ElephNet: A Large-Scale Audio Dataset with Applications to Low-Cost Wildlife Conservation

Abstract

Passive acoustic monitoring — a cost-effective alternative to traditional conservation efforts — relies on the ability to automatically detect animal vocalizations from audio recordings. To stimulate the development of robust detection algorithms, we introduce "ElephNet," a new large-scale database which consists of 4000 hours of forest audio recordings with 150,000 labeled calls. The difficulties that arise in this dataset — namely, adapting to unknown noises in new environments and surpassing human labeling error — are fundamental concerns found in many other application domains. We introduce three challenge tasks as well as a set of baseline algorithms utilizing convolutional neural networks. Our baseline makes use of a novel technique to stitch together partially observed predictions. With this dataset, we not only aim to aid conservation efforts, but to stimulate the development of new machine learning techniques.

1. Introduction

Remote sensing has the ability to provide vast quantities of data for a variety of applications where traditional methods are prohibitively expensive, prone to bias, or of limited scope. One very promising application is *passive acoustic monitoring* of endangered species. Rather than relying on infrequent field work to monitor the size of animal populations, autonomous recording systems can provide large quantities of continuous data from multiple recording locations. Microphones are able to detect the presence of animals at greater distances than possible by line of sight. From acoustic data, researchers can monitor animal activity, estimate population size, and receive warning of poaching activities — all without disturbing the animals.

Several challenging issues emerge in developing automated methods for detecting animal calls in the wild: namely, the ability to adapt to unknown conditions and noisy training data. In an isolated environment with little noise, simple algorithms may be sufficient to identify animal calls. However, the goal of conservation efforts is to deploy microphones throughout hundreds of regions. These efforts will be made possible only by algorithms that do not require labeling of location-specific data. In addition, the sparsity of animal activity coupled with background noise makes it difficult even for experienced humans to accurately identify animal calls. As a result, data that could be used for training will often contain some mislabeled information. These issues of adaptability and human labeling error are concrete concerns that are relevant to machine learning applications in many other domains.

Figure 1: Elephants gathering at the Dzanga forest clearing.

Perhaps the biggest hurdle facing passive acoustic monitoring is a shortage of publicly available bioacoustic data. Some previous datasets of note are the *Singing Insects of North America* database[1] (consisting of roughly 200 insect recordings of various species) and the *Warbler Flight Calls* database[2] (consisting of 200 bird recordings). In addition, several hours of labeled bird call and whale sound data are available from previous bioacoustics competitions.[3] However, most prior research has utilized small privately held datasets, which delays progress in this important research area.

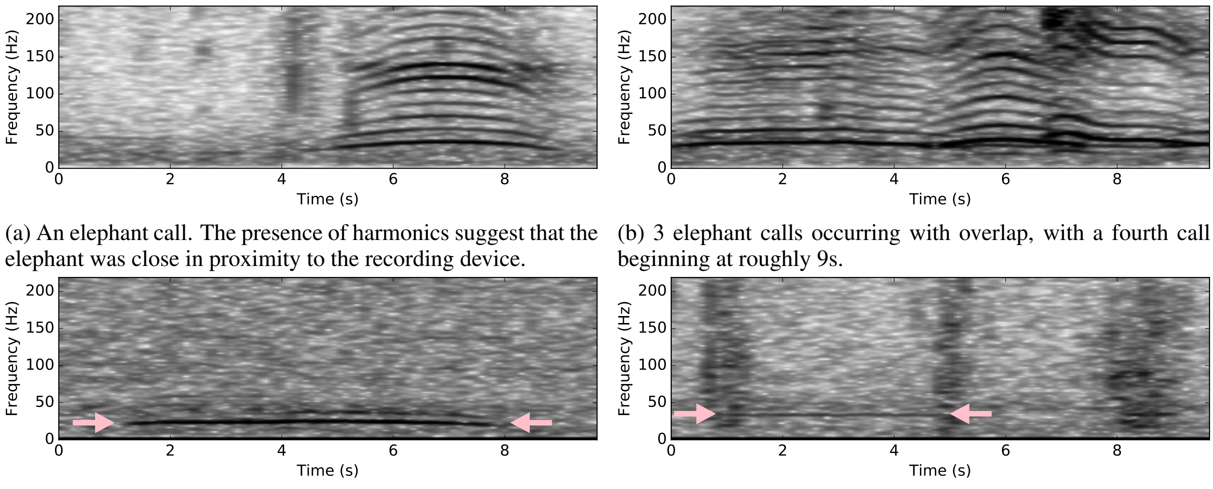
Public large scale datasets encourage advancement not only for a particular application domain but also for machine learning as a whole. Datasets such as ImageNet (Deng et al., 2009), MNIST (LeCun et al., 1998), PASCAL (Everingham et al., 2010), and TIMIT (Garofolo et al., 1993)

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1] http://entnemdept.ifas.ufl.edu/walker/buzz/

[2] http://computational-sustainability.cis.cornell.edu/projects/index.php

[3] http://sabiod.univ-tln.fr/icml2013/challenge_description.html



(a) An elephant call. The presence of harmonics suggest that the elephant was close in proximity to the recording device.

(b) 3 elephant calls occurring with overlap, with a fourth call beginning at roughly 9s.

(c) An elephant call with only 1 visible harmonic. It is harder to differentiate such calls from other low-frequency noises.

(d) A very faint elephant call, sandwiched by wind gusts. Calls such as are occasionally missed by human labelers.

Figure 2: Example elephant call spectrograms.

have led to seminal advances in machine learning architectures and methodologies. To that end, we provide the following contributions:

1. **We introduce "ElephNet," a large-scale dataset** to promote automated acoustic monitoring of forest elephants. ElephNet is a rich dataset consisting of forest audio in which all elephant vocalizations have been labeled. In total, **150,000 elephant calls** are labeled in **4000 hours of data**, which — to the best of our knowledge — makes ElephNet the largest publicly available bioacoustics dataset. This dataset features several challenging issues — ranging from high noise levels, adaptation to new environments, and human labeling error. We hope that the quantity and richness of this dataset will accelerate progress toward better bioacoustics models, while leading to models that benefit machine learning as a whole.

2. **We define three challenge tasks** for the dataset: binary *classification* of audio segments (into "elephant call " and "no elephant call"), *segmentation* of an audio segment (what regions of an audio file contain elephant calls), and *detection* of individual elephant calls. To ensure that algorithms are able to transfer to new locations, we withhold one location from the training data for testing and validation. Furthermore, while all datasets contain expert-labeled recordings, a majority of the training data is labeled by volunteers. To achieve expert-level performance, algorithms will need to account for volunteer labeler error.

3. **We provide baseline results for each of the challenge tasks** by using a convolutional neural network model (convnet). The networks view time windows of the audio in the spectral domain. Our baseline achieves surprisingly high precision and recall on the segmentation task. To obtain good performance on the detection task, **we introduce a novel method to stitch together fractional predictions**, which enables detection of long calls exceeding the receptive field of the convnet. This allows the network to be robust to large variations in scale and to distinguish multiple overlapping calls.

2. Description of ElephNet

In this section we describe the dataset, its collection methods, and characteristics of the data.

Most elephant vocalizations are low rumbles which are often below the range of human hearing. While speeding up a recording can bring the call into an audible range, it is often easier to detect calls in the spectral domain. Figure 2a shows a spectrogram of a very clear elephant call, taken from a forest audio recording. Distinguishing characteristics of this call are the low fundamental frequency, the slight pitch bend (no more than 10 Hz), and the duration (between 2 and 10 seconds). In crowded forest regions, multiple elephants may call simultaneously. While it is rare for two calls to arrive at the microphone at exactly the same time, it is common for an elephant to begin a call before another elephant finishes. For example, in Figure 2b, three overlapping calls occur in the 10 second window.

In noisy environments, it is not guaranteed that these harmonic frequencies will be captured by recording devices. When an elephant is far away from a microphone, it is

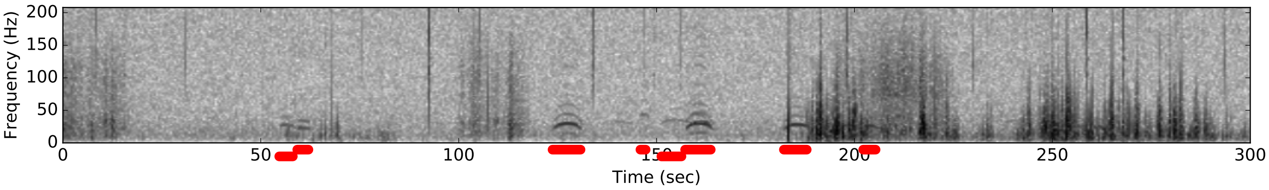


Figure 3: 5 min spectrogram clip with above-average elephant call activity. Red underlines indicate when a call occurs. This clip features strong wind gusts starting at roughly 180 s.

Location	Dates Collected	Labeled Hours	Labeled Calls	Calls/Hour	Percentage of Audio Containing Calls
Ceb1	04/2009 - 06/2011	1907	52074	27.31	3.74%
Ceb4	04/2008 - 03/2011	1280	23038	18.00	1.97%
Dzanga	04/2011 - 02/2012	312	63792	204.46	19.58%
Jobo	05/2009 - 06/2011	1286	24392	18.97	1.98%

Table 1: Data statistics by collection site

more likely that only the lowest frequencies will be captured (Figure 2c). Without the higher harmonics, it can be difficult for a human labeler to differentiate between elephant calls and other noises, such as cars, falling branches, or other animal vocalizations. Figure 2d shows a very faint call. Even an experienced human labeler may have difficulty accurately classifying this noise.

Figure 3 is a 5 minute spectrogram clip with above average elephant call activity. At roughly 180 s, loud wind gusts cloud much of the low frequency range. Such noises are common throughout the recordings.

2.1. Data Collection

Data were collected in 4 different forest regions throughout Central Africa. In each of the regions — denoted Ceb1, Ceb4, Dzanga, and Jobo — a single recording device was placed 7–10 m off the ground in a tree near a forest clearing. Forest elephants tend to gather at these clearings for multiple purposes. Nevertheless, elephant calls occur somewhat infrequently in these locations. Only 3.8 percent of the total collected audio contains elephant call activity. Each of the sites features a different amount of elephant call activity, with averages given in Table 1. For example, elephant calls occur *20 times* as frequently in Dzanga as they do in Ceb4 or Jobo. It is important for automated detection methods to be robust to this variable density.

Each site features different types of background noise, both natural and man-made. For example, Ceb4 is located next to a road, so the recordings occasionally feature vehicles and illegal logging activity. Heavy wind and thunder are common near Jobo. Rain, insect chirps, falling trees, and gunshots will occasionally appear in all recordings.

2.2. Labeling

A team of volunteers and experts constructed truth sets for each of the four locations. Most calls were identified through visual examination of the spectrogram. Labelers would listen to pitch-boosted audio recordings only for sounds that were difficult to classify. Sounds that remained ambiguous to volunteers were often validated by expert labelers.

We measured volunteer accuracy by comparing two or more labelers on several recordings. Depending on the recording, the number of calls volunteers agreed upon ranged from 81 to 87 percent of the total labels. We therefore estimate the accuracy of volunteer labels to be roughly 84%. Volunteers tended to differ on labels of ambiguous faint sounds. It is therefore difficult to determine whether these labeling inconsistencies are false positives (erroneous labels) or false negative (missed calls). Expert labelers, on the other hand, agreed on 94 percent of calls.

3. Challenges and Evaluation Metrics

There are many ways to quantify what constitutes as “detecting activity” for passive acoustic monitoring purposes. For some use cases, it may be helpful to know whether any activity occurred over a particular time scale. In other cases, it may be useful to know the exact times when calls occurred for further analysis. To cover all possible use cases, we define three tasks: 1. **Classification of audio sample:** Given an audio clip of length n, return a binary label indicating whether elephant call activity is present or not. For this task, n is set to 20 s. 2. **Segmentation of audio sample:** Given an audio clip of length n, return a sequence of time intervals indicating when elephant call activity oc-

Dataset	Locations	Labeled Hours	Labeled Calls	Label Accuracy
Train-volunteer	Dzanga, Ceb1, Ceb4	2568	123810	84%
Train-expert	Ceb1	931	15094	94%
Validation-volunteer	Jobo	166	4128	84%
Validation-expert	Jobo	147	3246	94%
Test	Jobo	147	3381	94%

Table 2: Training, validation, and test datasets.

curs. This *does not* require the identification of individual calls. For this task, n is set to $5\,\mathrm{min}$. Start and end times are rounded to the nearest $1/10^{th}$ of a second. 3. **Detection of individual calls:** Given an audio clip of length n, return a sequence of time intervals indicating when *individual elephant calls* occur. This task uses the same time parameters as the segmentation task.

For all tasks, we define a label marking elephant call activity as a *positive* label. Conversely, a *negative* label indicates no activity. Each task uses the same training, validation, and test sets (Table 2). The training dataset consists of recordings from Dzanga, Ceb1, and Ceb4. The majority of data ("train-volunteer") was labeled by volunteers ($84\,\%$ accuracy). A subset of this data ("train-expert") was labeled by experts ($94\,\%$ accuracy).

To test transfer to unseen environments, the validation and test datasets consist of recordings from Jobo — a location withheld from the training set. The test set only consists of recordings labeled by experts. The validation set consists of both expert labels ("validation-expert") and volunteer labels ("validation- volunteer"). This makes it possible to predict how susceptible a model is to labeling error.

3.1. Classification

The purpose of this task is to produce a binary classifier that examines audio segments on a coarse scale. A classifier for this task could provide conservation efforts with rough statistics of when and where elephant activity occurs. This task is designed to be an introduction level task to familiarize researchers with the dataset. Any sample that contains at least two seconds of an elephant call is considered positive. Negative samples are ones which contain no elephant calls. In all of the datasets, we only include samples that are either positive or negative — i.e. no samples with between 0 and 2 seconds of elephant call activity. Classifiers are judged by the F1-measure.

3.2. Segmentation

The purpose of the segmentation task is to report exactly at what times elephant call activity occurs. Such information would allow elephant conservation efforts to analyze elephant calls without having to search through large amounts of negative data. For every 5 minute clip, the task is to return a positive or negative label at each 0.1 second interval. The quality of a label is determined by how many intervals correspond with the ground-truth label. Most ground truth labels correctly bound calls within an error 0.2 seconds. To account for this, we add a buffer region of $0.2\,\mathrm{s}$ to either side of a call label. We use the F1-measure as the performance metric, where precision and recall are calculated from every $0.1\,\mathrm{s}$ interval.

3.3. Detection

Building population models requires an accurate count of how many calls occur. Because some calls may occur simultaneously, it is necessary to identify individual calls from a segment of elephant activity. In this task, we ask for how many calls occur in a 5 minute time slice. In addition, we ask for the time each call begins and ends. This allows researchers to verify and further examine all detected calls.

The scoring metric for this task is based off computer vision object detection metrics from ILSVRC (Deng et al., 2009) and PASCAL VOC (Everingham et al., 2010). A detector returns a list of start and end times for all detected calls. We will refer to these intervals as the *bounding boxes* of calls. Let $t = (t_{start}, t_{end})$ be the bounding box of a particular call. We consider t to correctly match a ground truth label $t^* = (t^*_{start}, t^*_{end})$ if the intersection of the two time intervals over their union (IoU) is greater than 0.6. For testing, detected calls are matched with ground truth calls based on IoU. Only one detection may be matched with any ground truth call, and vice versa. Only successfully matched calls are considered to be true positives. The F1-score is used to measure success, where precision and recall are calculated from the matched and unmatched calls.

4. Baseline

Bioacoustics has received some interest from the machine learning community for a variety of species. Researchers have had success identifying bird flight calls (Bardeli et al., 2010; Damoulas et al., 2010; Dufour et al., 2013; Ross & Allen, 2014), cicada chirps (Potamitis et al., 2007), and whale sounds (Jarvis et al., 2008; Baumgartner & Mussoline, 2011; Pourhomayoun et al., 2013; Shamir et al., 2014). Unlike elephant calls, the vocalizations of these species tend to occupy a unique frequency band, and are therefore models are less susceptible to environmental noise. Therefore, the algorithms proposed by these works do not translate directly to good performance at elephant call detection. Furthermore, much of this work has focused on detection in controlled settings, and therefore it is unlikely that these approaches transfer to new locations.

For each of the benchmark tasks defined in Section 3, we provide baseline results using a convolutional neural network (convnet) model. Convnets have been shown to be capable of identifying objects across different scales and to be robust to varying background conditions and noise (Russakovsky et al., 2014). For these reasons convnets may be well suited to learning the rich spectral characteristics of elephant calls of varying length, even in the presence of realistically high levels of background noise. The baseline models proposed take as input raw audio spectrograms without filtering or enhancement and produce outputs specific to each of the three tasks (discussed below). For training, we used both the train and train-expert datasets. Due to time constraints, we did not train models that differentiated between volunteer and expert labels. All models were implemented in Torch (Collobert et al., 2011).

4.1. Classification

Each audio file is converted into a spectrogram, using 512 fast-Fourier transform bins and 90 percent overlap. This produces 257 frequency bins between 0 and 500 Hz, with samples every 0.05 s. We examine only the first 112 frequency bins, providing a frequency range between 0–218 Hz. This frequency resolution and range is very similar to what human experts use to hand-label calls.

The spectrograms serve as input to the convnet. Previous uses of convnets for audio data perform convolution only on the frequency domain, relying on hidden markov models or memory networks for temporal aggregation (Abdel-Hamid et al., 2013; Deng et al., 2013). However, because of the visual characteristics inherent to the spectral representations of calls, we opt for convolution over both frequency and time. This allows a classifier to learn rich features that combine frequency and temporal information.

4.1.1. Receptive Field Size

The duration of an elephant call can be as short as two seconds and as long as ten seconds. The window size of the convnet should be sufficiently large to capture temporal context of a call. With a short time window (i.e. 2 seconds), there may not be enough information to differentiate between elephant calls and other noises in the same frequency range, such as cars or snapping tree branches. On the other hand, if the time window is made to completely encompass all elephant calls (i.e. greater than 10 s), the network may discard short calls as background noise. As a compromise, we use a 5.6 s sliding window, or 112 spectrogram frames. Approximately 80 percent of calls are between 2 and 5.6 s in length; thus, the sliding window will be able to contain the entirety of most calls. In addition, because no calls are shorter than 2 s, every call will occupy at least 35 percent of the sliding window.

4.1.2. Training

For training, the models are supplied spectrogram clips equal to the window size of the convnet. Because of the large class imbalance (3% elephant call), the network is very difficult to train when data are uniformly sampled, even when using a weighted loss function. To account for this, we upsample positive examples by including several clips containing different segments of each call. In addition, we subsample the clips that contain no elephant calls to create an equal balance of positive and negative data.

We train the network against a weighted cross-entropy loss function. Negative sample loss is weighted by a factor $w \geq 1$, while positive sample loss is unweighted. We found that this approach reduces the number of false positives due to subsampling. The network is trained for 25 epochs over the course of two days using stochastic gradient descent.

4.1.3. Network Implementation

We implement the classifier as a convolutional network based on the VGG-16 image classification network from Simonyan and Zisserman (2014). There are five 64 channel convolutional layers used for feature extraction, each with 3-by-3 filters. Max-pooling occurs after the first four layers. The receptive field size of each feature is 78-by-78, or 144 Hz-by-3.9 s. The features are then fed into a 7-by-7 convolution layer, and then finally into a 1-by-1 convolution layer with two filters. The final result is an output of size 2 every 0.8 s, with a receptive field size of 5.6 s — representing whether or not the receptive field contains elephant call activity. If an elephant call is detected in any output frame, then the sample is labeled as positive.

4.1.4. Results and Discussion

Figure 5 displays the precision/recall curve for the classification model. When applied to the test dataset — which is from a location completely withheld from the training data — it achieves a F1-score of $\mathbf{0.91}$, with a precision of $\mathbf{0.91}$ and a recall of $\mathbf{0.90}$. The deep network architecture is able to overcome the extreme class imbalance (3% of audio contains elephant calls) and varied background noises. Additionally, the representation learned by the convnet is

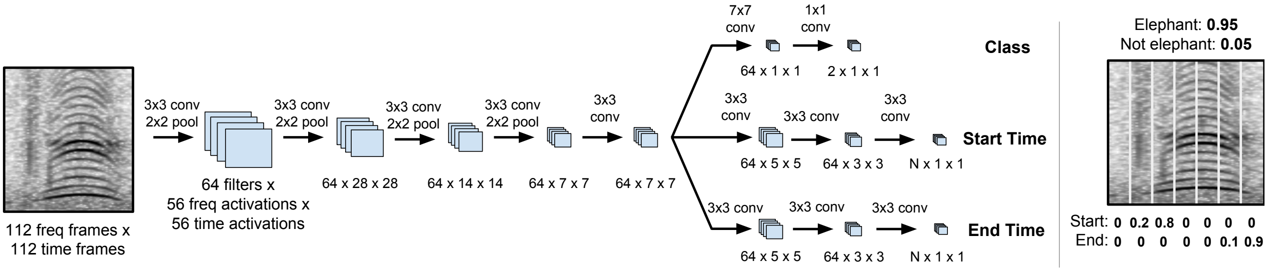


Figure 4: Left: Network diagram for segmentation task. Right: Class, start time and end time prediction for example input.

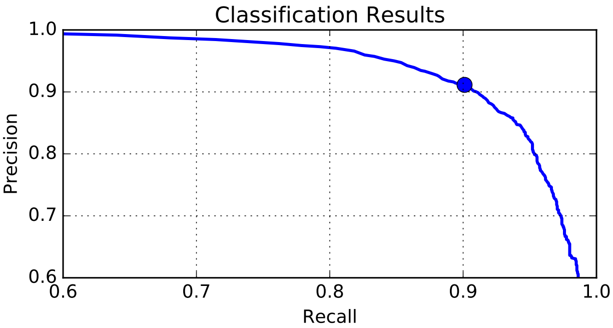


Figure 5: Precision/recall for classification task. Marker note maximum F1-score.

sufficiently general to be applied to a new unseen location. It is fair to attribute the success of the convnet architecture to the large quantity of labeled data.

4.2. Segmentation

The convnet architecture from the classification task can naturally be extended for the segmentation task. A trivial extension would be to use the classification network as is for the entire 5 minute sample, using the output at each pooled timestep for segmentation. However, the prediction at each time step would be based a receptive field of 5.6 s, which could lead to erroneous labels in regions of transition into elephant calls. To increase the resolution of predicted regions, we modify the classification architecture to predict the beginning and end of elephant activity within the convnet window. From the last feature extracting layer, we add two sets of convolution layers in parallel to the final layers for classification (Figure 4). These two new sets of layers are used for predicting the start and end times of elephant call activity in the convnet window. This is an adaptation of the region proposal networks (RPN) of Ren et al. (2015).

4.2.1. Predicting Start and End Times

Rather than predicting the start and end coordinates with a regression, we quantize the convnet window into N time-bins. The output of the start and end networks are softmax layers of size N which predict the bin of the start and end coordinates. While this approach produces coarser time estimates, we have found that it produces much more consistent time estimates than a standard regression.

Similarly to classification, the start and end coordinate networks use a 7-by-7 window of the feature map. The feature map is fed through three 3-by-3 convolution layers (Figure 4). The last of these layers has N filters, which produces a N-by-1-by-1 output for every time window. The receptive field of the output is 174 time frames, or roughly 8.5 seconds. We then divide the middle 5.6 seconds of the receptive field into bins $b_1 \ldots b_n$, assigning each bin to one of the N outputs. Calls that start before the 5.6 second window are assigned to b_1, and calls that end outside the window are assigned to b_n. Because of this, the network does not extrapolate start and end coordinates, but rather predicts with high confidence that the call is not fully contained within the window.

A natural choice for N is 7, which corresponds to the number of time features in the convnet window. However, we have found that N can be increased to higher multiples of 7, which creates bins of finer timescales. Because we are making predictions only in the time dimension, the resolution of predictions scales linearly with the size of N. We have tested models with up to 56 filters, which results in bins of size 0.1 s.

The network produces a time prediction for all windows, even when the classification network detects no activity. From these prediction fragments, we produce T — the predicted set of time frames containing elephant call activity:

$$T = \bigcup_{i \in POS} \left(t^i_{start}, t^i_{end}\right) \qquad (1)$$

where POS is the set of windows that the classifier network labels positive.

4.2.2. Training

Similarly to classification, we create 5.6 s clips of training spectrograms and subsample for an even representation of positive and negative samples from both locations. The shared feature-extraction layers and the classification layers are initialized with the weights from the classification network. In training we use the following loss function, derived from Ren et al. (2015):

$$L = \frac{1}{N}\left[L_{cls}(p, p^*) + p^* \left(L_{start}(s, s^*) + L_{end}(e, e^*)\right)\right] \tag{2}$$

Here, p and p^* are the predicted class and the ground-truth class, respectively. 1 represents elephant call activity, and 0 represents no activity. s and s^* represent the predicted and ground truth start time bins, and e and e^* represent the end time bins. For samples with more than one segment of elephant activity, we assign s^* and e^* to the largest segment present in the window. L_{cls}, L_{start} and L_{end} are all unweighted cross-entropy loss functions. The p^* term in front of L_{start} and L_{end} ensures that the network only trains the time prediction layers on positive samples.

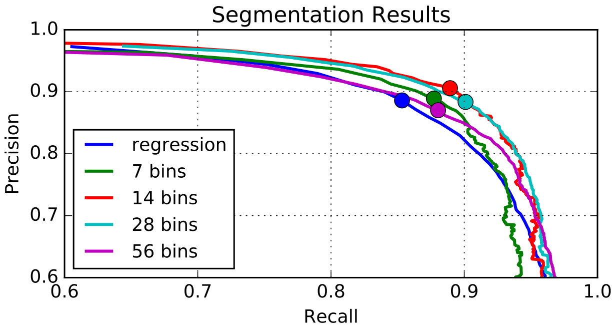


Figure 6: Precision/recall curve for segmentation task.

4.2.3. Results and Discussion

We test the quantized start/end prediction method with different resolutions. In particular we test the model with 7, 14, 28, and 56 bins, resulting in time resolution of 0.8, 0.4, 0.2 and 0.1 s respectively. For comparison, we test against a model where the final softmax layer is replaced with single linear output for continuous regression. In addition, we provide a naive baseline by using a network from the classification task. For this baseline, the segmentation prediction is the union of all receptive fields where activity is detected. In all experiments, shared convolution layers all have 64 filters which are initialized with parameters from the 3x-weight classification experiment.

The precision/recall curves for each network can be seen in Figure 6. We find that the 14 and 28 bin networks have on average the highest precision and recall, outperforming a standard regression. The 28 bin network achieves the highest F1-measure of $\mathbf{0.89}$, with a precision of $\mathbf{0.88}$ and a recall of $\mathbf{0.90}$. While this network does not provide the finest time resolution, the lower number of bins should correspond to higher start and end time accuracy. Additionally, the 0.20 s resolution provided by the 28 bin model lies within the segmentation buffer region, and should therefore be sufficient. All of these models significantly outperform the naive approach (not pictured), which achieves a precision of $\mathbf{0.51}$ and a recall of $\mathbf{0.75}$ at maximum F1-measure.

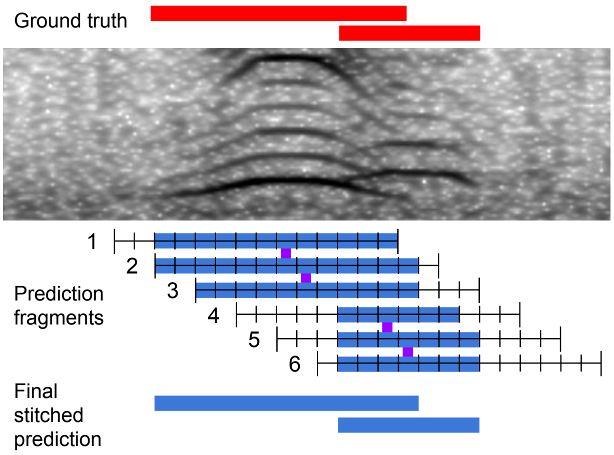


Figure 7: Stitching predictions fragments for detection. Starting in window 4, the second call becomes prominent, and thus the prediction fragment locates the second call. The jump from window 3 to window 4 results in two sets of connected components, corresponding to 2 calls.

Surprisingly, top precision and recall for the segmentation task are much higher than for classification. This is counterintuitive because classification appears to be a simpler task. We postulate that the aggregation of start and end time predictions provide a significant level of refinement to the model. This is justified by the large performance gap between the naive baseline and other approaches.

4.3. Detection

The network for detection is an extension of the segmentation network. The key difference between detection and segmentation is the prediction of individual calls, rather than aggregated elephant call activity. To account for this, s^* and e^* in (2) become the time parameters of the *individual call* most present in the window, rather than the parameters of activity as a whole. We define the “most present” call c^* to be the call most accurately predicted by the window (s_w, e_w), as measured by its intersection over union:

$$c^* = \arg\max_{i \in C} \operatorname{iou}\left((s_i, e_i), (s_w, e_w)\right) \tag{3}$$

where C is the set of all calls intersecting the window.

4.3.1. Stitching Predictions of Long Calls

Unlike the segmentation task, for detection we must identify individual calls from the outputs at all windows. Because calls may be longer than the window size, predicting

Algorithm 1 Stitch Prediction Fragments

```
Input: time prediction fragments $\{s_i, e_i\}$
Output: stitched time predictions $\{s'_i, e'_i\}$
for $i = 1$ to len$(\{s_i, e_i\})$ do
  for $j = i + 1$ to len$(\{s_i, e_i\})$ do
    if $|s_j - s_i| \leq t_{pool}$ and $|e_i - e_j| \leq t_{pool}$ then
      link$(\{s_i, e_i\}, \{s_j, e_j\})$
    end if
  end for
end for
stitched $\leftarrow \{\}$
conn_comp $\leftarrow$ connected_components_of $(\{s_i, e_i\})$
for $i = 1$ to len(conn_comp) do
  if size(conn_comp) $\geq 3$ then
    $s' \leftarrow \min\{s$ for $(s, e)$ in conn_comp$\}$
    $e' \leftarrow \max\{e$ for $(s, e)$ in conn_comp$\}$
    stitched $\leftarrow$ stitched $+ \{(s', e')\}$
  end if
end for
return stitched
```

the start and end of calls requires stitching together prediction fragments from neighboring windows, while still distinguishing individual calls.

To account for this, we introduce a stitching algorithm that accurately assigns prediction fragments to individual calls. Prediction fragments are determined to belong to the same call only if the differences in start and end times are both less than one pooled time step. This criterion is sufficient to distinguish between most overlapping calls. Let c_1 and c_2 be two overlapping calls, and let $\{w_i\}$ be the set of windows in which both calls are visible. If c_1 does not completely overlap c_2 and vice versa, there will be some subset of windows $\{w_i^1\}$ for which c_1 is the most prominent call, and another subset $\{w_i^2\}$ for which c_2 is most prominent. For 95 percent of calls in the training data set, the start and end times of c_1 and c_2 both differ by more than one pooled time step, so no call from $\{w_i^1\}$ will be stitched to a call from $\{w_i^2\}$. This can be visualized in Figure 7. The algorithm links pairs of prediction fragments that meet this start and end time criterion. The connected components are then stitched into a single prediction. To reduce the number of false positives, we throw out any stitched prediction made from fewer than three prediction fragments. This process is described in detail in Algorithm 1.

4.3.2. Results and Discussion

We test the network with 14, 28, and 56 output bins, and with a continuous regression output. Figure 8 displays the results. The 28-bin model achieves the highest F1-score of $\mathbf{0.77}$, with a precision of $\mathbf{0.82}$ and a recall of $\mathbf{0.72}$. Nevertheless, it does not appear that bin size has much effect on performance. All binned models significantly outperform the continuous regression. We believe this is because the accuracy of the regression predictions are insufficient to accurately determine which prediction fragments correspond to the same call.

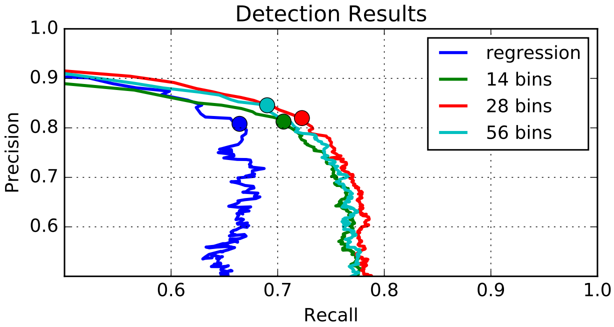


Figure 8: Precision/recall for detection task.

For the 28 bin network, the network recall on overlapping calls was $\mathbf{0.64}$, compared with the overall recall of $\mathbf{0.72}$. While the network is able to identify overlapping calls, the difference in the recalls suggests that the network should be trained on more samples with overlap. This will allow the network to better distinguish the start and end times of individual calls. It is possible that improvements to the stitching process could also improve this overlap recall.

5. Conclusion

In this paper, we introduce "ElephNet," a large-scale dataset with accompanying challenge tasks. We discuss the difficulties associated with this dataset — namely, adapting to unknown environments and learning from human labeling error. We introduce convnet baselines for all challenge tasks that are able to achieve high performance in an environment withheld from the training data, and we achieve especially promising results for the segmentation task. For detection, our novel stitching algorithm accommodates variation in call duration while still differentiating between individual calls. We hope that researchers will utilize this dataset to improve upon our baseline model. In particular, a promising future direction is differentiating between volunteer and expert labels during training. Developing methodologies to incorporate this information will produce models which approach expert-level performance.

References

Abdel-Hamid, Ossama, Deng, Li, and Yu, Dong. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *INTERSPEECH*, pp. 3366–3370, 2013.

Bardeli, R., Wolff, D., Kurth, F., Koch, M., Tauchert, K. H., and Frommolt, K. H. Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring. *Pattern Recognition Letters*, 31(12):1524–1534, 2010.

Baumgartner, Mark F. and Mussoline, Sarah E. A generalized baleen whale call detection and classification system. *The Journal of the Acoustical Society of America*, 129(5):2889–2902, 2011.

Collobert, Ronan, Kavukcuoglu, Koray, and Farabet, Clément. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

Damoulas, T., Henry, S., Farnsworth, A., Lanzone, M., and Gomes, C. Bayesian Classification of Flight Calls with a Novel Dynamic Time Warping Kernel. In *2010 Ninth International Conference on Machine Learning and Applications (ICMLA)*, 2010.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

Deng, Li, Hinton, Geoffrey, and Kingsbury, Brian. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8599–8603. IEEE, 2013.

Dufour, Olivier, Artieres, Thierry, Glotin, Herv, and Giraudet, Pascale. Clusterized mel filter cepstral coefficients and support vector machines for bird song identification. In *Proc of 1st workshop on Machine Learning for Bioacoustics, joint to*, 2013.

Everingham, Mark, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Garofolo, John S, Lamel, Lori F, Fisher, William M, Fiscus, Jonathan G, and Pallett, David S. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93, 1993.

Jarvis, Susan, DiMarzio, Nancy, Morrissey, Ronald, and Moretti, David. A novel multi-class support vector machine classifier for automated classification of beaked whales and other small odontocetes. *Canadian Acoustics*, 36(1):34–40, 2008.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The mnist database of handwritten digits, 1998.

Potamitis, I., Ganchev, T., and Fakotakis, N. Automatic acoustic identification of crickets and cicadas. In *9th International Symposium on Signal Processing and Its Applications, 2007. ISSPA 2007*, 2007.

Pourhomayoun, Mohammad, Dugan, Peter, Popescu, Marian, and Clark, Christopher. Bioacoustic Signal Classification Based on Continuous Region Processing, Grid Masking and Artificial Neural Network. *arXiv preprint arXiv: 1305.3635*, 2013.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

Ross, Jesse C. and Allen, Paul E. Random Forest for improved analysis efficiency in passive acoustic monitoring. *Ecological Informatics*, 21:34–39, 2014.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pp. 1–42, 2014.

Shamir, Lior, Yerby, Carol, Simpson, Robert, von Benda-Beckmann, Alexander M., Tyack, Peter, Samarra, Filipa, Miller, Patrick, and Wallin, John. Classification of large acoustic datasets using machine learning and crowdsourcing: Application to whale calls. *The Journal of the Acoustical Society of America*, 135(2):953–962, 2014.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.