115 Hz [21].

The other characteristic of hornets that could interest us was studied alongside their wing wave frequency by a team from the Zoological Institute of Berlin. Indeed, they studied "Vespa Crabro" hornets in order to determine their heat production while flying, and this considering the outside temperature [20]. They separated queens, drones and workers, and for drones they figured out that "The heat production rate showed a weak temperature dependence. Drones had no significantly different heat production rates at 20°C and 25°C (81.2+/-20.9 mW, n=9 at 20°C; 84.8+/-6.6 mW, n=8 at 25°C). The same holds true for the values at 30°C and 35°C (42.8+/-15.3 mW, n=19 at 30°C; 42.5+/-15.3 mW, n=7 at 35°C)" [20]. We could then try to put a calorimetric sensor in front of the hive, and detect the possible presence of an Asian hornet when a stable heat production value would be detected there. Then, like with the wing frequency captor, the camera and the IA recognition would take the lead to confirm the existence of an Asian hornet threatening the beehive.

These different options seem more or less implementable, in particular the last one, which seems quite complex to set up in real life. But they all are quite interesting because they open us up to various alternatives, leading to the creation of both an effective and low-consuming recognition system.

2.4.3. Computer vision without IA

It is possible to avoid having to use an AI model to recognize insects. This requires several successive image processing but is nevertheless possible.

French researchers have developed the HDIS segmentation method to detect bees from two cameras filming the entrance to the hive [5].

Their method is based on three main ideas. First, by subtracting images taken moments earlier from current images, it is possible to detect the appearance of motion. Then, by studying the colours of the areas in which the movement was detected and comparing the images from the two cameras (which makes it possible to create a kind of 3D view), the computer can calculate the sizes and orientation of the insects.

It remains a complex method to implement in view of the mathematical algorithms necessary to create this 3D vision, but it has paid off.

2.5. Hive protection strategies

To capitalise on the detection strategies for Asian hornets presented in the previous chapters a selective method of neutralising those must be implemented. Although most of the literature dealing with hornet detection focuses on approaches in computer vision rather than the neutralisation, there are roughly two categories of approaches to achieve the latter found in related areas of research: Termination using an aimed laser and trapping hornets in a container. These categories will be discussed in the following subsections. However, it can be concluded in advance that none of those has reached a level of technical maturity.

2.5.1. Laser firing at a target

The basic idea of using a laser to terminate any insect consists of using its thermal energy to damage the insect's cells with the ultimate goal of destroying a lethal amount of them. The general effect of laser-tissue interaction has been sufficiently studied as summarized for instance in [22], [23]. Thus, the possibility of causing lethal damage to organic cells is well established. However, Niemz et al. highlight the duration of laser exposure as the single most important parameter of the interaction process even before the laser's energy density [23]. This general observation draws attention to the importance of not only aiming correctly once but also keeping target for a necessary amount of time in order to neutralise a hornet.

Lasers have been used in different occasions to repel or neutralise other insects than hornets namely aphids [24], cockroaches [25] and mosquitos [26]. In each of the found studies challenges occurred based on the specific species.

While aphids do not move quickly, they are typically spread out over a larger area of interest (e.g., a row of plants in a greenhouse). Subsequently a mobile robot was built by Lacotte et al. [24] and tested under laboratory conditions. The neutralisation of detected aphids by a laser-beam that was aimed using two degrees of freedom actuated mirrors achieved the desired effect in 90% of all cases without harming the plants. Nevertheless, the authors summarise that especially the aiming would need major improvement for the robot to be useful in a realistic scenario.

In another study, a low-cost laser system automated by machine vision was presented and tested on cockroaches under laboratory conditions by Rakhmatulin et al. [25]. Although the applicability to hornets is limited by the fact that cockroaches only move in 2D and the highly simplified test scenarios, some interesting learning's can be drawn from it. Firstly, lasers of different power have been studied with the observation that a laser of insufficient power (here 300 mW) would make the cockroaches move too fast for the system to properly track them and fulfil the neutralisation. Thus, a more powerful laser (1600 mW) was found to be more efficient. Secondly, the exposure time was set to 0.5s to achieve the presented results. This means that tracking and aiming at a moving target can be estimated to have to persist for a similar period. Lastly, the authors emphasise that the presented system should not be used under circumstances where other living beings can be harmed, which presents an additional challenge if applied to a complex scenario as in front of a hive.

The closest use case to hornet neutralisation via laser-beam was presented by Kare et al. [26] who present a photoelectric barrier that can hit mosquitoes mid-air with a laser-beam. The barrier consists of four cameras positioned on top and bottom of two opposing uni-coloured (white) posts spanning a plane and a beam aiming system. This system can focus targets within a volume represented as a determined offset from the plane and is realised by galvanometers. To drastically simplify the aiming process, galvanometer and camera are aligned to share a common optical centre. Thus, the distance of an object is irrelevant (as long as in the region of interest that the system is designed for) and all parameters to aim can be calculated from the position of the object in an image.

Concerning the transfer of this approach to our research topic, there are two major concerns:

The vestigial detection procedure allows low image quality and few computational stepscompared to advanced methods which are presented in previous chapters. Also, it is likely that the laser exposure time to cause damage to hornets is significantly higher than for mosquitoes even when using more powerful lasers (as presented earlier in this section). This leads to a larger region of interest amplifying the first concern.

Comparing the presented literature with the use case of hornet neutralisation, major challenges present themselves, which have not yet been answered by the state of the art. This is because of the nature of hornets that combine many of the challenging properties of aphids, cockroaches and mosquitoes: They have at least partly the scale of cockroaches which makes them harder to neutralise than mosquitoes, but just as the latter the move freely in 3D.
Furthermore, in bees they have a similar looking species around them that needs to be protected from getting killed by detecting or aiming falsely. Considering that in most cases the cost of the deployed technology significantly exceeds this project's resources, it is found questionable if an approach that involves neutralising hornets via laser-beam presents a feasible solution.

2.5.2. Dark tunnel ending in a trap

The approach described in this section is based on an active patent [24] which means that its use might be limited by property rights. The patent presents an apparatus to trap living hornets in a container which other insects that are smaller than hornets (which especially applies to bees) can escape. The design works in three consequent phases: First the hornet is detected, then a spring powered mechanism is triggered which traps it in a dark tunnel with a hole at the end. The hornets naturally navigate towards the light which leads them into the final container. This container has escape vents for smaller insects which might be accidentally caught instead or together with the hornet. While the patent does not further specify the detection process the project's web presence [27] claims to use AI while no further specifications are given. The provided images show that the camera that was used points towards a uniformly green surface so that its images will ultimately have a uniform and green background.

It is however to mention that the existence of a patent does not allow statements about this system's efficiency. As there are no further studies published it remains uncertain whether the technical solution actually provides an efficient hive protection or if there might even be casualties on the bee population caused by the system (e.g., by squeezing surrounding bees with the spring mechanism).

2.6. Literature review conclusion

Recent studies have demonstrated the promising potential of artificial intelligence (AI) in detecting and identifying Vespa Velutina and Apis Mellifera in front of beehives, mainly through image or video analysis by object or trajectory tracking. However, the implementation of such algorithms on durable embedded software can be challenging due to their heavy computational requirements. Optimization techniques is necessary to achieve low power consumption systems.

Embedded AI offers a viable solution for beehive protection, enabling efficient and effective hornet detection while minimizing hardware complexity and cost. AI algorithms can be implemented on various embedded platforms such as CPUs, GPUs, FPGAs with the flexibility to balance performance, power consumption, and hardware resources. For example, video streams inputs can be processed using a CPU or GPU, while the neural network computation can be accelerated using an FPGA. This approach maximizes the performance and efficiency of the system, providing sustainable solutions for beehive protection.

One of the main challenges of embedded AI is building a strong dataset to train the neural network. Our approach will involve a mix of existing data and on-field acquisition, labeled with the triple-blind method to ensure the reliability and quality of the data.

To overcome the challenges of video recording and analysis, alternative solutions have been explored, such as studying the behaviors of hornets and bees. For instance, research has shown that the bee buzz has a higher frequency when danger is near, and the wingbeat frequency of hornets and bees differ.

Furthermore, laser pulses and traps have been proposed as efficient methods for hornet removal, although these methods may require external power supply and careful consideration of the potential environmental impact. On the other hand, a simple swatter could be a practical and cost-effective solution for almost stationary targets, and it merits further investigation.

3. Initial research objectives

The primary objective of this project is to develop a device that exploits the stationary flight behavior exhibited by Asian hornets in front of the hive's landing board. The proposed approach involves conducting frequency analysis of hive vibrations and utilizing a camera in conjunction with artificial intelligence and video processing techniques for hornet detection. Subsequently, an automated hornet trap mechanism will be triggered to eliminate the identified hornets.

The specific objective is to develop robust and accurate algorithms for effectively discriminating between different species in order to have the least negative impact on biodiversity, ensuring ecological sustainability. In addition, the device will have to be energy self-sufficient since it be deployed in front of the hives and will not have permanent access to a power source. Considerations will also be given to guarantee the device's affordability, reliability, and practicality to facilitate its widespread adoption among beekeepers.

Overall, this project aims to address the pressing need for an efficient, ecologically sustainable, and economically viable solution to protect honeybees from the predatory impact of Asian hornets.

4. Methods

4.1. Danger detection via frequency analysis of the hive

Firstly, we conducted a search for audio recordings of bees to identify frequency peaks in the sounds emitted during bee activity. To accomplish this, we utilized the publicly available dataset "To bee or not to bee" from the Kaggle platform [28]. This dataset comprises audio recordings captured in the vicinity of beehives.

Each audio recording has a duration of several minutes and is accompanied by a corresponding text file *.txt* indicating the instances when one or more bees were in proximity to the microphone.

Our objective is to repurpose this dataset to identify the frequencies and sounds associated with bees under normal conditions (frequency peak around 200 Hz). Subsequently, this would enable us to detect hive excitement, whereby the frequency peak shifts towards higher frequencies around 6 kHz.

We had to segment and label these audio recordings in order to create subsets of ”hive with bees nearby” and ”hive without bees nearby” recordings. After deliberation, we decided to create 3-second excerpts. We have made this database openly accessible for potential future use.

The selection of a 3-second duration was arbitrary, ensuring that the audio recordings contained sufficient relevant information while being suitable for efficient processing by an embedded system.

The subsequent step involved standardizing the format of each audio file. We employed the Python library ”librosa” to resample the audio to a frequency of 16 kHz. This frequency was chosen to surpass the Nyquist Frequency, as the highest frequency emitted by the hive range from 6 kHz to 6.5 kHz, which is less than half of 16 kHz.

Additionally, we ensured that the microphone in our embedded solution was capable of recording at 16 kHz, enabling us to apply the same processing to live recordings within our application. Following the resampling process, we implemented an antialiasing filter using the ”scipy.signal” library. This involved applying a 4th order Butterworth filter with a cutoff frequency of 6500 Hz to eliminate unwanted frequencies and sampling noise from the extracts. Due to the computational resource and time constraints associated with performing calculations involving imaginary numbers on an embedded solution, we augmented the original extracts by reversing them and adding them to the originals, thereby making the resulting signal even. Since every Fourier coefficient of an even function is real, we could later discard the imaginary part of the Fourier transform. To mitigate artifacts resulting from abrupt truncation of the extracts, we applied a volume fade-in and fade-out effect. Specifically, we gradually transitioned the volume from 0% to 100% over a span of 0.1 seconds at the beginning and end of the resulting 6-second extracts.The final step involved normalizing the array to minimize the impact of relative volume differences between extracts. It is important to note that this normalization step is optional if we assume that the noise emitted by the bees maintains a consistent volume regardless of the environment.

Upon completing the time-based processing, we applied a Fast Fourier Transform (FFT) using the NumPy library. The constant value at 0 Hz was disregarded, and the transformed array was modified to include only positive real numbers.

Following the previous processing steps, we proceeded to apply a peak detection function from “scipy.signal” to each processed extract. The peak detection function was specifically implemented within a frequency window ranging from 80 Hz to 120Hz. This frequency range was chosen based on our literature review, which indicated that the flig ht of bees typically emits frequencies within this range. However, the software allows for adjusting the minimum and maximum frequency parameters, enabling flexibility to detect other frequencies, such as the 200 Hz frequency associated with hornet flight, if needed, as it has similar characteristics from an audio perspective.

To determine the optimal parameters for the peak detection function, we conducted an iterative process. We systematically varied the peak height, peak width, and distance from other peaks within the function. For each set of parameters, we evaluated the performance by constructing a confusion matrix. This evaluation involved testing the function on a subset of the dataset, comprising 500 to 1500 extracts. In order to assess the effectiveness of each parameter set, we calculated a scoring metric. The scoring metric was derived from the harmonic mean of the number of bee-containing extracts correctly identified by the function, divided by the total number of bee-containing extracts in the sample, and the number of non-bee-containing extracts correctly identified as such, divided by the total number of non-bee-containing extracts in the sample. Based on the results obtained from the scoring method, we selected the parameter set with the highest score, encompassing the optimal peak width, peak height, and distance values. To further validate the chosen parameters, we performed the scoring method on the entire dataset, confirming the effectiveness of the selected width, height, and distance parameters.

4.2. Training an artificial intelligence by supervised learning to detect Asian hornets

In order to recognize Asian hornets on a picture using supervised learning, we first had to create a labeled dataset of hornets and bees pictures to train our model. We didn't find already existing dataset corresponding to what we needed, so we decided to create our own database.

At the beginning, we took pictures on Kaggle [28], and found videos on the internet from which we extracted images, but we were not satisfied of our set. Those images were either too good to be representative of the ones we could take with our camera, or too bad to see anything useful for the training. Furthermore, in those datasets, there were too few Asian hornet pictures to train an efficient IA model. Finally, we retrieved images from other data sets used on the IA training website Roboflow [29], that we relabeled as we needed.

To increase the size of the dataset and to make the training more efficient, we resized images to 640x640 pixel and apply one percent of blur and noise. We obtained pictures more relevant to an outside environment. Next we trained our model with an algorithm that already exists and that shows good results : yolov5s [30]. We trained the model with a batch size of 32 images and did 50 iterations. During the training, we used ClearML [31] for monitoring and result analysis.

Next, we uploaded the model weights on a Jetson Nano board and used the Pytorch algorithm and an USB camera with jetson nano specific library jetcam [32] to infer the presence or not of an hornet.

4.3. Embedded package for the hornet detection camera

Once we trained our model and found good results in finding Asian hornets on pictures, we had to create a package in which we could put the Jetson Nano and it's battery, and the camera to film hornets in front if the hive. We used the 3D design website Tinkercad [33] to create the package.

4.4. Eliminating the threat with a swatter

Our literature review (see chapter 2.5) showed that the most promising approaches to eliminating hornets in front of bee hives are either too technologically advanced or power consuming to be implemented in an embedded system within the means of this project or protected by intellectual property right. That is why we decided to follow a straight-forward approach for the physical device: A fully automated swatter that can be triggered immediately after detection to hit onto a region of interest. As we so far have only designed a prototype we do not know the real effectiveness of the swatter. It could be coupled with different solutions to be sure to eliminate the hornet, such as pushing it into a container of soapy water or crushing and cutting it on a metal grid.

During the design of the prototype three main principles were considered:

- Simplicity and robustness
- Availability of components
- Separation between trigger and loading mechanism

Therefore we mainly chose components that were already available at the institute and are generally easily accessible in the field of embedded engineering. Furthermore all gearing components and mountings were manufactured by 3D printing and do not place high demands on accuracy making them easy to reproduce. We designed all gear wheels and mountings in Solidworks [34] and printed them with an Ultimaker3 3D printer (preparation in Cura). As an axis a hollow 10x2 mm aluminium axis was used that can be found in typical hardware stores. The separation between loading and triggering aims at assuring a minimum response time while keeping the demand at single components low. We assured this by slowly stressing an elastic during the loading phase using a slow motor, locking the position with a solenoid and then triggering within less than 0.5 s thanks to the short response time of the solenoid. The prototype includes six main components that are illustrated in figure 2: Swatter, gearing, elastic, electric motor (12 V DC), solenoid (12 V), electronic periphery (Jetson, relay, cables). The operating principle comprises four phases beginning from a neutral position (elastic not stressed, swatter lying on region of interest):

1. Begin of loading cycle: The motor raises the swatter with 100 % power for 30 s. At this point the elastic is beginning to stress.
2. The motor continues to raise the swatter with 50 % power for 43 s as the tension in the elastic increases. At the end of this phase the solenoid locks the swatter position.
3. The direction of the motor is inversed and it returns to its initial position.
4. In the final phase the swatter is ready to get triggered. If a hornet is detected in the region of interest the solenoid retracts the elastic immediately applies force to the swatter.

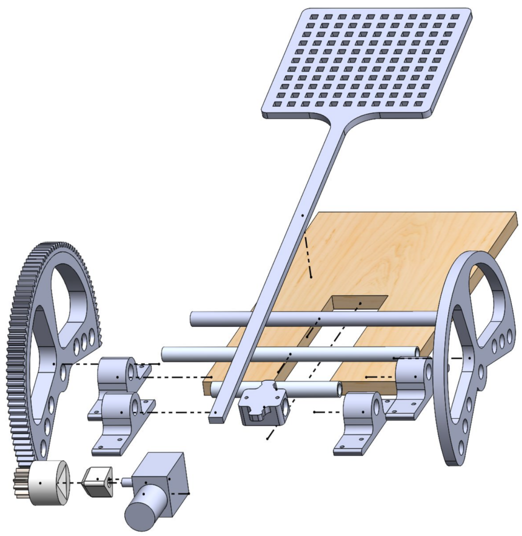

Figure 2: Explosion view of the CAD model of the prototype

5. Results

5.1. Frequency analysis

A comprehensive evaluation was conducted by running tests on 2700 sets, each consisting of randomly selected extracts from the database. Throughout the testing process, the obtained scores varied, with the minimum score recorded as 0.029 and the maximum score as 0.529. Remarkably, the highest score was achieved using specific parameter values: a distance of 2, a width of 3, and a height of 0.37. It is important to note that the width and distance parameters are measured in terms of points on the Fast Fourier Transform (FFT) array, while the height parameter represents the amplitude of the coefficient at the peak's highest point.

When applying these optimized parameters to the entire database, a total of 4077 extracts were successfully identified, while 3829 incorrect identifications were made.

5.2. Performance of the Artificial intelligence

Thanks to different source of images as mention in part 4.2, we finally obtained a database containing 1566 usable images, available on *https://universe.roboflow.com/pir/hornetv2* [35]. After adding blur and noise, we obtain 2200 train images, 310 for validation and 175 for testing.

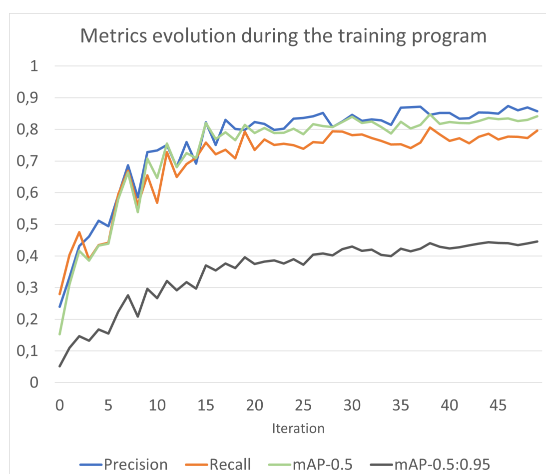

Figure 3: Metrics evolution during the training program

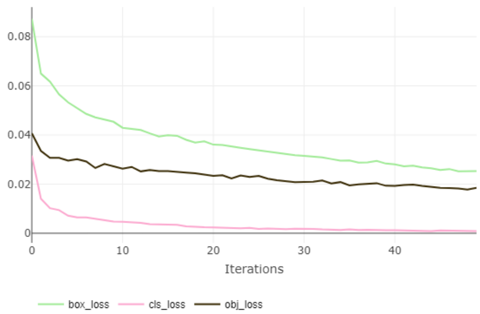

Figure 4: Loss evolution of the train set during the training program

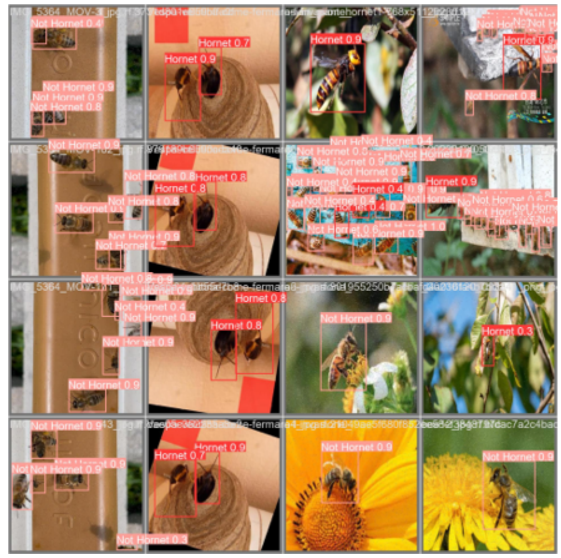

Figure 6: Detection by our model on a validation batch

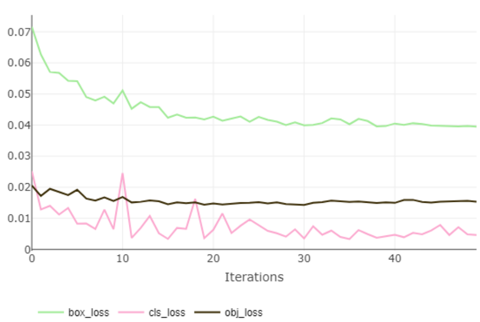

Figure 5: Loss evolution of the validation set during the training program

As we can see in Figure 3, we achieve a 80% recall and 85% precision model. The class and object loss curves of train and validation data that have almost the same values at the 50^{th} iteration, the model is not overfitting to the train data (Fig. 4-5).

The detection algorithm implemented on the Jetson Nano board that use PyTorch [36] *model()* script take around one second per iteration. As the hornet is mostly stationary ahead the beehive, that should not be a problem. We even added a counter to trigger the swatter only if there is an hornet three times in a row for more safety.

5.3. *3D package design*

Figure 7 represents our control system package. It is composed of three compartments :

- The back one for the Jetson Nano
- Left one for Jeton battery
- Right one for the camera with an opening to film the front of the beehive

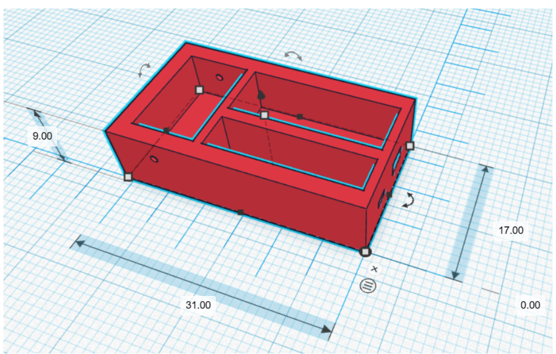

Figure 7: 3D package model for recognition system

5.4. *Assembly of the swatting device*

We assembled and tested the prototype for general functionality (see figure 8). However, we conducted all our tests so far under laboratory conditions using external power supply and without hitting real insects. While sticking to the planned design, we implemented minor improvements during the assembly process. Mainly fixation issues longitudinal of the axis were tackled. Also we reinforced the swatters bar to reduce deflection when under stress caused by the elastic. With those improvements the main functionalities (the four phases described in chapter 4.4) could be repeatedly and reliably proven.

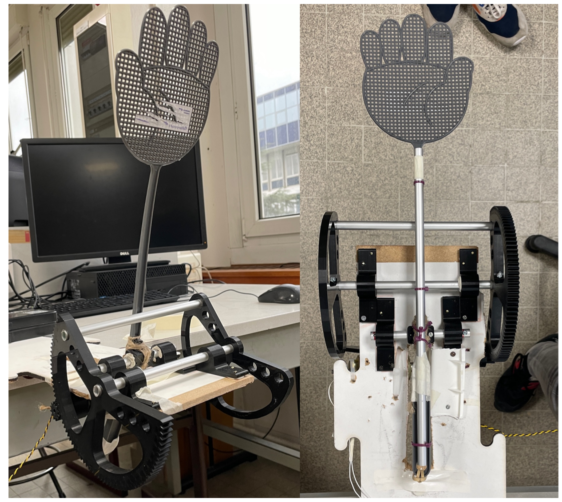

Figure 8: Assembly of the swatting device as tested under laboratory conditions.

6. Discussion

Regarding the main objective of this paper, we achieved to create a system able to detect and whip Asian hornet, but our system still need a lot of improvement.

First, the frequency analysis : Throughout the testing process, the obtained scores vary between 0.029 and 0.529. It means with the best parameters, we would only have a 53% chance of detecting a 'danger signal' coming from the hive. It is indeed too low to base the activation of the camera on that signal detection. However, this feature would allow to turn on only if a danger is detected, and so reduce significantly the energy consumed. It would the camera be interesting to find parameters we did not try, to improve the detection score, or figure out an other way to detect a danger for the hive.

Second, the hornet detection on pictures : The supervised model trained on our dataset achieved a 85% precision on hornet recognition and only 2% chance of mistaking a bee for a hornet. Also, the model does not present any sign of over-fitting. By adding a counter of up to 3 to check the presence of a hornet on 3 successive images, we ensure a conclusive result on hornet recognition. Due to the season, it was not possible to try the model in situ, but we simulated the presence of a hornet with a picture on a screen, and the recognition was accurate.

Third, the swatter : The automatic swatter is able to whip when the detection of an Asian hornet is confirmed by the Jetson camera. The whole system has been tested and found reliable under laboratory conditions. However, we do not know its impact on hornet, due to the lack of real tests. It would be necessary to try to know its efficiency, and adapt the mechanism if not powerful enough. In the same context the danger of hitting bees at the same time as a hornet is hard to estimate without field tests. It should be mentioned that generally a low number of bee casualties per eliminated hornet would be acceptable as the hornets could cause more damage over an extended period of time. Besides from uncertainties that derive from the lack of field experiments, there are some design improvements that we can already deduct from the laboratory tests: Firstly we would suggest decreasing the reload time to allow a higher rate of strikes. Also using contact sensors to know when the swatter is in a fully recharged position would increase the robustness of the system especially long term when performance drifts e.g. of the motor might occur. Finally, ensuring that the swatter is able to reach the hornet position thanks to his $[x,y]$ position is an interesting feature. It would prevent unnecessary striking, and therefore energy waste. This could either be reached by fine tuning the camera frame and the swatter position or by an extra image processing step. Although the current prototype is not yet fully embedded the low number of sensitive parts and the lack of need for high accuracy considering the used parts leads us to the conclusion that the necessary steps for achieving a fully embedded system will not pose any major technological obstacles.

Fourth, price of the system : As INSA students, we used the different components available in our school department, such as the NVIDIA Jetson Nano. It is viable from a technical point of view, but its price could be an issue on such system. An alternative option could be using a more affordable nano-computer such as Raspberry Pi to implement our IA, if it has enough processing power to handle the same amount of data. It would also implies programming modifications of the board.

Last, energy consumption and autonomy : In addition to creating a functional protection system, the aim was to design an embedded one, able to operate without being plugged to a continuous power source. To achieve it, we had to explore two different ways : on the one hand designing a low-consumption system. And on the other one using an autonomous power source, like a battery, that could be paired with an energy recuperation device as solar panel. We were not able to explore those ideas due to a lack of time, so the actual prototype need to be plugged. Solar panel could be added on top of the camera package, and recharge the battery during the day. We could also study the time from which hornets no longer attack hives in the day, and use a brightness sensor to detect that time and turn off the system for the night. Same with the seasons, as we know hornets have a long period of hibernation during winter.

Finally, the interpretable python code could be transformed as a compiled language to speed up processing and reduce consumption. However, it would represent a lot of work, because all the yolov5 detection algorithm is coded with python.

7. Project management

As we were seven students working on this project we needed to divide the project into 3 parts. We tried to work with an Agile method with review meetings with our tutors and the whole team every two weeks. In practice, it was difficult to find slots where we were all available to work in small teams as we study in different classes. So we had communication difficulties and this caused a delay in the project. In addition, at the end of the year, there are a lot of project submissions due, which further reduces the time we had to spend on each project. However, we managed to finish the assignment we were given in the time allotted. There are still improvements to be made (see discussion section).

For most of us, this was the first long-term project with so many participants. We discovered the difficulties of organisation, division of tasks and communication within the group. Also, we enjoyed discovering research work during our course, which allowed us to confirm or affirm our appetite for research. We learnt to evolve in quasi-autonomy, in particular by selecting technical solutions, by solving problems encountered and by managing a situation of a research axis that does not lead to a solution. Finally, we started from a need formulated by our teachers to arrive at a functional prototype in a relatively short time.

8. Conclusion

Our aim was to develop a mechanism for protecting beehives against Asian hornets based on image processing by supervised learning. Analysing the frequency of the hive should allow the detection of a hazard, but does not lead to concluding results concerning any threat of the bees. Based on the hornet and bees dataset we used to train our supervised learning model, the camera-computer group is able to detect with satisfying accuracy the presence of a hornet. Moreover, the automatic swatter is able to whip when the detection of an Asian hornet is confirmed by the Jetson camera, and reload itself after the strike. The system is not fully embedded as it as to be plugged on a power source, but it fulfills its main objective, as it detect a hornet and hit it to protect the bees.

Acknowledgements

We want to thank our tutors, G.Lecorre and E.Chanthery for proposing this subject, and following us through all the process of this paper.
We also want to thank bib'INSA tutors for the research and references formation sessions provided during the first semester, and G.White-Peeters for the scientific english writing course.
In addition, our thanks go to all those who helped us, or who were able to guide us during this process : T.Rocacher / P.Acco / J.Martin / S. Di Mercurio / E. Lombard. Finally, we are grateful to the GEI department for providing us resources and materials.

References

[1] Sandra V. Rojas-Nossa et al. "Performance of baited traps used as control tools for the invasive hornet Vespa velutina and their impact on non-target insects". In: *Apidologie* 49.6 (Dec. 1, 2018). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 6 Publisher: Springer Paris, pp. 872–885. ISSN: 1297-9678. DOI: 10.1007/s13592-018-0612-0. URL: https://link.springer.com/article/10.1007/s13592-018-0612-0 (visited on 02/01/2023).

[2] Frédéric Larguier. "Piège à insecte automatique pour capturer vivants et de manière sélective des insectes". French pat. 3103353. Nov. 22, 2019. URL: https://patents.google.com/patent/FR3103353B1/fr?inventor=Fr%5C%C3%5C%A9d%5C%C3%5C%A9ric+Larguier&patents=false&oq=Fr%5C%C3%5C%A9d%5C%C3%5C%A9ric+Larguier.

[3] Rama Chellappa, Ashok Veeraraghavan, and Gaurav Aggarwal. "Pattern Recognition in Video". In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 11–20. ISBN: 978-3-540-32420-1. DOI: 10.1007/11590316_2. URL: https://link.springer.com/chapter/10.1007/11590316_2.

[4] Hugo Hadjur, Doreid Ammar, and Laurent Lefèvre. "Toward an intelligent and efficient beehive: A survey of precision beekeeping systems and services". In: *Computers and Electronics in Agriculture* 192 (2022), pp. 1–16. DOI: 10.1016/j.compag.2021.106604. URL: https://hal.inria.fr/hal-03483914 (visited on 01/29/2023).

[5] Guillaume Chiron. "Full process of acquisition, multi-target tracking, behavioral modeling for naturally crowded environments : application to beehives monitoring". PhD thesis. La Rochelle: Université de La Rochelle, Nov. 28, 2014. 191 pp. URL: https://theses.hal.science/tel-01264108 (visited on 01/24/2023).

[6] Abdul Nasir, Muhammad Obaid Ullah, and Muhammad Haroon Yousaf. "AI in apiculture: A novel framework for recognition of invasive insects under unconstrained flying conditions for smart beehives". In: *Engineering Applications of Artificial Intelligence* 119 (Mar. 1, 2023), p. 105784. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2022.105784. URL: https://www.sciencedirect.com/science/article/pii/S0952197622007746 (visited on 01/28/2023).

[7] Abhinav Goel et al. "A Survey of Methods for Low-Power Deep Learning and Computer Vision". In: *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). New Or-

leans, LA, USA: IEEE, June 2020, pp. 1–6. ISBN: 978-1-72815-503-6. DOI: 10.1109/WF-IoT48130.2020.9221198. URL: https://ieeexplore-ieee-org.gorgone.univ-toulouse.fr/document/9221198 (visited on 02/04/2023).

[8] W.S.K. Fernando et al. "Video event classification and anomaly identification using spectral clustering". In: *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer). Colombo, Sri Lanka: IEEE, Aug. 2015, pp. 21–24. DOI: 10.1109/ICTER.2015.7377661. URL: https://ieeexplore-ieee-org.gorgone.univ-toulouse.fr/document/7377661.

[9] explorium_admin. *ML Clustering: When To Use Cluster Analysis, When To Avoid It*. Explorium. Feb. 3, 2020. URL: https://www.explorium.ai/blog/clustering-when-you-should-use-it-and-avoid-it/ (visited on 02/12/2023).

[10] MyeongAh Cho et al. "Unsupervised video anomaly detection via normalizing flows with implicit latent features". In: *Pattern Recognition* 129 (Sept. 1, 2022), p. 108703. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2022.108703. URL: https://www.sciencedirect.com/science/article/pii/S0031320322001844 (visited on 01/30/2023).

[11] George K. Thiruvathukal and Yung-Hsiang Lu. "Efficient Computer Vision for Embedded Systems". In: *Computer* 55.4 (Apr. 11, 2022). Conference Name: Computer, pp. 15–19. ISSN: 1558-0814. DOI: 10.1109/MC.2022.3145677. URL: https://ieeexplore-ieee-org.gorgone.univ-toulouse.fr/document/9755195 (visited on 02/04/2023).

[12] Sergei Alyamkin et al. "Low-Power Computer Vision: Status, Challenges, and Opportunities". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.2 (June 2019). Conference Name: IEEE Journal on Emerging and Selected Topics in Circuits and Systems, pp. 411–421. ISSN: 2156-3365. DOI: 10.1109/JETCAS.2019.2911899. URL: https://ieeexplore-ieee-org.gorgone.univ-toulouse.fr/document/8693826.

[13] Odrika Iqbal et al. "Design and FPGA Implementation of an Adaptive video Subsampling Algorithm for Energy-Efficient Single Object Tracking". In: *2020 IEEE International Conference on Image Processing (ICIP)*. 2020 IEEE International Conference on Image Processing (ICIP). ISSN: 2381-8549. Abu Dhabi, United Arab Emirates: IEEE, Oct. 2020, pp. 3065–3069. ISBN: 978-1-72816-395-6. DOI: 10.1109/ICIP40778.2020.9191146. URL: https://ieeexplore-ieee-org.gorgone.univ-toulouse.

fr / document / 9191146 (visited on 02/04/2023).

[14] Ivan A. Bogoslovskii et al. "Implementation of an Approximator Based on a Multilayer Perceptron and Wavelet-Neural Network on the STM32 Microcontroller". In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). ISSN: 2376-6565. Jan. 2020, pp. 1372–1377. DOI: 10.1109/EIConRus49466.2020.9039482. URL: https://ieeexplore-ieee-org.gorgone.univ-toulouse.fr/stamp/stamp.jsp?tp=&arnumber=9039482.

[15] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. 2e édition. Birmingham Mumbai: Packt Publishing, Sept. 20, 2017. 622 pp. ISBN: 978-1-78712-593-3.

[16] *Flickr*. URL: https://www.flickr.com/.

[17] *Wikimedia*. URL: https://www.wikimedia.org/.

[18] Jiaxu Li. "Research on false eyewitness Detection algorithm of Asian giant hornet Image based on support Vector Machine". In: *Journal of Image Processing Theory and Applications* 4 (2021), pp. 33–37. ISSN: 2560-6239. DOI: 10.23977/jipta.2021.41005. URL: https://www.clausiuspress.com/article/1589.html.

[19] Ping Wen et al. "The sex pheromone of a globally invasive honey bee predator, the Asian eusocial hornet, Vespa velutina". In: *Scientific Reports* 7 (Oct. 11, 2017). Number: 1 Publisher: Nature Publishing Group, p. 12956. ISSN: 2045-2322. DOI: 10.1038/s41598-017-13509-7. URL: https://www.nature.com/articles/s41598-017-13509-7 (visited on 02/03/2023).

[20] Erik Schmolz et al. "Direct calorimetric measurement of heat production rates in flying hornets (Vespa crabro; Hymenoptera)". In: *Thermochimica Acta* 328.1 (Mar. 22, 1999), pp. 3–8. ISSN: 0040-6031. DOI: 10.1016/S0040-6031(98)00617-0. URL: https://www.sciencedirect.com/science/article/pii/S0040603198006170 (visited on 02/02/2023).

[21] Cayetano Herrera et al. "Automated detection of the yellow-legged hornet (Vespa velutina) using an optical sensor with machine learning". In: *Pest Management Science* 79.3 (2023), pp. 1225–1233. ISSN: 1526-4998. DOI: 10.1002/ps.7296. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/ps.7296 (visited on 02/01/2023).

[22] Mohammad Ali Ansari, Mohsen Erfanzadeh, and Ezeddin Mohajerani. "Mechanisms of Laser-Tissue Interaction: II. Tissue Thermal Properties". In: *Journal of*

Lasers in Medical Sciences 4.3 (2013), pp. 99–106. ISSN: 2008-9783.

[23] Markolf H. Niemz. *Laser-tissue interactions: fundamentals and applications*. Fourth Edition. Cham: Springer, 2019. 311 pp. ISBN: 978-3-030-11917-1 978-3-030-11916-4.

[24] Virginie Lacotte et al. "Pesticide-Free Robotic Control of Aphids as Crop Pests". In: *AgriEngineering* 4.4 (Oct. 7, 2022), pp. 903–921. ISSN: 2624-7402. DOI: 10.3390/agriengineering4040058. URL: https://www.mdpi.com/2624-7402/4/4/58 (visited on 02/02/2023).

[25] Ildar Rakhmatulin, Mathieu Lihoreau, and Jose Pueyo. "Selective neutralisation and deterring of cockroaches with laser automated by machine vision". In: *Oriental Insects* (Sept. 21, 2022), pp. 1–18. ISSN: 0030-5316, 2157-8745. DOI: 10.1080/00305316.2022.2121777. URL: https://www.tandfonline.com/doi/full/10.1080/00305316.2022.2121777 (visited on 02/02/2023).

[26] Jordin Kare and Jude Buffum. "Build your own photonic fence to zap mosquitoes midflight [Backwards Star Wars". In: *IEEE Spectrum* 47.5 (May 2010), pp. 28–33, 52, 53. ISSN: 0018-9235. DOI: 10.1109/MSPEC.2010.5453138. URL: http://ieeexplore.ieee.org/document/5453138/ (visited on 02/10/2023).

[27] Association Apipolline. *VIGIVELUTINA le piège contre le frelon asiatique tueur d'abeilles*. https://jeparticipe.laregioncitoyenne.fr/. URL: https://jeparticipe.laregioncitoyenne.fr/project/les-trophees-pour-la-biodiversite-en-occitanie-2/collect/depot/proposals/vigivelutina-le-piege-contre-le-frelon-asiatique-tueur-dabeilles-modifier?fbclid=IwAR0rU7H4OOsoCckWcknDwgi_6AS3KbuYMeZRxwtcLU5cfXLG5JyE6IPwovc (visited on 02/11/2023).

[28] *Kaggle*. URL: https://www.kaggle.com/.

[29] *Roboflow*. URL: https://roboflow.com/.

[30] *yolov5 open source*. URL: https://github.com/ultralytics/yolov5.

[31] *ClearML*. URL: https://clear.ml/.

[32] *jetson nano librarie*. URL: https://github.com/NVIDIA-AI-IOT/jetcam.

[33] *Tinkercad*. URL: Tinkercad.%20https://www.tinkercad.com.

[34] *Solidworks*. URL: https://www.flickr.com/https://www.solidworks.com/fr.

[35] *Hive protector dataset*. URL: https://universe.roboflow.com/pir/hornetv2.

[36] *PyTorch library*. URL: https://pytorch.org/.