A light trap and computer vision system to detect and classify live moths (Lepidoptera) using tracking and deep learning

Kim Bjerge[a,*], Jakob Bonde Nielsen[a], Martin Videbæk Sepstrup[a], Flemming Helsing-Nielsen[b], Toke Thomas Høye[c]

[a] *School of Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark*
[b] *NaturConsult, Skrænten 5, 9520 Skørping, Denmark*
[c] *Department of Bioscience and Arctic Research Centre, Aarhus University, Grenåvej 14, 8240 Rønde, Denmark*

Abstract

This paper presents a portable computer vision system, that is able to attract and detect live insects. More specifically, the paper proposes detection and classification of species by recording images of live moths captured by a light trap. A light trap with multiple illuminations and a camera was designed to attract and monitor live insects during twilight and night hours. A computer vision algorithm referred to as *Moth Classification and Counting*, based on deep learning analysis of the captured images then tracked and counted the number of insects and identified moth species. This paper presents the design and the algorithm that were used to determine and identify the moth species. Observations over 48 nights resulted in the capture of more than 250,000 images with an average of 5,675 images per night. A customized convolutional neural network was trained on 864 labelled images of live moths, which were divided in to eight different species, achieving a high validation F_1-score of 0.96. The algorithm measured an average classification and tracking F_1-score of 0.83 and a tracking detection rate of 0.79. This result was based on an estimate of 83 individual moths observed during one night with insect activity in 122 minutes collecting 6,000 images. Overall, the proposed computer vision system and algorithm showed promising results in nondestructive and automatic monitoring of moths as well as classification of species. The system provides a cost-effective alternative to traditional methods, which require time-consuming manual identification and typically provides coarse temporal solution to capturing data. In addition, the system avoids depleting moth populations in the monitoring process, which is a problem in traditional traps that kill individual moths. As image libraries grow and become more complete, the images captured by the trapping system can be processed automatically and allow users with limited experience to collect data on insect abundance, biomass, and diversity.

Keywords: Moth, Light Trap, Computer Vision, Tracking, Deep Learning, CNN

1. Introduction

There is growing evidence that insect populations are declining in abundance, diversity, and biomass Wagner (2020). Multiple anthropogenic stressors are proposed as drivers of these changes Harvey et al. (2020), but the complexities of the responses and differences in the ecology of the millions of species involved makes it difficult to pinpoint causes of these declines as well as mitigation measures. For many taxa, long-term population census data are also non-existing and more extensive monitoring. Thus, more especially of species rich taxa, is critically needed.

More than half of all described species on Earth are insects. With more than 160,000 described species, the insect order Lepidoptera, which consists of moths (95%) and butterflies (5%), is one of the largest. Moths are important as pollinators, herbivores and prey for e.g. birds and bats. Reports of declining moth populations have come from Great Britain Fox et al. (2013) Sweden Franzén and Johannesson (2007) and the Netherlands Groe-

[*] Corresponding author
Email address: kbe@ase.au.dk (Kim Bjerge)

nendijk and Ellis (2011). Recently however stable populations have also been observed Macgregor et al. (2019). Changes in the abundance of moths could have cascading effects through the food web suggesting that moths are a very relevant group of insects to monitor more effectively in the context of global change.

The most widely applied methods of monitoring moths are to use light traps Jonason et al. (2014). The moths are typically killed and manually counted and classified by human intervention. However, most methods requires experts and tedious manual labor to visit the traps and classify individual species. In this paper, we describe a new method for automatic counting and classification of moth species in order to accelerate the study of moth population ecology and minimize the impact on moth population.

1.1. Related work

Attempts to trap and record insects with computer vision or identify moths based on images have already been described in various papers.

A survey of moths using light traps has been done in Jonason et al. (2014), where two different light sources were used to attract different species of moths. The specimens were collected in a box without killing them. The box was then emptied every morning for manual species identification. The survey explored the influence of weather, time of year and light source. It was observed that a high levels of species richness and abundance per night were mainly influenced by a high temperature, low humidity and a lamp type with high power intensity. It was concluded that sampling during the warmest nights in each month (March to October) achieved the best results for understanding the species' phenology. Inspired by the work of Jonason et al. (2014) our work includes the use of light to attract insects and moths during twilight and night hours.

Ding and Taylor (2016) have presented a trap for automatic detection of moths, which contained a pheromone lure to attract insects of interest and an adhesive surface, where the insects became stuck. Digital images with a relatively low resolution (640x480 pixels) were taken of the dead and live insects at a fixed time point daily and transmitted to a remote server. The paper showed that it is possible to train a convolutional neural network (CNN) to identify moths. However, it only dealt with a binary classification problem to recognize if an insect in the landfill was a moth or another type of pest insect.

Watson et al. (2004) and Mayo and Watson (2007) have provided an analysis of 774 live individuals from 35 different moth species to determine whether if computer vision techniques could be used for automatic species identification. Focusing on data mining for feature extraction and a Support Vector Machine (SVM) for classification, their work achieved a classification accuracy of 85% among 35 classes distributed across 774 images.

While previous analyses of the same dataset would require manual selection of regions on the moths to identify an individual, Mayo and Watson (2007) could effectively classify from a single image. In addition, Batista et al. (2010) have proposed a nearest neighbor algorithm based on features such as texture, color and shape to classify moth species from the same dataset obtaining an accuracy of 79.53%. The work of Watson et al. (2004), Mayo and Watson (2007) and Batista et al. (2010) were published before CNNs became a widespread technology, and their work contributed to the future, as a foundation of solving the problem of species classification, with deep learning.

Several of the issues associated with moth classification when the dataset contains a large number of classes have been illustrated by Chang et al. (2017). Their work presented a dataset of 636 butterflies and moths distributed over 14,270 highly detailed images, which were collected using internet search engines. The challenge with such a large dataset is that the variation between species is minimal while the variation within the individual classes can be large. In addition, the dataset consisted of images with complex backgrounds which makes it difficult to distinguish the individual insects. This makes it necessary to use more complex and larger models such as Visual Geometry Group (VGG) Simonyan and Zisserman (2015), Inception Szegedy et al. (2015) and Residual Networks (ResNet) He et al. (2016) to perform classification reliably. Furthermore, training an effective model with is challenging with rare species, where there is not enough data. In our work, we presents a simpler CNN model, which is able to classify moth species based on images with a controlled illumination, background and camera setup.

Several others Chang et al. (2017); Xia et al. (2018) have proposed deep CNNs to make fine-grained insect, butterfly and moth classifications based on images from the Internet. The challenge of using this approach is that butterflies and moths are typically captured with unfolded wings in such images. This is not the natural resting position for live moths on a plane surface and thus not the way they would appear in a light trap such as the one presented in the present paper.

Zhong et al. (2018) have also worked on a trap that captures individual insects for later analysis. Their system was implemented on a Raspberry Pi 2 model B with the corresponding Raspberry camera module. An algorithm was proposed for the detection and classification of different orders of flying insects. To detect and count the number of individuals, a first version of the deep neural network ”You Only Look Once” (YOLO) Redmon et al. (2016) was used. A SVM was after wards used to perform classification of insect order based on features. Using a combination of YOLO and SVM minimized the amount of training data required. The work concentrated on the classification of insect orders and thus did not make the determination of different families or species. The result was a counting and classification precision of 92.5% and 90.8% respectively. Since the classification was made with SVM based on manually defined features, it is unlikely the method would be easy to expand for moth species classification as selecting useful features would be challenging. Since the trap captures and retains the individuals, counting the individuals would be a relatively trivial task compared to tracking live moving insects.

This paper presents a new light trap with a camera and several illuminations to attract and monitor flying and resting insects and moth species automatically without killing them. The computer vision system is based on a Raspberry Pi and a high resolution web camera, which is able to capture detailed images of the individual insects.

In relation to image processing, our paper presents a novelty in the form of an extended image-processing pipeline that also considers the temporal image dimension. Compared to the results in Watson et al. (2004); Xia et al. (2018) and Eliopoulos et al. (2018), we present a fully functional image-processing pipeline that includes tracking of insects and utilizes a CNN for moth spices classification in the intermediate steps of the pipeline. In the last part of the pipeline, we incorporate a novel image-processing algorithm named Moth Classification and Counting (MCC) which is able to track and count moth species in the temporal time dimension.

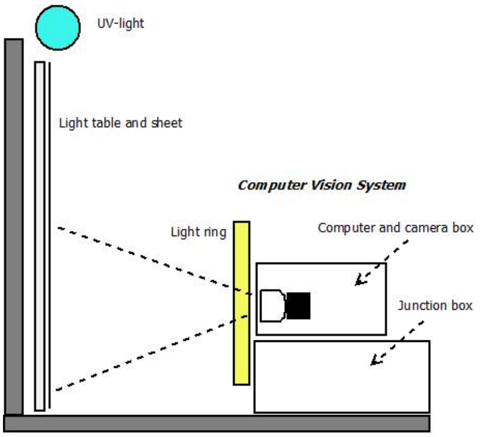


Figure 1: The portable light trap with a light table, a white sheet and UV light to attract live moths during night hours. The computer vision system consisted of a light ring, camera with computer and electronics and a powered junction box with DC-DC converter.

2. Materials and methods

In this section, the portable light trap and computer vision system is described, which was used to detect and monitor live insects and moths. The system was designed to attract moths and insects in the night hours and automatically captured images based on motion. Whenever a change in the camera field of view was detected by the computer vision system a sequence of images of the moth or insect was captured and stored on a hard drive. Any insect above a certain size would be detected by the camera and insect motion would be captured. The construction and selection of components of the system and algorithm to count and identify the species of moth is described in the following sections.

2.1. Hardware equipment

The primary components of the light trap vision system was the ultra-violet (UV-light) fluorescent tube from Bioform (accessed on 1/3-2020) (Article No.: A32b), light table from Computermester (accessed on 1/3-2020) and computer vision system as illustrated in figure 1. The purpose of the UV-light was to attract insects to the location of the trap from a long distance. A light table (LED A3 format) placed in front of the computer vision system was covered with a white sheet to ensure a diffuse illumination without reflection in the images. The white uniform background ensured easy detection of the insects and the light table helped to ensure

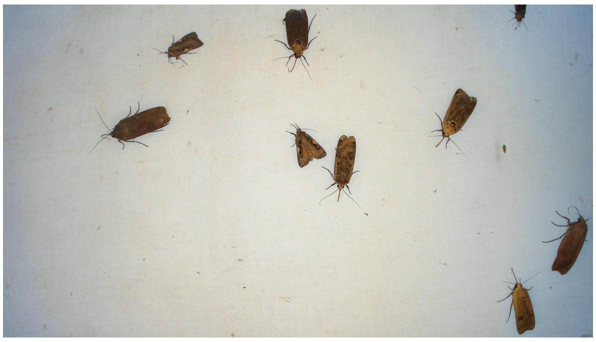


Figure 2: A picture of 3840x2160 pixels taken of light table and resting moths with a light ring illumination on a white uniform background.

that insects settled on the white sheet. The computer vision system was composed of a light ring from 24hshop (accessed on 1/3-2020), web camera from Logitec (accessed on 1/3-2020) (Brio Ultra HD Pro Webcam) with a resolution of 3840x2160 pixels and a Raspberry Pi 4 computer. The light ring (diameter 153/200 mm) was placed in front of the light table to ensure a diffuse foreground illumination of the insects. The intensity of the light ring could be adjusted from a combination of yellow and blue/white LED light. Camera focus and exposure was manually adjusted to achieve an optimal image quality as seen in figure 2. The field of view was adjusted in order to ensure sufficient resolution of the insects and cover approximately half of the area of the light table. A working distance of 200 mm from camera to light table was chosen. This gave a field of view of 320x170 mm and resulted in a sufficient image quality being able for identifying moth species. The camera and light were adjusted so that the moth species could be identified by an expert based on an enlarged photo of as shown in figure 3.

Sugar water was sprinkled on the white sheet to attract insects. The sugar water caused more insects to stay on the sheet for a longer period of time. However, the trap also worked without sugar water, although less effectively. A motion program running on the Raspberry Pi 4 was installed to capture a sequence of images whenever a movement was detected in the camera view. It was programmed to save images in JPG format whenever it detected change in more than 1500 pixels. This setting ensured that only larger insects where captured and thus filtered smaller insects such as mosquitos and flies from the footage. The maximum frame rate was limited to 0.5-2 fps, resulting in a timeframe

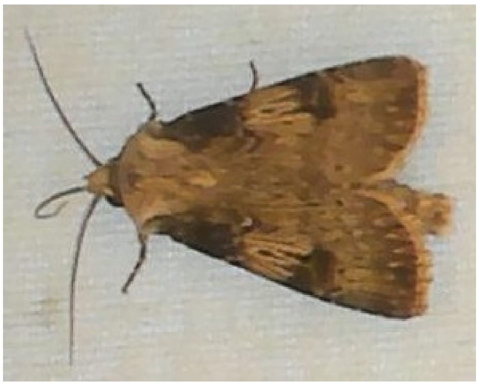


Figure 3: An enlarged picture taken of a moth identified as species *Agrostis puta* by an expert.

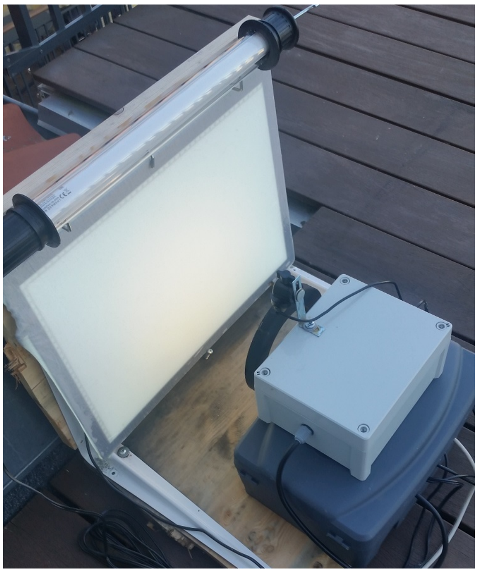


Figure 4: Photo of the light trap and computer vision system.

of 0.5-2 seconds between images. On warm summer nights with a high level of insect activity more than 20,000 images were captured per night. To save space on the hard drive, a frame rate of 0.5 fps was selected, which was sufficient for identifying and counting moth species. To save power, the Raspberry Pi was programmed to capture images between 9pm and 5am and to turn off the UV-light during daytime. A circuit with relay was constructed to control the UV-light placed inside the computer and camera box. A solid-state drive (500 GB) was connected to the Raspberry Pi to store the captured images. Inside the junction box, a DC-DC converter was installed to convert the supply voltage from 12V to 5V. The Raspberry Pi and light ring was supplied with 5V and UV-light, and the light table was supplied with 12V.

The whole system was supplied with 12V. The power consumption was 12.5 Watt during daytime when the UV-light was turned off and the computer system was idle. During the night, the system used 30 Watt when the UV-light was turned on and the computer vision system was recording images. With a power consumption of only 12.5 Watt during day hours and 30 Watt during night hours, a 12V battery with a 25Wp solar panel and regulator should be sufficient for powering the system during the entire summer period. Figure 4 shows a photograph of the actual constructed light trap and computer vision system seen from the outside.

2.2. Counting and classification of moths

We developed a novel computer vision algorithm refered to as *Moth Classification and Counting* (MCC), which was able to count the number of insects and identify known species of moths. The algorithm produced statistical data of individual moths and their species, as well as the number of unknown insects detected. For every individual moth detected, the time and identity was recorded. In the following subsections, important parts of the algorithm are explained in more detail. However, we will first present brief overview of the MCC algorithm. The MCC algorithm was composed by a number of sequential steps, where each image in the recording was analyzed as illustrated in figure 5.

The first step was to read an image for a selected observed night sorted by time of capture. This was segmented as black and white, followed by blob detection to mark a bounding box around each detected insect. This was achieved via the blob detection and segmentation described in section 2.2.1. The position of each insect region in the image was estimated based on the center of the bounding box.

The second step tracked insects in the image sequence as described in section 2.2.2. Tracking was important for recording the movement and behavior of the individual insect during its stay in the light trap and ensuring it only was counted once.

For every insect track, it was evaluated whether the insect was a known moth species. A trained customized CNN, which used a fixed cropped and resized image of each insect, was used to predict moth species. This third step is described in more detail in section 2.2.3.

The final step collected information and derived a summary of counted individuals of known moth species and unknown insects detected and tracked by the algorithm. The summary information were annotated to the image with highlighted marks of each insect tracks.

2.2.1. Blob detection and segmentation

A grey-scaled foreground image was made by subtracting a fixed background image of the white sheet without insects. Several methods were investigated to perform segmentation of a black and white image like using a global threshold or regions of adaptive threshold segmentation. The Otsu (1979) threshold algorithm turned out to be the best choice. A black and white image was made using Otsu threshold on the foreground image, followed by a morphological open and close operations to filter small noisy blobs and closing of blobs. Finally the contour of blobs was found and the bounding box of insect regions was estimated. Estimating the position of insects based on bounding boxes did not work properly in rare cases, when the individuals were too close to each other. A result of segmentation and bounding box estimation is shown in figure 6.

2.2.2. Insect tracking and counting

If the same insect was found in a sequence of images, it should only be counted once. Thus, it was necessary to track the insect in order to count the number of individual insects correctly. The individual insects were relatively stationary during their stay, and images were captured at two second intervals in case of activity. Therefore, it was assumed that a match was more likely for the shortest distance between two bounding boxes. That is, two boxes that were close to each other were likely to be the same individual.

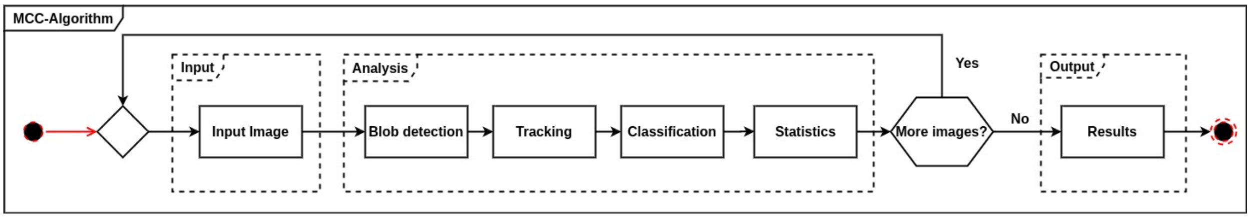


Figure 5: The processing pipeline of the intermediate- to high-level image processing algorithms to track and count the number of insects. A customized trained convolutional neural network (CNN) was used for the moth species classification.

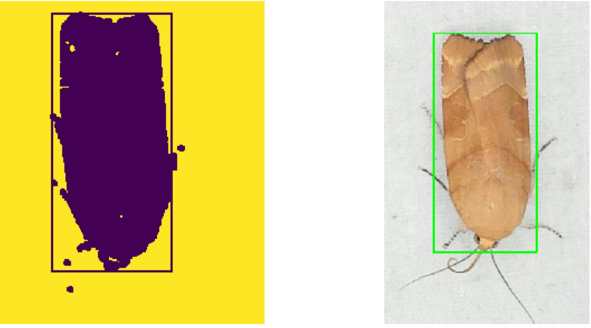


Figure 6: A blob segmented image of a detected moth (*Noctua fimbriata*) with bounding box.

The position and size of each insect were estimated for every single frame, and tracking could therefore be solved by finding the optimal assignment of insects in two consecutive images. The Hungarian Algorithm Munkres (1957) was our chosen method for finding the optimal assignment for a given cost matrix. In this application, the cost matrix should represent how likely it was that an insect in the previous image had moved to a given position in the current image. The cost function was defined as a weighted cost of distance and area of matching bounding boxes in the previous and current image. The Euclidean distance between center position (x, y) in the two images was calculated as follows:

$$Dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

This distance was normalized according to the diagonal of the image:

$$Max_{dist} = \sqrt{(I_{height})^2 + (I_{width})^2} \tag{2}$$

The area cost was defined as the cost between the size of bounding boxes:

$$Area_{cost} = \frac{Min_{area}}{Max_{area}} \tag{3}$$

A final cost function in equation 4 was defined with a weighted cost of distance W_{dist} and weighted cost of area W_{area}.

$$Cost = \frac{Dist}{Max_{dist}} W_{dist} + (1 - Area_{cost}) W_{area} \tag{4}$$

The Hungarian Algorithm required the cost matrix to be squared and, in our case, was defined as an $N \times N$ matrix, where each entry was the cost assigning $insect_i$ in previous image to $insect_j$ in current image. After a match with minimum cost, the entry in the current matrix was assigned a Track ID from the entry in the former. The found Track IDs and entries were stored and used in the upcoming iteration. Dummy rows and columns were added to the matrix to ensure that it was always squared. All entries in the dummy row or column had to have a cost significantly larger than all other entries in the cost matrix to ensure that the algorithm did not make a “wrong” assignment to a dummy. The resource or task being assigned to a dummy could be used to determine which insect from the previous image had left, or which insect had entered into the current image.

To evaluate the performance of the tracking algorithm, two metrics were defined based on the paper Bashir and Porikli (2006). The measure False Alarm Rate (FAR) was an expression of the probability that a given track was incorrect. It describes the number of false alarms relative to the total number of tracks; that is, how many times the tracker made a wrong track compared to the times it made a track.

$$FAR = \frac{FP}{TP + FP} \tag{5}$$

While a true positive (TP) was defined as an individual who retained its uniquely assigned Track ID in its entire presence of the observation. A false positive (FP) was defined as an individual who was either counted multiple times or assigned a new Track ID.

The term Tracking Detection Rate (TDR) was a measure of the number of individuals who maintained their own Track ID in relation to the established ground truth (GT), during the course of observation. The size was therefore used as the primary scale to express the tracker's ability to maintain the same Track ID for the individual insects in an observation.

$$TDR = \frac{TP}{TG} \tag{6}$$

GT was defined as the total number of unique individuals in the test set measured by manual counting.

2.2.3. Moth species classification

In the field of deep learning, specific architectures of CNNs have provided particularly positive results in many areas of computer vision Liu et al. (2017). CNNs use both pixel intensity values and spatial information about objects in the image. It was a challenging task to find a suitable CNN architecture for classification of moth species. Several CNN architectures Bjerge et al. (2019); Sandler et al. (2018), were investigated. A customized network was finally designed inspired by Krizhevsky et al. (2017). Hyperparameters of the architecture were explored to find the optimal network architecture to classify moth species. The model was designed to be light and fast for the purpose of being able to be executed on the embedded Raspberry Pi computer used in the light trap.

Since the camera recorded images of insects with a constant working distance, the insects did not change in size in the images. The moths were labeled with bounding boxes with an average size of 473×475×3 pixels and a standard deviation of 100 for pixel height and width. Initial experiments gave poor results with a resized input size of 224×224×3, which many CNNs Huang and Wang (2017) use. Improved results were achieved by reducing the input size, while still being able to visually identify the moth species. Based on the given camera setup the bounding boxes were finally resized approximately seven times to a fixed window size of 64×64×3 as input for the customized CNN model.

Several images with eight different species of moths were used for training the CNN model. The CNN model had four layers for feature detection and two fully connected layers for final species classification. The optimal architecture was found by using combinations of hyperparameters for the first and last layer in the CNN. Below are the parameters used to train different CNN's for species classification:

Fixed pool size and stride, $n \times n$, $n \in \{2\}$

Kernel size $n \times n$, $n \in \{3, 5\}$

Convolutional depth n, $n \in \{32, 128\}$

Fully connected size n, $n \in \{256, 512\}$

Optimizer n, $n \in \{Adam, SGD\}$

The optimal chosen CNN architecture is shown in figure 7. The first layer performed convolution using 128 kernels with a kernel size of 3×3 followed by maximum pooling of size 2×2 and stride 2. The second and third layer performed convolution using 64 kernels with the same kernel and pooling size as mentioned above. The final layer used 128 kernels based on the optimization of hyperparameters. All convolutional layers used the rectified linear unit (ReLu) activation function. The last fully connected layer had two hidden layer with 2048 and 256 neurons and a softmax activation function in the output layer. Two of the most commonly used optimizers, Adaptive Moment Estimation (Adam) and Stochastic Gradient Decent (SGD), were investigated. While Adam was an optimizer that converged relatively quickly, it did so at the expense of a greater loss. SGD, on the other hand, converged more slowly, but achieved a smaller loss.

The final architecture was chosen because it achieved average precision, recall and an F_1-score of 96%, which indicated a suitable model classification. More results regarding the moth classifier are described in section 3.2.

2.2.4. Statistic

Statistics were generated based on the number of counted insects, the number of moth species found, and the number of unknown insects found (i.e. unknown to the trained CNN algorithm). The statistics were updated live as the images were processed and analyzed. Thus, the statistics were always updated through the execution of one iteration of the algorithm see figure 5. The classification phase simultaneously classified each individual and assigned labels to each individual species. That is, individuals in an image could be classified as a different moth species in the previous image. This phase ensured that the moth species most frequently classified in a track were represented in the

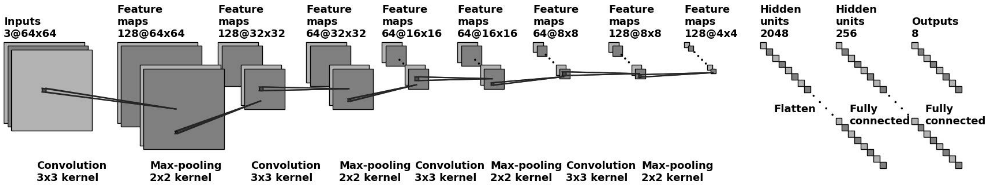


Figure 7: CNN architecture for moth species classification used an input of a 64×64 RGB image.

final statistics. Several parameters were defined and adjusted to filter noisy tracks such as insects flying close to the camera.

3. Results

An experiment was conducted in Aarhus, Denmark in the period 3 August to 20 September 2019, capturing 272,683 images in total over 48 nights. The constructed light trap was located close to the forest of Risskov, where it was active in the period from 9pm to 5am each night. The collection of image data was limited to a smaller sample of Danish moths of the family *Noctuidae*. An observation was defined as the images from a single night's footage. On average, one observation consisted of 5,675 images. It should be noted that the number of images captured in the observations varied significantly. The smallest observation consisted of 134 images, while the largest consisted of 27,300. The primary reason for this was the variation in weather conditions. While the activity of moths was limited during rainy weather or strong winds, for instance, it increased in warm late summer evenings and nights. Figure 8 shows an example of many moths in the trap on a warm summer night.

In the following sections, the results from the experiment, and the last stages of the algorithm concerning tracking, classification and statistics (as described in figure 5) will be presented.

3.1. Insect tracking and counting

Tracking was tested on an observation consisting of 6000 different images distributed over 122 minutes of insect activity. The observation was collected on the night between 25 August and 26 August 2019 and represents the average observation for a single night. The set was selected in order to provide the tracking algorithm with a sufficient challenge. The observation was challenging because the individual insects moved in and out of the camera view repeatedly. In addition, there were examples of two insects siting close together. Furthermore, some insects flew very close to the camera lens. The algorithm was therefore challenged in its ability to retain the correct tracking. The observation was manually reviewed to establish a ground truth (GT) thereby enabling evaluation of the performance. In total, a GT of 82 individual moths was observed.

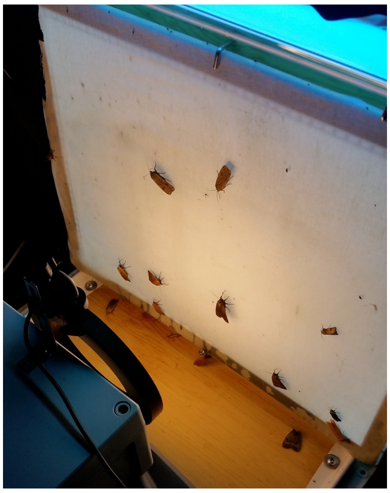


Figure 8: Photo of the light trap with moths on the sheet of the light table during a warm summer night.

The tracker algorithm measured 83 individual tracks, where 65 were identified as true positive (TP) and 18 as false positive (FP) moth tracks. Table 1 shows a calculated Tracker Detection Rate (TDR) of 79% and False Alarm Rate (FAR) of 22% based on the observation of 6,000 images.

3.2. Moth species classification

From the experiment mentioned in the beginning of this section, images of eight different species with, frequent-occurring species from different dates were selected, annotated and identified

Metric	Equation	Result
TDR	6	0.79
FAR	5	0.22

Table 1: Result of the tracking algorithm. From a ground truth (GT) of 82 moth tracks here 65 were identified as true positive (TP) and 18 as false positive (FP). The Tracker Detection Rate (TDR) and False Alarm Rate (FAR) is calculated.

No.	Moth Species	Numbers
1	Arotis puta	108
2	Amphipyra pyramedia	108
3	Autographa gamma	108
4	Caradrinina	108
5	Mythimna pallens	108
6	Noctua fimbriata	108
7	Noctua pronuba	108
8	Xestia c-nigrum	108
	Total	864

Table 2: Number of moth species in the dataset used for training and validation.

to train different CNN models. Examples of individual moth species are shown in figure 9

Table 2 shows an overview of the occurrence of all species in the chosen dataset for training and validation of the CNN algorithm.

The moth class *Caradrinina* consists of the three moth species *Hoplodrine ambigua*, *Hoplodrina blanda* and *Hoplodrina octogenaria*. These species are very similar in appearance and it was too difficult to distinguish between them from the images alone. Data augmentation was applied to all images with a flip vertical, horizontal, zoom, different illumination intensity and rotation of different degrees. This operation provided more training data and was used to create a uniform distribution of moth species. The dataset was scaled with a factor of 32 times resulting in 27,648 images where each class contained 3,456 data points after augmentation. From this dataset, 80% was used for training and 20% for validation of the CNN model.

To find the best CNN architecture for moth species classification, different hyperparameters were adjusted as described in section 2.2.3. A total of 64 architectures were trained using a dropout probability of 0.3 after the second last hidden layer. The average F_1-score for all classes was used as a measure for a given architecture's performance. Table 3 shows a summary of the architectures with the highest validation F_1-scores.

Rating	Hyperparameters	Learnable parameters	F_1-score
1.	3, 3, 128, 128, 256	714760	0.95831
2.	5, 1, 32, 128, 256	592776	0.95827
3.	3, 3, 128, 32, 256	266152	0.95817
...			
31.	5, 1, 128, 32, 512	389288	0.88462
32.	5, 3, 128, 32, 512	405672	0.88112
33.	5, 3, 32, 128, 256	658312	0.87377
34.	5, 3, 128, 128, 512	1247496	0.87249
...			
62.	3, 1, 128, 32, 512	655368	0.80969
63.	3, 3, 32, 32, 256	341544	0.79608
64.	5, 3, 32, 32, 256	591240	0.63751

Table 3: Ranking of the CNN architectures with highest and lowest F_1 classification scores. Rank 1 to 32 were trained using the Adam optimizer. Rank 32 to 64 were trained using the SGD optimizer. The hyperparameters column shows values of: {kernel size layer 1, kernel size last layer, convolutional depth layer 1, convolutional depth last layer, fully connected size}.

The three best architectures had very high F_1-scores, which only varied by 0.00024, but had a varying number of learnable parameters. Compared to SGD, Adam turned out to be the superior optimizer for training of all models as shown in table 3. In the end, the architecture with the highest amount of learnable parameters (714,760) was chosen because it had the highest F_1 score and was still fast to train. However, there was a risk of overfitting with many parameters and few training data. The chosen model shown in figure 7 had an F_1-score of 95.8%, which indicated that the trained CNN was very accurate in its predictions. The confusion matrix (figure 10) was based upon the validation of the chosen model. The confusion matrix has a diagonal trend, which indicates that the model matched the validation set well. The model had a recall of 96% indicating that only 4% of the moth species in the validation set were missed. A similar precision of 96% was obtained indicating that only 4% were wrongly classified.

3.3. Summary statistics

To evaluate the final system, including tracking and classification, the same observation was used as a test for the tracking algorithm. None of the images from this observation were used to train the

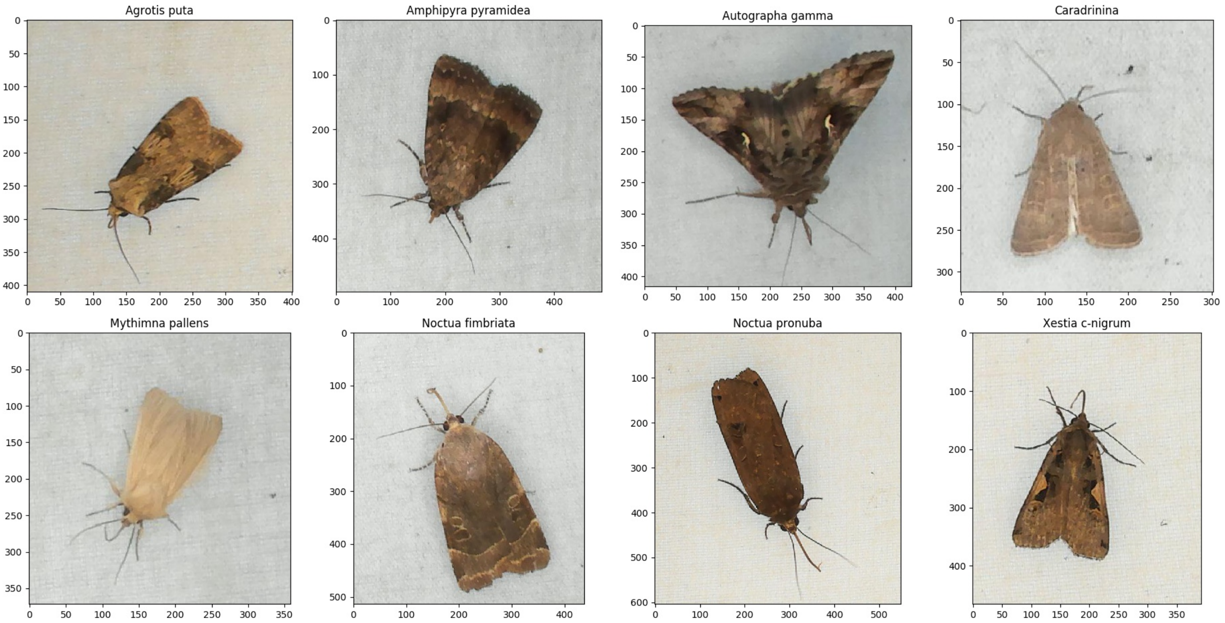
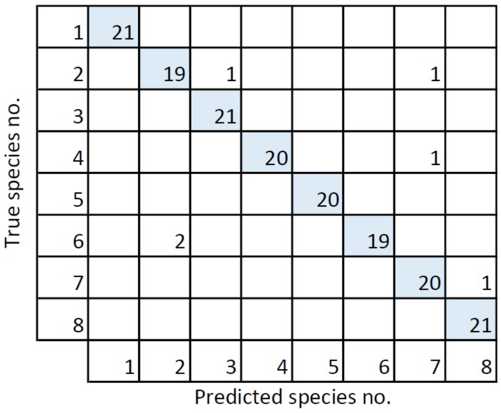


Figure 9: Examples of the eight moth species that were observed and labelled for training of the CNN models.



True species no.	1	2	3	4	5	6	7	8
1	21							
2		19	1				1	
3			21					
4				20			1	
5					20			
6		2				19		
7							20	1
8								21

Predicted species no.

Figure 10: Confusion matrix for the validation of the best model. The numbers corresponds to the species numbers in table 2

Moth Species	GT	TP	FP	FN
Arotis puta	2	2	1	0
Amphipyra pyramedia	0	0	5	0
Autographa gamma	0	0	2	0
Caradrinina	8	5	0	1
Mythimna pallens	2	0	0	2
Noctua fimbriata	10	14	3	1
Noctua pronuba	33	26	2	4
Xestia c-nigrum	22	19	2	3
Unknown	5	0	2	0
Total	82	66	17	11

Table 4: Number of individuals of each moth species in the test dataset detected by the MCC algorithm and ground truth (GT). A true positive (TP) is defined as an insect that retained the same Track ID and correct classification in the entire observation. A false positive (FP) is defined as an insect that was incorrect classified or counted several times or assigned a new Track ID. A false negative (FN) is defined as an insect that was not detected and counted in the final statistic.

CNN model. In this way, the algorithm was by this way tested on data material it had never seen before.

The observation represents a typically night, where 6,000 images were collected in the trap. The images were studied to establish a GT. This involved a manual count of occurrences for each moth

Moth Species	Precision	Recall	F_1-score
Arotis puta	0.67	1.00	0.80
Amphipyra pyra.	0.00	NA	NA
Autographa gam.	0.00	NA	NA
Caradrinina	1.00	0.83	0.91
Mythimna pallens	0.00	NA	NA
Noctua fimbriata	0.82	0.93	0.87
Noctua pronuba	0.93	0.87	0.90
Xestia c-nigrum	0.90	0.86	0.88
Unknown	0.00	NA	NA
Total	0.80	0.86	0.83

Table 5: Precision, recall and F_1-score based on the test dataset of one nights observation consisting of 6000 images.

species. Table 4 shows the algorithm's estimate of the number of moth species and the established GT of the observation. Note that the two species *Amphipyra pyramidea* and *Autographa gamma* did not occur in this observation. There were also unknown moth species in the dataset. However, none of them were detected correctly. Table 5 shows the algorithm's precision, recall and F_1-score based on the tracked and predicted moths in table 4. In total a precision of 80%, recall of 86% and F_1-score of 83% was achieved.

4. Discussion

The tracker measured a Tracking Detection Rate (TDR) of 79% which means it tracked the majority of the observed insects correctly. However, the algorithm had a False Alarm Rate (FAR) of 22% which means that nearly one quarter of the detected tracks were incorrect. The test was made on a night with average insect activity, but with many challenges and variations of movements. The MCC algorithm measured an average combination of tracking and classification F_1-score of 0.83 where recall was 6% higher than precision. This score was an accepted performance of the hole system. However, the final validation was only performed on 82 individual moths and only six of the eight species was represented.

The most frequent source of error were individual insects moving at the edge of the camera's field of view. This resulted in insects disappearing completely or partially from an observation only to potentially appear again later. Furthermore, errors occurred in cases where an insect flew close to the camera lens. In such cases, the algorithm could place multiple boxes on a single individual and make a match with these fake boxes. However, because the insects flying in front of the lens rarely stayed in the field of view for more than a few frames, the design of the tracker often prevented this type of error. An individual insect had to be detected in at least 50 images before it was counted. Consequently, a flying insect appearing in few frames was below the threshold filter value, and the final statistics were therefore not affected. The last identified source of error occurred when two individuals were located closely together in the image. In this case, the tracking algorithm could not separate the two individuals and therefore only placed one bounding box.

One of the primary sources of error in the algorithm was the dataset used for training and validation of the CNN model. One expert did a sample study of the dataset and concluded that especially the classes *Caradrinina* and *Ampipyra pyramedia* included individuals that were annotated incorrectly. This means that noise was introduced into the dataset and the CNN model must thus be expected to have had a deteriorated performance. Collecting a sufficiently large dataset with enough data points for efficient classification of the rarer species was also a significant challenge.

The current classification algorithm relies heavily on padding the bounding box found during blob segmentation. The performance of the system changes significantly with variation in padding before CNN prediction. The CNN algorithm was trained on a dataset using manual annotations of the moths. These do not surround the moths near as closely as the bounding boxes placed by the blob segmentation (see figure 6 and figure 9). Thus, there is a difference in the sizes. It is likely that accuracy could be improved with training on images cropped by the blob segmentation.

As a proof of concept, the proposed light trap and MCC algorithm is promising as an automated system for insect observations during night hours. In order to conduct better statistical analysis and draw empirical conclusions about the proposed system, there is a need for more recordings from different observations and locations. In addition, the CNN classification algorithm can be improved by ensuring that the dataset for training is annotated correctly.

5. Conclusion

In this paper, we have present a light trap and computer vision system to monitor live insects and moth species. The automated light trap was composed of a web camera, Raspberry Pi computer and special illumination to attract insects during the night. The camera and illumination was adjusted to achieve a high resolution and quality of images of eight different moth species. The light trap recorded more than 250,000 images during 48 night over the summer 2019 with an average of 5675 images per night.

A customized convolutional neural network was proposed and trained on 864 labeled images of live moths. This allowed us to detect and classify eight different species. The network achieving a high validation F_1-score of 0.96.

The algorithm measured an average classification and tracking F_1-score of 0.83 and a tracking detection rate of 0.79. This result was based on an estimate of 83 individual moths observed during one night with insect activity in 122 minutes collecting 6,000 images.

Furthermore, the paper identified potential improvements to the system where the amount of training data for the presented CNN model for species classification was highlighted as a focus area.

Overall, the proposed light trap and computer vision system showed promising results in nondestructive and automatic monitoring of moths and classification of species. It should be considered as a viable alternative to traditional methods which typically requires tedious manual labor (i.e. visiting the trap several times in a season for observation) and often rare species of insects are killed.

Acknowledgments

The work was supported by the Danish 15. Juni Fonden [grant number 2019-N-23].

References

24hshop . Usb led light ring. accessed on 1/3-2020. URL: https://www.24hshop.dk/mobiltilbehor/iphone-tilbehor/iphone-ovrigt/usb-led-ring-belysning.

Bashir F, Porikli F. Performance Evaluation of Object Detection and Tracking Systems. Mitsubishi Electric Research Laboratories 2006;TR2006-041. URL: http://www.merl.com/publications/docs/TR2006-041.pdf.

Batista GE, Campana B, Keogh E. Classification of live moths combining texture, color and shape primitives. In: Proceedings - 9th International Conference on Machine Learning and Applications, ICMLA 2010. 2010. doi:10.1109/ICMLA.2010.142.

Bioform . Leucht-kleinanlage 12v, 15w superaktinisch. accessed on 1/3-2020. URL: https://www.bioform.de/shop.php?wg=1&action=fastsearch.

Bjerge K, Frigaard CE, Mikkelsen PH, Nielsen TH, Misbih M, Kryger P. A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony. Computers and Electronics in Agriculture 2019;doi:10.1016/j.compag.2019.104898.

Chang Q, Qu H, Wu P, Yi J. Fine-Grained Butterfly and Moth Classification Using Deep Convolutional Neural Networks 2017;URL: https://pdfs.semanticscholar.org/4cf2/045b811c9e0807f9c94fc991566a6f5adbf4.pdf.

Computermester . A3 led light table. accessed on 1/3-2020. URL: https://www.computermester.dk/shop/lysborde-214c1.html.

Ding W, Taylor G. Automatic moth detection from trap images for pest management. Computers and Electronics in Agriculture 2016;123:17 – 28. URL: http://www.sciencedirect.com/science/article/pii/S0168169916300266. doi:https://doi.org/10.1016/j.compag.2016.02.003.

Eliopoulos P, Tatlas NA, Rigakis I, Potamitis I. A "smart" trap device for detection of crawling insects and other arthropods in urban environments. Electronics (Switzerland) 2018;doi:10.3390/electronics7090161.

Fox R, Parsons M, Chapman J. The State of Britain's Larger Moths 2013. Technical Report; 2013.

Franzén M, Johannesson M. Predicting extinction risk of butterflies and moths (Macrolepidoptera) from distribution patterns and species characteristics. Journal of Insect Conservation 2007;doi:10.1007/s10841-006-9053-6.

Groenendijk D, Ellis WN. The state of the Dutch larger moth fauna. Journal of Insect Conservation 2011;doi:10.1007/s10841-010-9326-y.

Harvey J, Heinen R, Armbrecht Iea. International scientists formulate a roadmap for insect conservation and recovery. Nature Ecology & Evolution 2020;doi:10.1038/s41559-019-1079-8.

He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016. doi:10.1109/CVPR.2016.90. arXiv:1512.03385.

Huang YY, Wang WY. Deep residual learning for weakly-supervised relation extraction. In: EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings. 2017. doi:10.18653/v1/d17-1191. arXiv:1707.08866.

Jonason D, Franzén M, Ranius T. Surveying moths using light traps: Effects of weather and time of year. PLoS ONE 2014;doi:10.1371/journal.pone.0092453.

Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM 2017;doi:10.1145/3065386.

Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing 2017;doi:10.1016/j.neucom.2016.12.038.

Logitec . Brio ultra hd pro webcam. accessed on 1/3-2020. URL: https://www.logitech.com/en-us/product/brio.

Macgregor CJ, Williams JH, Bell JR, Thomas CD. Moth biomass increases and decreases over 50 years in Britain. Nature Ecology and Evolution 2019;doi:10.1038/s41559-019-1028-6.

Mayo M, Watson AT. Automatic species identification of live moths. Knowledge-Based Systems 2007;doi:10.1016/j.knosys.2006.11.012.

Munkres J. Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics 1957;doi:10.1137/0105003.

Otsu N. Threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 1979;doi:10.1109/tsmc.1979.4310076.

Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016. doi:10.1109/CVPR.2016.91. arXiv:1506.02640.

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2018. doi:10.1109/CVPR.2018.00474. arXiv:1801.04381.

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. 2015. arXiv:1409.1556.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2015. doi:10.1109/CVPR.2015.7298594. arXiv:1409.4842.

Wagner DL. Insect Declines in the Anthropocene. Annual Review of Entomology 2020;doi:10.1146/annurev-ento-011019-025151.

Watson A, O'neill M, Kitching I. Automated identification of live moths (Macrolepidoptera) using digital automated identification system (DAISY). Systematics and Biodiversity 2004;doi:10.1017/S1477200003001208.

Xia D, Chen P, Wang B, Zhang J, Xie C. Insect detection and classification based on an improved convolutional neural network. Sensors (Switzerland) 2018;doi:10.3390/s18124169.

Zhong Y, Gao J, Lei Q, Zhou Y. A vision-based counting and recognition system for flying insects in intelligent agriculture. Sensors (Switzerland) 2018;doi:10.3390/s18051489.