

Article

Audio, Image, Video, and Weather Datasets for Continuous Electronic Beehive Monitoring

Vladimir Kulyukin

Department of Computer Science, Utah State University, Logan, UT 84321, USA;
vladimir.kulyukin@aggiemail.usu.edu

Abstract: In 2014, we designed and implemented BeePi, a multi-sensor electronic beehive monitoring system. Since then we have been using BeePi monitors deployed at different apiaries in northern Utah to design audio, image, and video processing algorithms to analyze forager traffic in the vicinity of Langstroth beehives. Since our first publication on BeePi in 2016, we have received multiple requests from researchers and practitioners for the datasets we have used in our research. The main objective of this article is to provide a comprehensive point of reference to the datasets that we have so far curated for our research. We hope that our datasets will provide stable performance benchmarks for continuous electronic beehive monitoring, help interested parties verify our findings and correct errors, and advance the state of the art in continuous electronic beehive monitoring and related areas of AI, machine learning, and data science.

Keywords: electronic beehive monitoring; audio classification; image classification; video processing; audio datasets; image datasets; video datasets; weather datasets; machine learning; data science; apis mellifera



Citation: Kulyukin, V. Audio, Image, Video, and Weather Datasets for Continuous Beehive Monitoring. *Appl. Sci.* **2021**, *11*, 4632. <https://doi.org/10.3390/app11104632>

Academic Editor: Amerigo Capria

Received: 29 April 2021

Accepted: 14 May 2021

Published: 19 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In 2014, we designed and built BeePi [1–3], a multi-sensor electronic beehive monitoring (EBM) system. While BeePi can be used to monitor bee traffic in the vicinity of beehives of various common designs (e.g., Dadant, Top-Bar, Langstroth [4]), all our EBM research has been conducted on honeybee colonies in Langstroth hives [5]. In this article, we use the terms *hive* and *beehive* to refer to a Langstroth beehive and use the terms *bee* and *honeybee* to refer to the *Apis Mellifera* honeybee [6].

The original BeePi monitor consisted of a raspberry pi 2 computer, a pi T-Cobbler, a breadboard, a waterproof temperature sensor, a pi camera, a ChronoDot clock, and a Neewer 3.5 mm mini lapel microphone placed above the landing pad of a Langstroth hive. All BeePi hardware components, after being soldered and connected with jumper cables, fit in a single Langstroth super (i.e., a wooden box of specific dimensions). Small holes were drilled in the super's walls for hardware ventilation. Since 2014 we have been using the BeePi platform to design audio, image, and video processing algorithms to analyze forager traffic in the vicinity of Langstroth hives [7–11]. With the exception of the raspberry pi computer model, which has been upgraded in some deployed BeePi monitors to pi 3 or to pi 4, we continue to use the same hardware design.

Reproducibility and replicability have been two fundamental objectives of the BeePi project from its inception: other researchers, practitioners, and citizen scientists must be able to reproduce our experiments and replicate our designs at minimum costs and time commitments, which is why all BeePi monitors are built with off-the-shelf hardware components and the BeePi software algorithms are developed on top of open source packages with no license fees. Interested readers are referred to [1,2] for BeePi hardware design diagrams, photos, technical specifications, and assembly videos.

Another fundamental principle guiding our research is the sacredness of honeybee space: the deployment of sensors cannot interfere with the natural cycles of honeybees. This

principle prevents us from deploying sensors inside the beehive or directly on honeybees (e.g., RFID labels on foragers [12,13]).

To preserve the objectivity of our data and observations, we do not intervene in the life cycle of the monitored colonies. For example, we do not apply any chemical treatments (e.g., Varroa mite or hive beetle treatments) to our colonies or re-queen failing or struggling colonies. When we hive new bee packages in late April or early May, we place a small jar of natural raw honey into each new colony as a nutritional supplement at the beginning of the season. No further treatment is applied to honeybee colonies during the beekeeping season.

Since our first peer-reviewed publication on the BeePi project in 2016 [3], we have been receiving requests from researchers, practitioners, and citizen scientists for the datasets we have curated for our research. The main objective of this article is to provide a comprehensive point of reference to the datasets we have so far curated and used in our experiments. We hope that our datasets will not only provide benchmarks for continuous EBM, but also help interested parties verify our findings, discover and correct errors, and advance the state of the art in EBM and related areas of AI, machine learning (ML), and data science.

BeePi monitors thus far have had seven field deployments. The first deployment was in Logan, Utah (UT), USA (41.7370° N, 111.8338° W) in September 2014 when a single BeePi monitor was placed on an empty Langstroth hive for two weeks to test our hardware design and data collection software. The second deployment was in Garland, UT, USA (41.73638° N, -112.16292° W) from December 2014 to January 2015 when a BeePi monitor was placed on a hive with overwintering honeybee colony and successfully collected ≈ 200 MB of data. The third deployment was in North Logan, UT (41.7694° N, 111.8047° W) from April to November 2016 when four BeePi monitors were placed on four beehives at two apiaries and acquired ≈ 20 GB of data. The fourth deployment was in Logan and North Logan, UT (April–September 2017), when four BeePi units were placed into four beehives at two apiaries to collect ≈ 220 GB of data. The fifth deployment started in April 2018, when four BeePi monitors were placed on four beehives at an apiary in Logan, UT. In September 2018, we decided to keep the monitors deployed through the winter to stress test our hardware and software in the harsh winter weather conditions of northern Utah. By May 2019, we had collected ≈ 400 GB of raw image, video, audio, and temperature data. The sixth field deployment started in May 2019 with four freshly installed Russian bee packages and ended in March 2021 with ≈ 350 GB of data collected. Two Russian colonies died in March 2021. The seventh deployment started in late April 2021 with four BeePi monitors: two monitors on the two older Russian colonies that survived and two new Russian colonies hived in late April 2021.

During the first four deployments in 2014–2017, we experimented with three types of power supply in BeePi: solar, battery, and grid [14]. A deployed BeePi monitor requires ≈ 440 mA to power its pi computer and the three sensors connected to it: the temperature sensor, the microphone, and the pi camera. The hardware clock is powered by its own button-size battery that lasts ≈ 1 year. The amount of current drawn by the temperature sensor and the microphone appears to be insignificantly small insomuch as connecting and disconnecting these sensors does not appear to change the measurable amount of drawn current. With the camera unit disconnected, the amount of drawn current fluctuates between ≈ 310 mA and ≈ 330 mA. Consequently, we estimate the camera to draw ≈ 120 mA of 440 mA drawn by a single BeePi monitor.

In the solar version of BeePi, a solar panel was placed either on top of or next to a beehive (See Figure 1). For solar harvesting, we experimented with Renogy 50 Watts 12 Volts monocrystalline solar panels, Renogy 10 Amp PWM solar charge controllers, and Renogy 10 ft 10 AWG solar adaptor kits. We also experimented with two rechargeable batteries: the UPG 12 V 12 Ah F2 lead acid AGM deep cycle battery and the Anker Astro E7 26,800 mAh battery. Our field experiments in 2016–2017 [7,8,14] convinced us that solar power was not a viable option in northern Utah. While some experiments ran to completion and allowed us to acquire audio, video, and temperature data, we found solar

power harvesting to be unreliable for continuous data collection over longer time periods (e.g., at least one calendar month without interruptions).

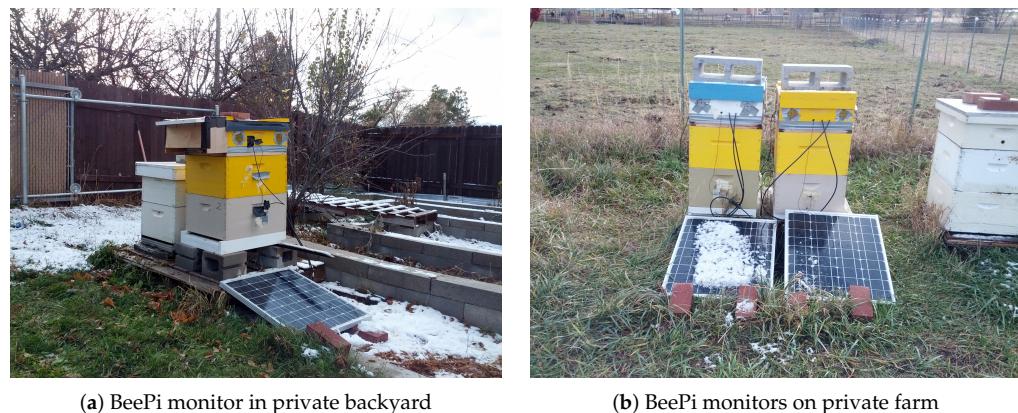


Figure 1. BeePi monitor (a) with a solar panel on the side; two BeePi monitors with solar panels behind the hives.

Rechargeable batteries also had notable drawbacks. In particular, we discovered that many UPG batteries stopped holding charge in cold ($\leq -15^{\circ}\text{C}$) or hot ($\geq +25^{\circ}\text{C}$) temperatures. We found the Anker battery's performance to be significantly better than that of the UPG battery. However, a fully charged Anker battery can power one deployed BeePi monitor for ≈ 24 h, after which it must be replaced with a new fully charged battery, which is not acceptable to many researchers and practitioners who drive long distances to their apiaries. For these reasons, all deployed BeePi monitors have been powered from the grid since 2017.

In the subsequent sections of this article, we specify how we have curated our datasets, describe the experiments in which we have used them, and briefly summarize our findings. We refer interested readers to the citations of our prior EBM research in the introduction and the subsequent sections for detailed formal and experimental treatments of our results. In Section 2, we present our audio datasets we have used in our research on audio beehive monitoring. In Section 3, we describe our image datasets we have used in our research on omni-directional bee traffic. In Section 4, we describe our video datasets we have used in our research on omni-directional and directional honeybee traffic. In Section 5, we describe our weather dataset curated in 2020 that we are currently using in our experiments to align the weather data with video-based honeybee traffic curves. In Section 6, we summarize our data curation efforts and provide information on the availability of our datasets.

2. Audio Datasets

We obtained our first audio datasets from six BeePi monitors deployed in Logan, UT and North Logan, UT on Langstroth hives with Carniolan and Italian honeybee colonies in 2017–2018 [8]. We placed the microphones ≈ 10 cm above the hives' landing pads (See Figure 2). Each monitor saved a 30-s audio wav file every 15 min on a USB storage device connected to the monitor's pi computer. Each 30-s audio sample was automatically segmented into 2-s wav samples with a 1-s overlap, which resulted in 28 2-s wav samples per one 30-s audio file.

We obtained the ground truth by manually labeling 2-s audio samples. Three human listeners listened to each sample and placed it into one of the three non-overlapping categories: bee buzzing (B), cricket chirping (C), and ambient noise (N). The B category consisted of the samples where at least two listeners heard bee buzzing. The C category consisted of the audio files collected at night where at least two listeners heard the chirping of crickets and no bee buzzing. The N category included all samples where none of the human listeners could clearly hear either bee buzzing or cricket chirping. The N category included samples with static microphone noise, thunder, wind, rain, vehicles, human

conversation, sprinklers, and other types of ambient noise. We used the same curation techniques on all audio datasets described in this article.

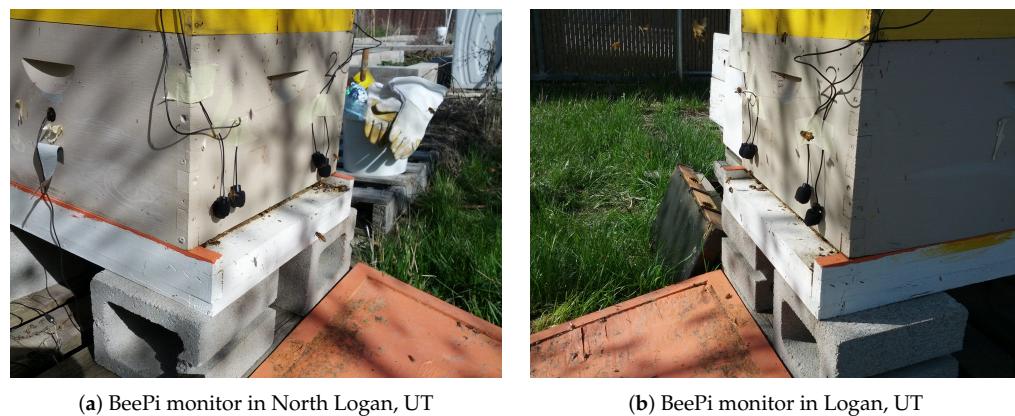


Figure 2. Microphones placed ≈ 10 cm above landing pads in BeePi monitors in North Logan, UT (a) and Logan, UT (b).

We called the first labeled dataset BUZZ1. As shown in Table 1, this dataset includes 10,260 audio samples: 6494 training samples (63%), 2616 testing samples (25%), and 1150 validation samples (12%) used for model selection. The samples in the validation dataset are separated from the audio samples in the training and testing samples by beehive and location.

Table 1. Audio sample distribution in BUZZ1.

	Bee (B)	Cricket (C)	Noise (N)	Total
Training	2128	2128	2238	6494
Testing	872	872	872	2616
Validation	300	500	350	1150
Total	3300	3500	3460	10,260

Several months later we curated another dataset, which we called BUZZ2, of 12,914 audio samples by taking 7582 (76.4%) labeled samples for training from a beehive in one apiary and 2332 (23.52%) labeled samples for testing from a different beehive in a different apiary. The sample distribution of BUZZ2 is given in Table 2. All training and testing data were obtained from Italian honeybee colonies in 2017 whereas the validation data for model selection were obtained from two Carniolan colonies in 2018. Thus, in BUZZ2, the train/test samples are separated by beehive and location while the validation beehives are separated from the train/test beehives by beehive, location, time (2017 vs. 2018), and bee race (Italian vs. Carniolan).

Table 2. Audio sample distribution in BUZZ2.

	Bee (B)	Cricket (C)	Noise (N)	Total
Training	2402	3000	2180	7582
Testing	898	500	934	2332
Validation	1000	1000	1000	3000
Total	4300	4500	4114	12,914

We used BUZZ1 and BUZZ2 in our raw audio classification experiments with convolutional neural networks (ConvNets) and four standard ML methods: logistic regression (LR), k-nearest neighbors (KNN), support vector machines (SVM), and random forests (RF). All ConvNets were trained, tested, and validated on raw audio data (i.e., amplitude vectors) without any feature extraction. The standard ML models were trained on feature vectors of 193 elements extracted with Librosa [15], an open source Python library for audio analysis: 40 Mel Frequency Cepstral Coefficients (MFCCs), 12 chroma coefficients, 128 mel-spectrogram coefficients, seven spectral contrast coefficients, and six tonnetz coefficients.

On BUZZ1, a shallower ConvNet with a custom layer outperformed three deeper ConvNets and performed on par with the standard ML methods trained to classify feature vectors extracted from raw audio samples. On BUZZ2, a more challenging audio dataset, all ConvNets outperformed the four ML methods and a ConvNet trained to classify spectrogram images of audio samples. We observed that a major trade-off between deep learning (DL) and standard ML was between feature engineering and training time: while the ConvNets required no feature engineering and generalized better on raw audio files (i.e., amplitude vectors), they took considerably more time to train than the standard ML methods.

To continue our investigation of audio beehive monitoring, we curated two more audio datasets (BUZZ3 and BUZZ4) in 2019 from the audio samples acquired by different BeePi monitors in Logan and North Logan, UT. BUZZ3 includes 15,254 audio samples manually labeled as B (5121 samples), C (5346 samples), and N (4787 samples). In 2020, we augmented BUZZ3 with a fourth category of lawn mowing (L) into which we placed 3340 audio samples where two out of the three human listeners could hear a lawn mower's sound. Thus, BUZZ3 is a proper subset of BUZZ4. The sample distribution of BUZZ3 and BUZZ4 are given in Table 3.

Table 3. Audio sample distribution in BUZZ3 (first three columns) and BUZZ4 (first four columns).

	Bee (B)	Cricket (C)	Noise (N)	Lawn (L)	Total
Training	2880	3600	2520	2120	11,120
Testing	1071	577	1098	840	3586
Validation	1170	1169	1169	380	3888
Total	5121	5346	4787	3340	18,594

We used BUZZ1, BUZZ2, and BUZZ3 to investigate whether automated feature engineering could improve standard ML methods to perform on par with DL methods [16]. We experimented with recursive feature elimination (RFE) [17], sequential feature selection (SFS) [18], relief-based feature selection [19], and RF feature selection [20] to find optimal feature subsets. Our feature space included thirty-four audio features (e.g., zero crossing rate, energy, spectral flux, MFCCs, etc.) extracted with the pyAudioAnalysis library [21]. We confined our investigation to LR, KNN, SVM, and RF and compared their performance with that of the top performing ConvNets we had previously trained on BUZZ1, BUZZ2, and BUZZ3 [8].

Table 4 shows the accuracies of our models on BUZZ1, BUZZ2, and BUZZ3 with the thirteen MFCCs selected by all automated feature engineering methods. On BUZZ1, the best validation accuracy of 98.43% was achieved by the RF with 100 decision trees; on BUZZ2, the best validation accuracy of 95.33% was achieved by LR; on BUZZ3, the best validation accuracy of 97.91% was achieved by RF.

Table 4. Model accuracies with 13 MFCCs; best validation accuracies are in bold.

Model	Dataset	Test. Acc. (%)	Valid. Acc. (%)
KNN	BUZZ1	99.58	83.21
	BUZZ2	98.84	90.83
	BUZZ3	77.49	86.68
RF (100 trees)	BUZZ1	99.54	98.43
	BUZZ2	99.05	90.46
	BUZZ3	79.16	97.91
LR	BUZZ1	96.11	94.78
	BUZZ2	77.78	95.33
	BUZZ3	79.97	95.32
SVM (One vs. Rest)	BUZZ1	99.17	98.00
	BUZZ2	97.64	72.46
	BUZZ3	85.21	95.55

We also compared the accuracies of the best performing models on each dataset with the accuracies of the best performing ConvNet that we had previously trained, tested, and validated on the same datasets [8]. Table 5 gives the model accuracies. The RF model slightly outperformed the ConvNet on BUZZ1 and BUZZ2 while the ConvNet slightly outperformed LR on BUZZ2. The validation accuracies of all models were above 95% on all datasets. A lesson we learned from this investigation is that standard ML models are a viable alternative to ConvNets on these datasets, because they train much faster than ConvNets and produce smaller RAM and disk memory footprints.

Table 5. ConvNet, RF, and LR accuracies on BUZZ1, BUZZ2, and BUZZ3; best validation accuracies are in bold.

Dataset	Model	Test. Acc. (%)	Valid. Acc. (%)
BUZZ1	ConvNet	99.93	95.21
	RF	99.54	98.43
BUZZ2	ConvNet	95.67	96.53
	LR	77.78	95.33
BUZZ3	ConvNet	93.04	96.97
	RF	79.16	97.91

In 2020, we continued our investigation of optimal feature sets for audio classification with standard ML models and experimented with RFE and SFS [22]. RFE is a wrapper-based method that iteratively fits a given model to a given dataset, computes the feature importance coefficient of each feature, and eliminates a specified number of least important features until the target number of features is selected. We varied the target number of features (i.e., the hyperparameter `n_features_to_select` in the `scikit-learn` library) from 1 to 34 and eliminated 1 feature at each iteration. The feature importance is provided by the `scikit-learn` model object's `coeff_` or `feature_importances_` attribute [23]. RFE cannot be used on models (e.g., KNN) that do not implement these attributes.

SFS is a greedy search method that provides a reasonable alternative to the exhaustive search through each feature subset of the feature power set, whose cardinality, in our case, is 2^{34} . SFS reduces the dimensionality of the feature space by adding or removing one feature at a time on the basis of the model's performance. The iteration stops when the target number of features is selected.

There are two types of SFS methods: forward selection and backward selection. Forward selection starts with an empty set of features. Given a model M and n features, n instances of M are trained for each feature. The validation accuracy of each instance is

computed and the feature that results in the greatest classification accuracy is added to the initially empty set of optimal features. The process continues with $n - 1$ instances of the model M trained with each of the remaining $n - 1$ features and the previously selected feature, and the next best feature is added to the set of optimal features. The process continues until the target number of features is selected.

The backward selection method is the reverse of the forward selection. It trains a model on the entire feature set first and then removes one feature per iteration so that the feature being removed is the one that contributes the most to the model's classification accuracy upon its removal. This process continues until the target number of features is selected. In our investigation, we used sequential forward selection. Thus, the abbreviation SFS in the remainder of the article refers to sequential forward selection.

Table 6 summarizes the results of our experiments with RFE and SFS on BUZZ1, BUZZ2, BUZZ3, and BUZZ4. On BUZZ1, the best validation accuracy was achieved by an RF of 100 trees for which 5 features were selected with RFE. On BUZZ2 and BUZZ3, the best validation accuracies were achieved by KNN with 7 and 6 features, respectively, selected by SFS. On BUZZ4, the best validation accuracy was achieved with SVM with 11 features selected by RFE.

Table 6. Datasets, models, feature selection methods (Feat. Sel.), numbers of features (Num. Feat.), and test and validation accuracies on BUZZ1, BUZZ2, BUZZ3, BUZZ4.

Dataset	Model	Feat. Sel.	Num. Feat.	Test. Acc. (%)	Valid. Acc. (%)
BUZZ1	RF (100 trees)	RFE	5	99.82	99.08
BUZZ2	KNN	SFS	7	96.81	95.43
BUZZ3	KNN	SFS	6	83.21	97.54
BUZZ4	SVM	RFE	11	71.75	95.65

Table 7 shows the features selected by RFE and SFS for the best models on each dataset. On BUZZ1, for the RF model, feature 5 (spectral entropy) and MFCC features 10–15 were selected with RFE; on BUZZ2, for the KNN model, feature 5 (spectral entropy) and MFCC features 12–15, MFCC feature 18, and MFCC feature 20 were selected with SFS; on BUZZ3, for the KNN model, feature 5 (spectral entropy) and MFCC features 10–14 and MFCC feature 16 were selected with SFS; on BUZZ4, for the SVM model, feature 3 (spectral centroid), feature 4 (second moment of spectrum), feature 5 (spectral entropy), and MFCC features 9, 11–14, and 18–20 were selected with RFE. This investigation corroborated our earlier finding [16] that MFCCs are useful features in standard ML models trained to separate bee buzzing from other audio categories in external audio beehive monitoring.

Table 7. SFS-based features for best models on BUZZ1, BUZZ2, BUZZ3, BUZZ4; feature numbers correspond to the indices of the feature array extracted with pyAudioAnalysis library [21].

Dataset	Model	RFE/SFS Features
BUZZ1	RF (100 trees)	RFE: [5, 10, 13, 14, 15]
BUZZ2	KNN	SFS: [5, 12, 13, 14, 15, 18, 20]
BUZZ3	KNN	SFS: [5, 10, 12, 13, 14, 16]
BUZZ4	SVM	RFE: [3, 4, 5, 9, 11, 12, 13, 14, 18, 19, 20]

3. Image Datasets

In 2018–2019, we started designing algorithms to analyze omnidirectional bee traffic in videos taken in the vicinity of landing pads of Langstroth hives. In [9], we defined omnidirectional bee traffic as

"... the number of bees moving in arbitrary directions in close proximity to the landing pad of a given hive over a given period of time."

We designed a two-tier algorithm (2TA) to count bee motions in bee traffic videos in the vicinity of a Langstroth hive. The 2TA combines class-agnostic motion detection (tier 1) with class-specific image classification (tier 2) (See Figure 3). Tier 1 generates a set of regions where moving objects may be present. Tier 2 applies trained class-specific classifiers (e.g., ConvNets and RFs) to the image regions centered around the motion points generated by tier 1 to detect the presence or absence of specific objects (e.g., bees) in the regions. In tier 1, we used three motion detection algorithms in OpenCV 3.0.0 (i.e., KNN [24], MOG [25], and MOG2 [26]). In tier 2, we used trained ConvNets, SVMs, and RFs. Each motion region generated in tier 1 is classified by a trained classifier into two classes: BEE or NO_BEE.

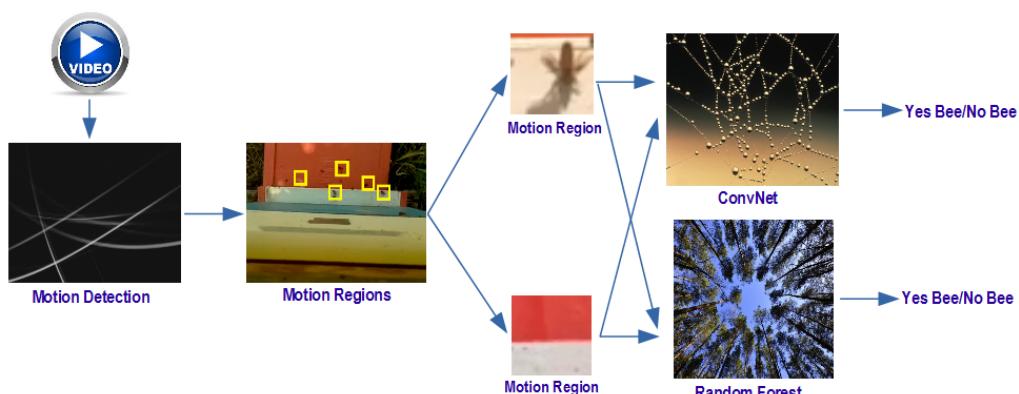


Figure 3. Bee motion counting in 2TA: tier 1 does motion region generation; tier 2 does motion region classification.

We curated two image datasets to train, test, and validate the tier 2 classifiers. The first dataset, which we called BEE1, consists of 54,382 32×32 images [27]. We manually labeled each of the 54,382 images with two categories: BEE, if the image contained at least one complete bee, or NO_BEE, if it contained no complete bee or only a small part of a complete bee (See Figure 4). The BEE1 images were obtained from 40 videos randomly selected from the video dataset of ≈ 3000 videos captured by four BeePi monitors deployed on four Langstroth hives (See Figure 5). All four monitors were deployed on hives with Italian colonies. Each video had a resolution of 360×240 with a frame rate of ≈ 25 frames per second. Two monitors were deployed in an apiary in North Logan, UT and the other two in an apiary in Logan, UT, from April 2017 to September 2017. The two apiaries were ≈ 17 km apart. We randomly selected 19,082 BEE and 19,057 NO-BEE images for training; 6362 BEE and 6362 NO_BEE for testing, and 1801 BEE and 1718 NO_BEE for validation. We used the training and testing images for model fitting and the validation images for model selection. We ensured that the data for the training and testing datasets and the data for the validation datasets came from different hives.

The second dataset, which we called BEE2, contained 112,879 images obtained from the videos acquired by four BeePi monitors on four Langstroth beehives with Carniolan honeybee colonies in Logan, UT in May and June 2018. All colonies were hived in late April 2018. A total of 5509 1-super videos and 5460 2-super videos were obtained with the BeePi monitors. We refer to a video as 1-super when it is captured by a BeePi monitor mounted on a hive that consists of one deep Langstroth super. We refer to a video as 2-super when it is captured by a BeePi monitor mounted on a hive that consists of two deep Langstroth supers. All videos had a 1920×1080 resolution. The 1-super videos were acquired from 5 May 2018 to the first week of June 2018 when the second supers were mounted on all four beehives and the BeePi monitors started acquiring videos from the second supers. All

2-super videos were captured from the first week of June 2018 to the end of the first week of July 2018.

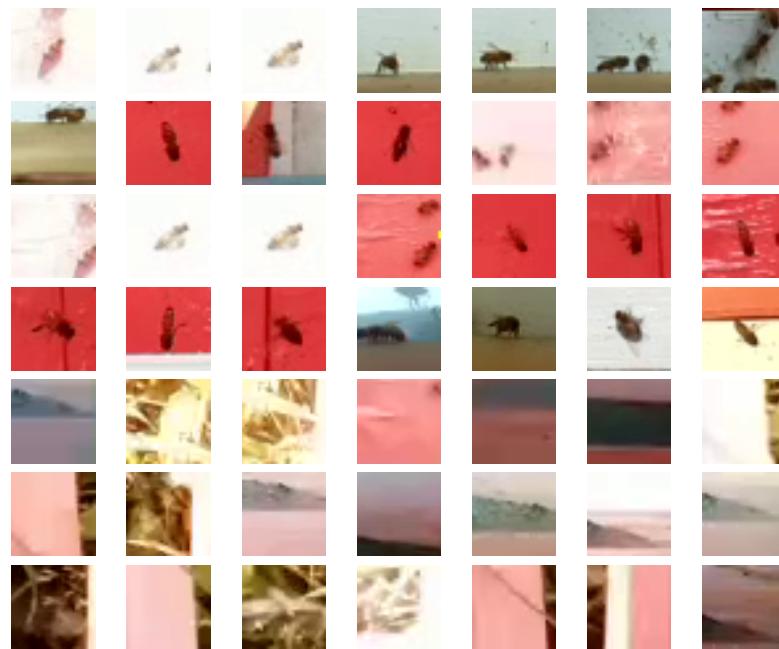


Figure 4. 49 sample images from BEE1 dataset used for experiments in [9]; first 4 rows are images manually labeled as BEE; last 3 rows are images manually labeled as NO_BEE.



(a) BeePi camera looking down on hive's pad (b) BeePi camera in wooden box protecting it against elements

Figure 5. BeePi monitor's camera looks down on hive's landing pad (a) and is protected along five sides against elements with custom wooden box (b).

We randomly selected 50 1-super videos and 50 2-super videos. We obtained the ground truth classification using the MOG2 algorithm to automatically extract 58,201 150×150 detected motion regions from the 1-super videos and 54,678 90×90 detected motion regions from the 2-super videos. We manually labeled each region as BEE (if it contained at least one complete bee) or NO_BEE (if it contained no complete bees or only a small part of a complete bee). Figures 6 and 7 show samples of 1-super and 2-super images, respectively, in BEE2. We called the 1-super image dataset as BEE2_1S and the 2-super image dataset as BEE2_2S. Table 8 gives the exact numbers of labeled images in BEE2_1S and BEE2_2S used for training, testing, and validation. In both BEE2_1S and BEE2_2S,

the images used for training and testing, on the one hand, and validation, on the other, came from different hives.

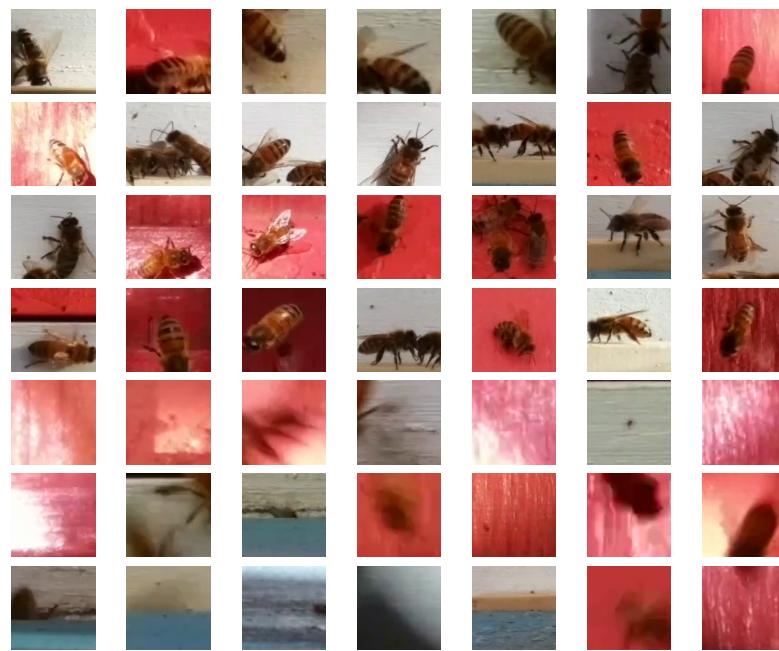


Figure 6. Sample of 1-super images from BEE2 (BEE2_1S) we used in [9]; the first four rows include images classified as BEE; the last three rows consist of images classified as NO_BEE.



Figure 7. Sample of 2-super images from BEE2 (BEE2_2S) we used in [9]; first four rows include images classified as BEE; last three rows consist of images classified as NO_BEE.

We used BEE1 to train, test and validate on several hundred automatically and manually designed ConvNets in Python 3.4 with TFlearn [28] on an Ubuntu 16.04.S LTS computer with an AMD Ryzen 7 1700X Eight-Core Processor with 16 GiB of DDR4 RAM and a GeForce GTX 1080 Ti GPU with 11 GB of onboard memory. All ConvNets were trained for 50 epochs with a batch size of 50 and normalized all images to have pixel values on each channel to be $[0, 1]$. We compared the performance of the ConvNets, RFs,

and SVMs on BEE1. We designed RFs and SVMs with the `scikit-learn` library. We varied the number of trees in RFs from 20 to 100 in increments of 20 and trained and tested all RFs with the gini entropy function. We then used BEE2 to train, test, and validate the same ConvNets, RFs, and SVMs on BEE2_1S and BEE2_2S [29].

Table 8. Numbers of images in BEE2_1S and BEE2_2S (collectively referred to as BEE2) used for training, testing, and validation.

	Trn/Tst (BEE)	Trn/Tst (NO_BEE)	Vld (BEE)	Vld (NO_BEE)
BEE2_1S	11,094	36,143	8298	2666
BEE2_2S	17,176	21,310	6823	9369
Total	28,270	57,453	15,121	12,035

Table 9 summarizes the best validation accuracies in each model category on BEE1, BEE2_1S, and BEE2_2S. On BEE1, the best automatically designed ConvNet had a validation accuracy of 99.09%; the best manually designed ConvNet had a validation accuracy of 99.45% on BEE1. The best performing RF on BEE1 had 40 trees and had a validation accuracy of 93.67%. All SVMs used the linear kernel with the max-iter parameter varying from 10 to 1000, an L2 penalty, a squared hinge loss function, and a tolerance of 0.0001. The best SVM had a validation accuracy of 63.66%. On BEE2_1S, the best ConvNet achieved a validation accuracy of 94.08%, the best RF had 100 trees and achieved a validation accuracy of 74.29%, and the best SVM had a validation accuracy of 69.36%. On BEE2_2S, the best ConvNet had a validation accuracy of 78.90%, the best RF had 60 trees and achieved a validation accuracy of 64.02%, and the best SVM achieved a validation accuracy of 64.53%.

Table 9. Validation accuracies of each model type on BEE1, BEE2_1S, and BEE2_2S; best accuracies are in bold.

Model	BEE1 Acc. (%)	BEE2_1S Acc. (%)	BEE2_2S Acc. (%)
ConvNet	99.45	94.08	78.90
RF	93.67	74.29	64.02
SVM	63.66	69.36	64.53

The performance of all classifiers on BEE1 was better than their performance on BEE2. An important qualitative difference between BEE1 and BEE2 is that the validation datasets of BEE2 contain more images of bee shadows than the validation set of BEE1. The image size of BEE1 (32×32) is smaller than the image size of BEE2 (64×64), which indicates that ConvNets, RFs, and SVMs may generalize better on smaller images than on larger ones.

In 2019, we curated another dataset, which we called BEE3 [30] (See Figure 8), taking another random sample of 50 1-super and 50 2-super videos acquired from four Langstroth beehives with Carniolan honeybee colonies in Logan, UT in May and June 2018. We labeled the acquired images from these videos with three labels: BEE, NO_BEE, and SHADOW_BEE. The first two categories were used in the same way as in BEE1 and BEE2. The third category was used on images where two (out of three) human evaluators detected the shadow of a bee. Since the numbers of images labeled as SHADOW_BEE was small compared to the numbers of images labeled as BEE or NO_BEE, we took another random sample of 50 videos taken between 12:00 and 16:00 p.m. in the same apiary and manually cropped regions with bee shadows to increase the numbers of images in the SHADOW_BEE category. Table 10 gives the final distribution of images in BEE3 in each category.

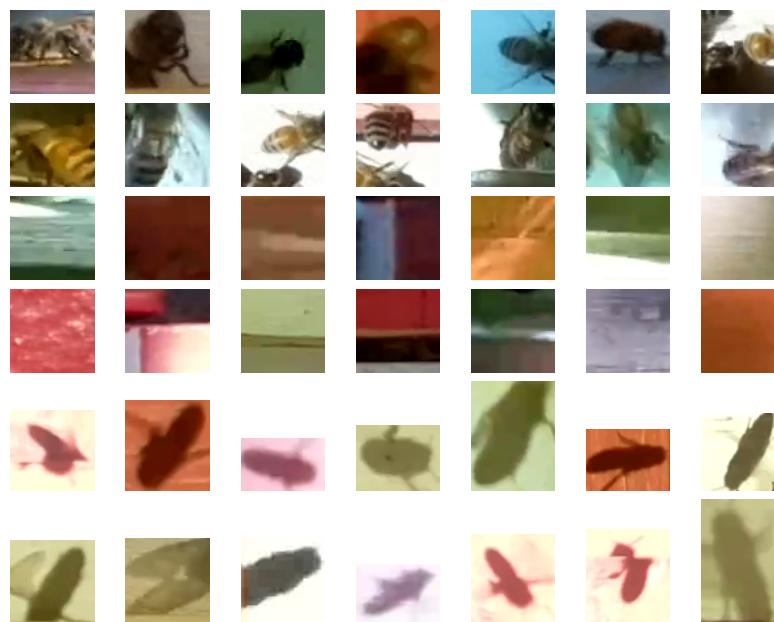


Figure 8. Sample of images from BEE3; first two rows include images classified as BEE; second two rows are images classified as NO_BEE; third three rows are images classified as BEE_SHADOW.

Table 10. Numbers of images in BEE3 used for training, testing, and validation.

Label	Train	Test	Validation	Total
BEE	12,948	4236	5187	22,371
NO_BEE	12,857	4242	5179	22,278
SHADOW_BEE	12,006	3993	4350	16,749
TOTAL	37,811	12,471	14,716	64,998

We tested 25 different ConvNets on BEE3 [30], including our own manually and automatically designed ConvNets as well as ResNet 32 [31], AlexNet [32], and VGG 16 [33]. We compared their performance with RFs and the best SVM One-Vs-Rest (OVR) classifier with a linear kernel. All ConvNets were implemented, trained, tested, and validated in Python 3.4 with TFlearn [28] on an Ubuntu 16.04.S LTS computer with an AMD Ryzen 7 1700X Eight-Core Processor with 16 GiB of DDR4 RAM and a GeForce GTX 1080 Ti GPU with 11 GB of onboard memory. All ConvNets were trained for 50 epochs with a batch size of 50 and normalized all images to have pixel values on each channel to be [0, 1].

We compared the performance of the ConvNet models on BEE3 with that of RFs and SVMs implemented with the *scikit-learn* library. We varied the number of trees in RFs from 20 to 100 in increments of 20 and trained and tested all RFs with the gini entropy function. Table 11 gives the accuracies of the best models of each type on BEE3. ResNet was the top ConvNet with a validation accuracy of 91.00%. The best RF had 80 trees and achieved a validation accuracy of 83.36%. The SVM OVR classifier with a linear kernel achieved a validation accuracy of 65.34%.

Table 11. Test and validation accuracies of best models of each type on BEE3; best accuracies are in bold.

Model	BEE3 Test Acc. (%)	BEE3 Valid. Acc. (%)
ResNet 16	96.58	91.00
RF (80 trees)	92.67	83.36
SVM (OVR)	73.53	65.34

4. Video Datasets

In 2020, we started applying concepts of particle image velocimetry (PIV) to the analysis of honeybee traffic videos [10]. Our first algorithm used PIV to compute motion vectors, classified them as incoming, outgoing, or lateral by vector direction, and returned the classified vector counts as measurements of directional traffic levels.

To evaluate this algorithm, we created our first video dataset, which we called BEE_VID1, that consists of four 30-s videos. We took ≈ 3500 timestamped 30-s bee traffic videos acquired by two deployed BeePi monitors in Logan, UT in June and July 2018. Each video had $744,640 \times 480$ frames. From this collection, we took four random samples of 30 videos each: (1) a sample from the early morning (06:00–08:00); (2) a sample from the early afternoon (13:00–15:00); (3) a sample of videos from the late afternoon (16:00–18:00); (4) a sample from the evening (19:00–21:00). From each of the four video samples, we randomly selected one video. Thus, we acquired one early morning video, one early afternoon video, one late afternoon video, and one evening video. The total frame count for the four videos is 2976 frames. We labeled the first video as no traffic (NT_VID), the second video as medium traffic (MT_VID), the third video as high traffic (HT_VID), and the fourth video as low traffic (LT_VID), which reflects the general bee traffic patterns we have observed in multiple videos acquired with different BeePi monitors at different apiaries in northern Utah in 2018–2020.

For each of these four videos, we manually counted full bee motions, frame by frame, in each video. A full bee motion is the change of position of a complete honeybee body in a given frame taken at time $t > 1$ (i.e., F_t) relative to the previous frame taken at time $t - 1$ (i.e., F_{t-1}). The number of bee motions in the first frame of each video F_1 is taken to be 0. In each subsequent frame, we manually counted the number of full bees that made any motion when compared to their positions in the previous frame. We also counted as full bee motions complete bee bodies appearing in F_t and not present in F_{t-1} (e.g., when a bee flies into the camera's field of view when F_t is captured).

Manual count of full bee motions to obtain the ground truth was labor intensive: it took us ≈ 2 h to count bee motions in NT_VID, ≈ 4 h in LT_VID, ≈ 5.5 h in MT_VID, and ≈ 6.5 h in HT_VID, for a total of ≈ 18 h. Table 12 gives the results of the top four 2TA configurations [9] on the four videos. In each of the top 4 configurations, tier 1 used MOG2 [26] and tier 2 used the trained ConvNets VGG16, ResNet32, and ConvNetGS4.

Table 12. Performance of 2TA on 4 videos: columns VGG16, ResNet32, ConvNetGS4 give the bee motion counts returned by configurations MOG2/VGG16, MOG2/ResNet32, MOG2/ConvNetGS4, respectively; last column gives human bee motion counts for each video; counts closest to human counts in rightmost column are in bold.

Video	VGG16	ResNet32	ConvNetGS4	Human Count
NT_VID	151	75	127	73
LT_VID	47	25	43	353
MT_VID	1245	145	316	2924
HT_VID	16,647	13,362	15,109	5738

In 2021, we created another video dataset, which we called BEE_VID2, for our continuing investigation of PIV principles in the analysis of bee traffic. This dataset includes the four videos from BEE_VID1 and 28 new 30-s videos from two BeePi monitors deployed in an apiary in Logan, UT from May to November 2018. The new videos had a resolution of 1920×1080 pixels and a frame rate of ≈ 25 frames per second.

We used BEE_VID2 to design and evaluate BeePIV, a video-based algorithm to measure both omnidirectional and directional honeybee traffic [11]. In BeePIV, frames from bee traffic videos are converted to particle motion frames with uniform white background and multiple motion points generated by a single bee are clustered into a single particle. PIV is subsequently applied to the particle motion frames to compute particle displacement

vectors that are classified as incoming, outgoing, and lateral. The respective vector counts are used as measures of incoming, outgoing, and lateral bee traffic. We are currently using BeePIV to compute bee motion curves for different hives to verify our hypothesis that the incoming and outgoing bee traffic patterns closely follow each other. Our preliminary experiments (See Figure 9) indicate that this hypothesis may be valid. We plan to continue our work on improving the accuracy of BeePIV by integrating various image pre-processing techniques (e.g. [34]).

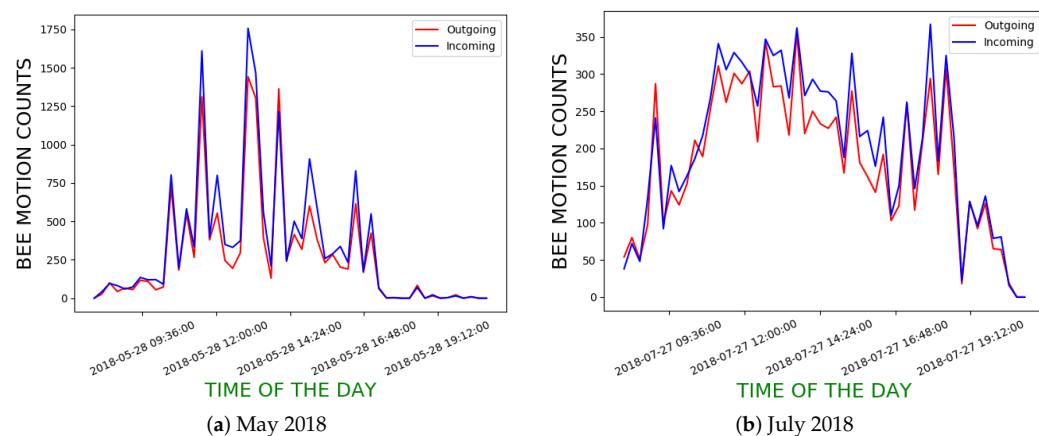


Figure 9. Incoming and outgoing bee motion count curves for two hives in Logan, UT.

5. Weather Datasets

There have been numerous investigations of correlating honeybee behavior with weather (e.g., [12,35–37]). In 2020, we started investigating possible correlations between audio and video bee traffic features and weather [38]. We curated our first dataset, which we called BEEPI_WEATHER1, to investigate possible correlations between bee audio and traffic patterns and different weather variables. To create BEEPI_WEATHER1, we used the publicly available data from the Utah Climate Center (UCC) [39]. The UCC has a weather station on the Utah State University (USU) campus in Logan, UT, which is located ≈ 3 km east of the apiary in Logan, UT where 4–5 BeePi monitors have been regularly deployed since 2018. This weather station collects weather data every hour for educational and research purposes. The station measures 43 different weather and climate variables [40], of which we chose 21 variables such as relative humidity, evapotranspiration, solar radiation, precipitation, air temperature, wind speed, etc. We took the measurements of these variables from March 2018 to July 2019 and aligned them with the video bee traffic measurements obtained with BeePIV [11] from the videos recorded by deployed BeePi monitors.

Our preliminary experiments (See Figures 10 and 11) indicate that there may be a negative correlation between the concentration CO_2 in the air and forager traffic as measured by the omnidirectional bee motion counts computed by BeePIV. Each of the four graphs show that as CO_2 concentration decreases, forager traffic increases and that smaller changes in CO_2 concentration appears to have no impact forager traffic. The correlation value between CO_2 concentration and bee motion traffic for the majority of days in June and July 2018 is ≈ -0.60 .

We also investigated the impact of net radiation on forager traffic. Net radiation is the balance between the amount of incoming solar radiation absorbed by the Earth's surface and the amount of radiation reflected back from the Earth [41]. Net radiation estimates the total energy available at the Earth's surface. Different places on the surface of the Earth absorb different amounts of solar radiation.

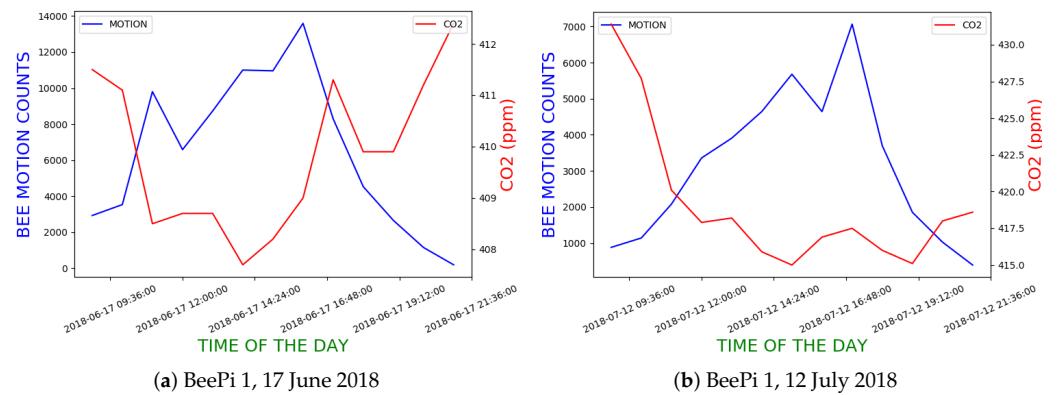


Figure 10. Alignment of CO₂ measurements from USU UCC weather station with BeePIV motion counts from videos from BeePi monitor 1 in Logan, UT.

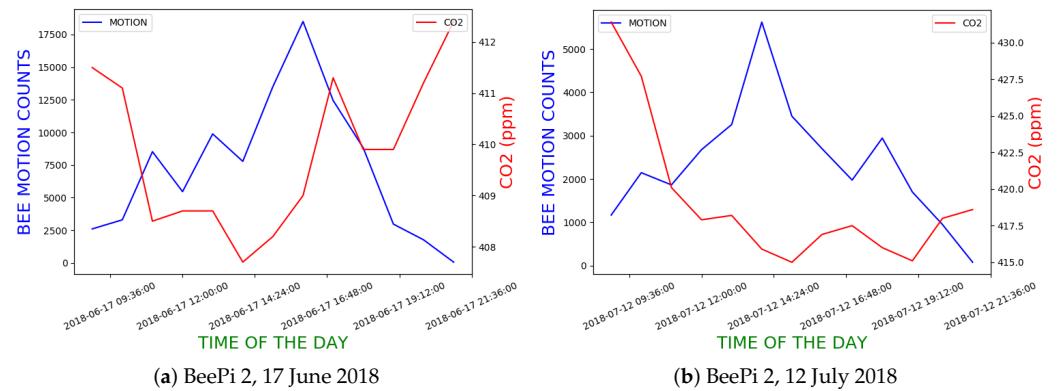


Figure 11. Alignment of CO₂ measurements from USU UCC weather station with BeePIV motion counts from videos from BeePi monitor 2 in Logan, UT.

While we observed some positive correlation (0.60 and above) between net radiation and forager traffic on several individual days (See Figure 12) the distribution of correlation values for the entire months of June and July 2018 had a median value of 0.40, which indicates that there may not be a strong correlation between net radiation and forager traffic in these two months.

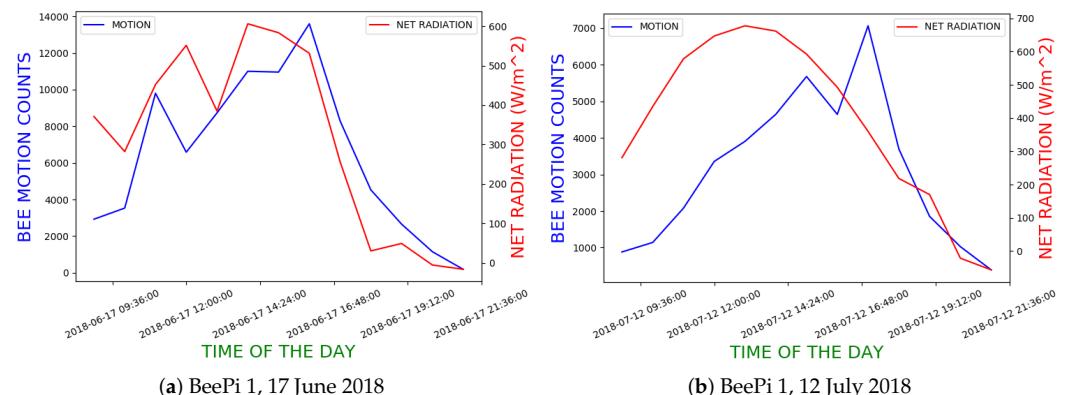


Figure 12. Alignment of net radiation measurements from USU UCC weather station with BeePIV motion counts from videos from BeePi monitor 1 in Logan, UT.

While curating BEEPI_WEATHER1, we had many informal discussions with several researchers at the UCC, which convinced us that we should build our own weather station to monitor the local weather conditions at each apiary. Such weather variables as wind

speed, net radiation, and CO_2 concentration may vary from neighborhood to neighborhood. In 2020, we designed and built the first version BeePiW, a multi-sensor weather station and deployed it at a private apiary in Logan, UT. BeePiW has seven sensors: (1) a temperature sensor to measure ambient temperature; (2) a barometer to measure atmospheric pressure (3) a humidity sensor to measure relative humidity; (4) an anemometer to measure wind speed and direction; (5) a rain sensor to measure rainfall; (6) a pyranometer to measure solar irradiance; (7) an electromagnetic field sensor to measure electro-magnetic frequencies, radio frequencies, and electric fields. All sensors are connected to a raspberry pi computer that controls data acquisition. The data are saved on a USB storage device connected to the pi computer. We are currently curating a new weather dataset, which we called BEEPI_WEATHER2, which aligns the weather data collected by our BeePiW weather station in 2020 and 2021 with omnidirectional and directional forager traffic obtained with BeePIV. We plan to release BEEPI_WEATHER2 and document our experiments and findings with this dataset in a future publication.

6. Summary

In this article, we provided a comprehensive point of reference to the datasets we have so far curated and used in the BeePi project. We hope that our datasets will provide benchmarks for continuous EBM and help interested parties verify our findings, discover and correct errors, and advance the state of the art in EBM and related areas of AI, ML, and data science.

The datasets BEE1 [42], BEE2_1S [43], BUZZ1 [44], and BUZZ2 [44] can be downloaded directly from the links given in the references. The video dataset BEE_VID1 is available in the supplementary materials to our first article on the application of PIV to the analysis of honeybee traffic [10].

We currently lack sufficient resources to host all our datasets online and encourage interested parties to make email arrangements with the author if they want to obtain BEE3, BEE4, BEE_VID2, BUZZ3, BUZZ4, and BEEPI_WEATHER1. We are working on curating another weather dataset, which we called BEEPI_WEATHER2, to couple omnidirectional and directional bee traffic to the weather data collected with our BeePiW stations in 2020 and 2021.

Our future research will focus on curating more image and audio datasets to improve the accuracy of BeePIV in measuring incoming, outgoing, and lateral forager traffic and to investigate correlations and alignments between weather and video-based honeybee traffic. As opportunity arises, we will integrate audio features to correlate them with bee traffic and weather.

Funding: This research is funded, in part, by two Kickstarter research fundraisers in 2017 [1] and 2019 [2].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The author would like to thank all Kickstarter backers of the BeePi project and especially the Angel Backers (in alphabetical order): Prakhar Amlathe, Ashwani Chahal, Trevor Landeen, Felipe Queiroz, Dinis Quelhas, Tharun Tej Tammineni, and Tanvir Zaman. The author expresses his gratitude to Gregg Lind, who backed both fundraisers and donated hardware to the BeePi project. The author is grateful to Richard Wagstaff, Craig Huntzinger, and Richard Mueller for letting him use their private property in northern Utah for longitudinal EBM tests. The author would like to thank his graduate and undergraduate students Sarbjit Mukherjee, Kristoffer Price, Anastasiia Tkachenko, Daniel Hornberger, Nikhil Ganta, Matthew Lister, Aditya Bhuraskar, Laasya Alavalapati, Chelsi Gupta, Prakhar Amlathe, Astha Tiwari, Keval Shah, Myles Putnam, Matthew Ward, Sai Kiran Reka, and Sarat Kiran Andhavarapu for their pro bono software, hardware, and data curation contributions to the BeePi project.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EBM	Electronic beehive monitoring
RFID	Radio frequency identification
ML	Machine learning
ConvNet	Convolutional neural network
RF	Random forest
LR	Linear regression
KNN	K-nearest neighbors
MFCC	Mel frequency cepstral coefficient
DL	Deep learning
SVM	Support vector machine
RFE	Recursive feature elimination
SFS	Sequential feature selection
PIV	Particle image velocimetry

References

1. Kulyukin, V. BeePi: A Multisensor Electronic Beehive Monitor. Available online: <https://www.kickstarter.com/projects/970162847/beepi-a-multisensor-electronic-beehive-monitor> (accessed on 11 May 2021).
2. Kulyukin, V. BeePi: Honeybees Meet AI: Stage 2. Available online: <https://www.kickstarter.com/projects/beephoneybeesmeetai/beepi-honeybees-meet-ai-stage-2> (accessed on 11 May 2021).
3. Kulyukin, V.; Putnam, M.; Reka, S. Digitizing buzzing signals into A440 piano note sequences and estimating forager traffic levels from images in solar-powered, electronic beehive monitoring. In *Lecture Notes in Engineering and Computer Science, Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 16–18 March 2016*; Newswood Limited: Hong Kong, China, 2016; Volume 1, pp. 82–87.
4. Beehive. Available online: <https://en.wikipedia.org/wiki/Beehive> (accessed on 11 May 2021).
5. Langstroth Beehive. Available online: https://en.wikipedia.org/wiki/Langstroth_hive (accessed on 11 May 2021).
6. Apis Mellifera. Available online: https://en.wikipedia.org/wiki/Western_honey_bee (accessed on 11 May 2021).
7. Kulyukin, V.; Reka, S. Toward sustainable electronic beehive monitoring: Algorithms for omnidirectional bee counting from images and harmonic analysis of buzzing signals. *Eng. Lett.* **2016**, *24*, 317–327.
8. Kulyukin, V.; Mukherjee, S.; Amlathe, P. Toward audio beehive monitoring: Deep learning vs. standard machine learning in classifying beehive audio samples. *Appl. Sci.* **2018**, *8*, 1573. [[CrossRef](#)]
9. Kulyukin, V.; Mukherjee, S. On video analysis of omnidirectional bee traffic: Counting bee motions with motion detection and image classification. *Appl. Sci.* **2019**, *9*, 3743. [[CrossRef](#)]
10. Mukherjee, S.; Kulyukin, V. Application of digital particle image velocimetry to insect motion: Measurement of incoming, outgoing, and lateral honeybee traffic. *Appl. Sci.* **2020**, *10*, 2042. [[CrossRef](#)]
11. Kulyukin, V.; Mukherjee, S.; Minichiello, A.; Truscott, T. BeePIV: A method to measure apis mellifera traffic with particle image velocimetry in videos. *Appl. Sci.* **2021**, *11*, 2276. [[CrossRef](#)]
12. He, X.J.; Tian, L.-Q.; Wu, X.-B.; Zeng, Z.-J. Rfid monitoring indicates honeybees work harder before a rainy day. *Insect Sci.* **2016**, *23*, 157–159. [[CrossRef](#)] [[PubMed](#)]
13. De Souza, P.; Marendy, P.; Barbosa, K.; Budi, S.; Hirsch, P.; Nikolic, N.; Gunthorpe, T.; Pessin, G.; Davie, A. Low-cost electronic tagging system for bee monitoring. *Sensors* **2018**, *18*, 2124. [[CrossRef](#)] [[PubMed](#)]
14. Shah, K. Power Analysis of Continuous Data Capture in BeePi, a Solar-Powered Multi-Sensor Electronic Beehive Monitoring System for Powered Multi-Sensor Electronic Beehive Monitoring System for Langstroth Beehives. Master’s Thesis, Utah State University, Logan, UT, USA, 2017.
15. Librosa. Available online: <https://librosa.github.io/librosa/> (accessed on 11 May 2021).
16. Gupta, C. Feature Selection and Analysis for Standard Machine Learning Classification of Audio Beehive Samples. Master’s Thesis, Utah State University, Logan, UT, USA, 2019.
17. Scikit-Learn Feature Selection. Available online: <https://scikit-learn.org/stable/> (accessed on 14 April 2021).
18. Sequential Feature Selector. Available online: http://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/ (accessed on 14 April 2021).
19. Kira, K.; Rendell, L.A. The feature selection problem: Traditional methods and a new algorithm. In Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI’92), San Jose, CA, USA, 12–16 July 1992 ; AAAI Press, San Jose, CA, USA, 1992; pp. 129–134.
20. Uddin, M.T.; Uddin, M.A. A Guided random forest based feature selection approach for activity recognition. In Proceedings of the International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Jahangirnagar University, Dhaka, Bangladesh, 21–23 May 2015.

21. Giannakopoulos, T. pyAudioAnalysis: An open-source python library for audio signal analysis. *PLoS ONE* **2015**, *10*, e0144610. [[CrossRef](#)]
22. Bhouraskar, A. Automation of Feature Selection and Generation of Optimal Feature Subsets for Beehive Audio Sample Classification. Master’s Thesis, Utah State University, Logan, UT, USA, 2019.
23. Scikit-Learn Recursive Feature Selection. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html (accessed on 11 May 2021).
24. Zivkovic, Z. Improved adaptive gaussian mixture model for background subtraction. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Cambridge, UK, 26–26 August 2004; Volume 2, pp. 28–31.
25. KaewTraKulPong, P.; Bowden, R. An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In *Video-Based Surveillance Systems*; Remagnino, P., Jones, G.A., Paragios, N., Regazzoni, C.S., Eds.; Springer: Boston, MA, USA, 2002; doi:10.1007/978-1-4615-0913-4_11. [[CrossRef](#)]
26. Zivkovic, Z.; van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.* **2006**, *27*, 773–780. [[CrossRef](#)]
27. Tiwari, A. A Deep Learning Approach to Recognizing Bees in Video Analysis of Bee Traffic. Master’s Thesis, Utah State University, Logan, UT, USA, 2018.
28. TFlearn. Available online: <https://github.com/tflearn/tflearn> (accessed on 11 May 2021).
29. Vats, P. Design and Evaluation of Convolutional Networks for Video Analysis of Bee Traffic. Master’s Thesis, Utah State University, Logan, UT, USA, 2019.
30. Alavalapati, L. Bee Shadow Recognition in Video Analysis of Ominidirectional Bee Traffic. Master’s Thesis, Utah State University, Logan, UT, USA, 2019.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27 June–30 June 2016.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Versaci, M.; Morabito, F.C. Image edge detection: A new approach based on fuzzy entropy and fuzzy divergence. *Int. J. Fuzzy Syst.* **2021**. [[CrossRef](#)]
35. Clarke, D.; Robert, D. Predictive modelling of honey bee foraging activity using local weather conditions. *Apidologie* **2020**, *49*, 386–396. [[CrossRef](#)]
36. Burrill, R.M.; Dietz, A. The response of honey bees to variations in solar radiation and temperature. *Apidologie* **1981**, *12*, 319–328.:19810402. [[CrossRef](#)]
37. Vicens, N.; Bosch, J. Weather-dependent pollinator activity in an apple orchard, with special reference to osmia cornuta and apis mellifera (hymenoptera: Megachilidae and apidae). *Environ. Entomol.* **2000**, *29*, 413–420. [[CrossRef](#)]
38. Mukherjee, S. Acquisition, Processing, and Analysis of Video, Audio and Meteorological Data in Multi-Sensor Electronic Beehive Meteorological Data in Multi-Sensor Electronic Beehive Monitoring. Ph.D. Thesis, Utah State University, Logan, UT, USA, 2020.
39. Utah Climate Center. Available online: <https://climate.usu.edu> (accessed on 11 May 2021).
40. USU Weather Station. Available online: <https://climate.usu.edu/mchd/dashboard/overview/USUwx.php> (accessed on 11 May 2021).
41. Seemann, J. *Net Radiation*; Springer: Berlin, Germany, 1979.
42. BEE1 Image Dataset. Available online: <https://usu.app.box.com/s/0a5yurmn7ija15cp236awa1re65mbcnr> (accessed on 11 May 2021).
43. BEE2_1S Image Dataset. Available online: <https://usu.box.com/s/p7y8v95ot9u9jvbbayci61no3lzbgx3> (accessed on 11 May 2021).
44. BUZZ1 and BUZZ2 Datasets. Available online: <https://usu.app.box.com/v/BeePiAudioData> (accessed on 11 May 2021).