AudioMask: Robust Sound Event Detection Using Mask R-CNN and Frame-Level Classifier

Preprint · October 2019

CITATIONS
0

READS
797

6 authors, including:

Alireza Nasiri
New York Structural Biology Center
22 PUBLICATIONS **186** CITATIONS
SEE PROFILE

Yuxin Cui
University of South Carolina
23 PUBLICATIONS **672** CITATIONS
SEE PROFILE

Yong Zhao
Northeastern University (Shenyang, China)
27 PUBLICATIONS **170** CITATIONS
SEE PROFILE

AudioMask: Robust Sound Event Detection Using Mask R-CNN and Frame-Level Classifier

Alireza Nasiri, Yuxin Cui, Zhonghao Liu, Jing Jin, Yong Zhao, and Jianjun Hu
Dept. of Computer Science and Engineering
University of South Carolina
Columbia, SC, 29201 USA
Email: {anasiri, ycui, liu338, jingj, yongz}@email.sc.edu, jianjunh@cse.sc.edu

***Abstract*—Deep learning methods have recently made significant contributions to sound event detection. These methods either use a block-level approach to distinguish parts of audio containing the event, or analyze the small frames of the audio separately. In this paper, we introduce a new method, AudioMask, for rare sound event detection by combining these two approaches. AudioMask first applies Mask R-CNN, a state-of-the-art algorithm for detecting objects in images, to the log mel-spectrogram of the audio files. Mask R-CNN detects audio segments that might contain the target event by generating bounding boxes around them in time-frequency domain. Then we use a frame-based audio event classifier trained independently from Mask R-CNN, to analyze each individual frame in the candidate segments proposed by Mask R-CNN. A post-processing step combines the outputs of the Mask R-CNN and the frame-level classifier to identify the true events. By evaluating AudioMask over the data sets from 2017 Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge Task 2, We show that our algorithm performs better than the baseline models by 13.3% in the average F-score and achieves better results compared to the other non-ensemble methods in the challenge.**

***Index Terms*—Sound Event Detection, Mask R-CNN, Audio Analysis, Audio Classifier.**

I. Introduction

Sound event detection (SED), which deals with sound analysis in order to identify everyday audio events, has received a lot of attention recently. It has many interesting applications in environmental and wildlife surveillance systems[2], [3], smart homes [4], [15], and video event detection [5].

Deep learning (DL) models have become very popular in detecting sound events in recent years [8], [11], [12]. These DL models either use frame-level approaches or region-based ones. In the frame-level approach [8], [12], they analyze every small frame in audio to determine if it belongs to an event or not. The architectures of these models are usually based on convolutional and recurrent layers. Convolutional layers are applied to extract high-level features before recurrent layers are applied to learn the longer term temporal dependencies among them. One of the biggest downsides of the frame-level methods is that they fail to consider the longer contextual dependencies in the audio. This is addressed by the second group of DL-based SED models: the region-based approaches [10], [11]. These algorithms treat mel-spectrograms of audio as images and use object detection models from the computer vision field, to identify segments of audio that belong to different types of events. Wang et. al [10] used a region-based approach by employing a slightly modified version of R-FCN to detect rare sound events. R-FCN is a fully convolutional neural network for object detection [21]. Another region-based approach is proposed by Kao et. al [11]. They used a network called CRNN, which has a similar architecture to the object detection algorithm Faster-RCNN [13], to get the regions of interest and classify them as the event/not-event.

Region-based models can be useful in addressing some of the main challenges in SED according to [17], which is creating tight boundaries around the event considering the background noise. These models detect events by finding their patterns in log mel-spectrograms. This strategy might be problematic because some events do not have very distinguishable shapes such that these methods might end up with non-trivial number of false positive predictions.

In this paper, we propose AudioMask, a novel algorithm for rare sound event detection by taking advantage of both region-based and frame-based methods. To identify candidate event-regions from the audio, we first take a region-based approach by using Mask R-CNN model [9] as a region proposal network to identify potential event-regions. Mask R-CNN is a state-of-the-art object detection and segmentation model that outperformed all of the winners of the Common Objects in Context (COCO) 2016 challenge [20]. It has a special capability to create tight and highly accurate boxes around the target objects, which allows us to identify candidate audio events with tight, pixel-accurate bounding boxes. After potential event-region detection, we then analyze each of these regions by looking into the small frames inside them using a frame-level classifier composed of several convolutional and recurrent layers. This technique achieves competitive results in detecting the audio events with highly variant and non-regular shapes in log-mel spectrograms.

The contribution of this paper includes the following:

- We take advantage of Mask R-CNN, a state-of-the-art object detection model in computer vision, to propose regions of audio as candidate event-regions that have similar patterns to the target events.

This work was partially supported by the NSF grant 1655740 and by National Natural Science Foundation of China with grant no. 51741101

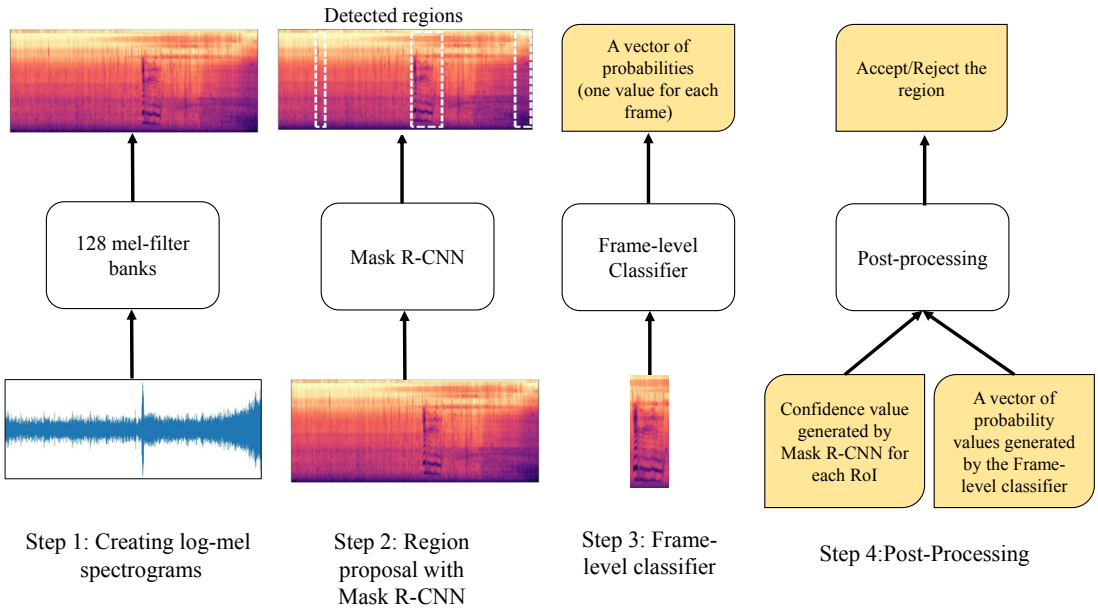

Fig. 1. Four steps in AudioMask. Each detected region by Mask R-CNN is mapped to a segment of the audio with a fixed length on the mel spectrogram, before being fed to the frame-level classifier

- We introduce a frame-level classifier, based on the combination of convolutional and recurrent layers to analyze frames in each candidate segment and identify the true event-regions out of the proposed ones identified by the Mask R-CNN model.
- Experimental results indicate that our AudioMask algorithm outperforms all of the non-ensemble methods of DCASE 2017 [1] challenge and improves the baseline model by 13.3% in F-score.

II. Proposed Method

We introduce AudioMask, a novel method that takes advantages of both region-based methods and frame-level analyses. Our algorithm consists of four stages: 1) extracting log mel-scaled energies from audio, normalizing and pre-processing them, 2) training a Mask R-CNN to identify chunks of the audio that potentially belong to the target event, 3) training a frame-level classifier that analyzes the frames inside each candidate segment of the audio and outputs a probability value for them, and 4) post-processing based on the confidence values generated via Mask R-CNN and probabilities from the frame-level classifier to identify the true event-regions. Fig 1. shows the general framework of our proposed AudioMask algorithm.

A. Feature Extraction based on Log mel-scaled filter banks

We use pre-processed and normalized log-scaled mel spectrograms as input features to our model. Spectrograms are 2D representations of audio in time-frequency domain with brightness or color representing strength of signals of specific frequencies in that time frame. They preserve more information than most of hand-crafted features [7]. The log-mel feature-map exhibits locality in both time and frequency domains [19].

Mel-spectrograms have proven to be good features of audio for many deep learning based sound event detection methods [8], [10], [11], [12]. Their difference with normal spectrograms is that log mel-scale filter banks are used in them to imitate the non-linear human ear perception of sound [8].

We apply a window size of 46 ms, being overlapped with half of its size to the audio signal to extract 128 mel-filter banks for each frame of the audio signal. The window size and number of the mel-filter banks are the same as [10]. We then calculate the logarithms of these filter bank values and normalize them.

B. Region-Proposal with Mask R-CNN

We use Mask R-CNN to detect precise bounding boxes around areas of the spectrogram that possibly corresponds to a target event. Mask R-CNN is a fully convolutional network that has shown great performance in object detection and segmentation in computer vision. It outputs a class label and a tight bounding-box offsets for each candidate object, along with a mask for each object. Fig 2. shows how Mask R-CNN works in general.

Mask R-CNN framework for object detection consists of two stages: Region Proposal Network (RPN) is the first stage which outputs approximate bounding boxes for potential objects in the image. The second stage is responsible for predicting object class labels, tightening bounding boxes and creating object segmentation masks. In RPN stage, a backbone of the residual blocks of ResNet-101 network is used to extract high-level features out of the images. Later, a small

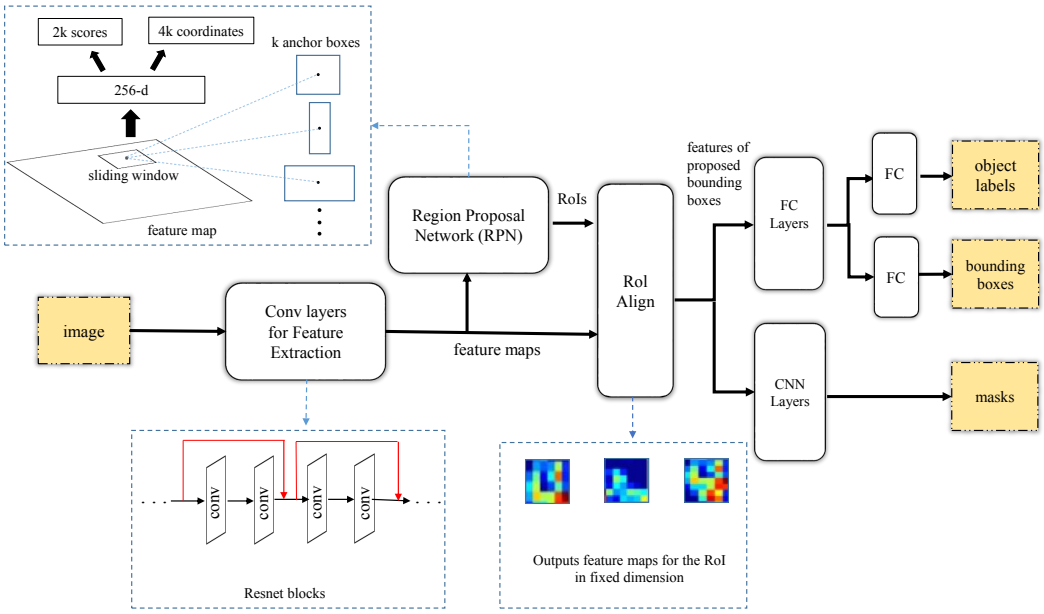

Fig. 2. Mask R-CNN framework for object detection

network slides over this feature map to produce a set of region proposals along with an objectness score for each region. This small network predicts k region proposals at each sliding-window location, by outputting 4k values as coordinates and 2k values as probability estimates of object / not-object for each proposed rectangular region [13].

Poposed regions can be of different dimensions and need to be represented with a fixed-length feature map before being fed to the fully connected layers. Mask R-CNN uses a method called RoIAlign to create fixed-size feature maps for each region while ensuring a close correspondence between the values on the feature map and the regions on the actual image. In RoIAlign, when a RoI is not aligned exactly with the values in the feature map, instead of using harsh quantization and mapping the RoI to the discrete granularity of the feature map as done in Faster R-CNN[13], bi-linear interpolation is used to compute more accurate values of the features [9] . Fig 3. shows an example of how RoIAlign uses bi-linear interpolation to create a better estimation of features for the boxes of the objects.

Mask R-CNN through the RoIAlign procedure, makes sure that the masks and the bounding boxes have a pixel-to-pixel alignment on the real objects in the image. This makes non-trivial improvements in object detection accuracy [9]. This feature of Mask R-CNN is a major reason that makes it very suitable for SED, in which detecting the exact onset of the event is crucial. Indeed, according to the evaluation metrics of sound event detection as DCASE 2017 did [1], onset detection is the defining factor in correctly detecting an audio event.

A major module of Mask R-CNN is a parallel branch for segmentation mask prediction as shown in Fig 2. This feature is believed to improve the performance of Mask R-CNN in object detection in images. We did not utilize the segmentation branch because generating segmentation masks for audio events like 'gunshot' is problematic as there is no clear borders for audio events and are also not provided in any of the known data sets.

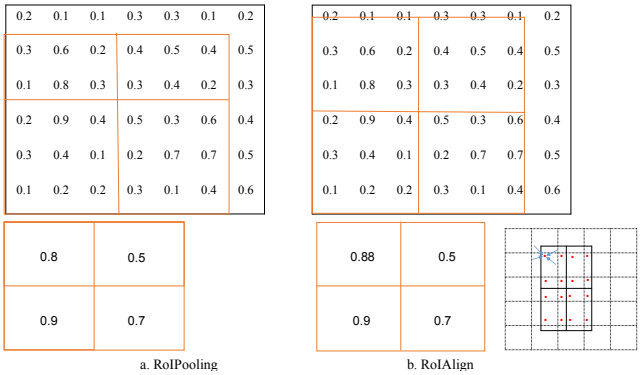

Fig. 3. (a) RoIPooling used in Faster R-CNN; (b) RoIAlign used in Mask R-CNN. RoIPooling gets the bounding box for each RoI and divides it to smaller sub-windows by quantization, which is mapping the borders of these sub-windows to the closest cells on the feature map. Then it max-pools each one of these sub-windows to get a feature representation for the RoI. The quantization process will cause misaligned feature values for RoIs. RoIAlign avoids this problem by using bilinear interpolation. The bottom right picture shows how this process works in general [9]. The dashed grid is the feature map and the solid one represents an RoI and red dots are 4 sampling points in each bin. RoIAlign computes the value for each one of these bins, by bilinear interpolation from the overlapping grid cells. Notice how the interpolation operation in RoIAlign changes the maximum feature value for the top left cell from 0.8 to 0.88.

We train a Mask R-CNN model for each event using log-mel spectrograms extracted from the audio files. We utilize Mask R-CNN to generate bounding boxes around spectrogram areas

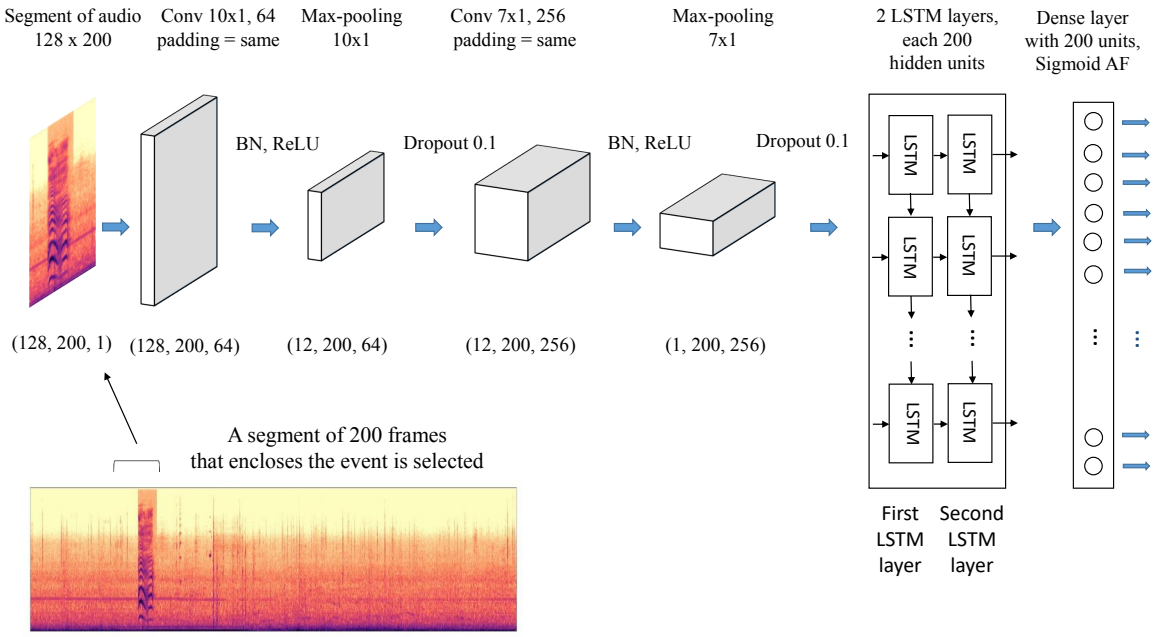

Fig. 4. Frame-level classifier

that show high resemblance to the target event. We identify these regions of the spectrograms as the potential event-regions that might contain the event. Since we define target objects as audio events that can occur with variant lengths across the time axis, we modify the Mask R-CNN models to only generate rectangular bounding boxes that cover all of the frequency bins. In other words, all of the potential event-regions, have the same height as the frequency bins in the log-mel spectrogram and they cover variant ranges on the time axis. Mask R-CNN assigns a confidence value to each of these boxes based on their similarity to the general pattern of the target audio event.

After training, we choose a model that has the lowest validation loss to be used on the test stage. During the test stage, we collect all of the event-regions with a confidence value above a specific threshold. A list of these proposed regions will be forwarded to the frame-level classifier.

At the end of this step, we have a list of candidate event-regions on the log-mel spectrograms for each target event. Now we need to filter out true events out of them, which is done by the frame-level classifier.

C. Frame-level Classifier

While Mask R-CNN generates a large number of potential event-regions with tight boundaries, it may also report many false positives. This is partially due to the non-regular shapes of the events in the time-frequency context that can lead to detecting parts of the background noise as the target event. This problem is more severe for short events like gunshots which have highly variant patterns in terms of their mel-spectrogram representations.

To filter the real events out of the proposed event-regions, we train a classifier that receives segments of the audio represented as log-mel values and determines if that region belongs to a specific event or not. These segments are consisted of log-mel frequency vectors of a number of consecutive time-steps. We call each one of these vectors, a frame. Each frame gets a label of 0 or 1, based on whether it belongs to the event or to the background noise, respectively. The output of the frame-level classifier is a one-dimensional vector that consists of probability of belonging to the target event for all of the frames in the segment.

This classifier consists of two convolution layers, two LSTM layers and a fully connected layer. The convolution layers have 64/256 one-dimensional filters of size 10/7 that stride over the mel-spectrograms across the frequency axis. Each convolution layer is followed by a batch normalization(BN), rectified linear unit (ReLU) activation function, and a max-pooling layer. The size of filters in the first and second max-pooling layers are 10 and 7, respectively. A dropout layer with a rate of 0.1 is applied to the outputs of the max-pooling layers.

The LSTM layers are used to capture the temporal context across the segment using the high-level features extracted by the convolution layers. The last layer of the segment classifier is a fully connected layer (FC) with Sigmoid activation function that produces a probability value for each frame in the segment. Our segment classifier model is inspired from the model in [8] , although we observed that using 2 convolution layers with smaller filter sizes has better performance. Fig 4. shows the architecture of our frame-level classifier.

D. Post-processing

In this step, we utilize both block-level analysis by Mask R-CNN and frame-level analyses by the classifier to detect the true event boundaries. Frame-level classifier outputs a

probability value for each frame in the segment. In the post-processing step, we calculate the average of the probability values of the frames to obtain a single probability value for the whole segment. We refer to this value as α. Also, we represent the confidence value calculated by the Mask R-CNN for each event-region as β. We use formula (1) to utilize both probability values from the frame-level classifier(α) and confidence values from Mask R-CNN(β) to calculate a score value ($0 \leq D \leq 1$) for each segment.

$$D = \lambda * \alpha + (1 - \lambda) * \beta \tag{1}$$

In the DCASE 2017 benchmark dataset, each audio file can have at most one event in it. So we compare the D values of all of the proposed regions for one audio and choose the event-region with the highest score as the true boundaries for the target event. We also reject the regions with a D value below a certain threshold, which can be determined using the training set.

III. Experiments

A. Data

We use the data set from task 2 of the DCASE 2017 challenge to demonstrate the performance of our method. This dataset consists of isolated recordings for three target events including 'babycry', 'glassbreak' and 'gunshot'. It also contains recordings of 15 different audio scenes to be used as background sounds. We used the synthesizer provided by the task 2 of DCASE 2017 challenge to create 5000 monophonic audio files for each event with 44.1 KHz sampling rate and event-to-background ratio (EBR) of -6, 0 and 6 dB.

To ease the problem of data imbalance, we set the probability of event being present in each audio to one as the DCASE challenge lets this modification for the training set. We also use a validation set of 1000 audio files for each event. We evaluate our method on the development and evaluation set provided by task 2 of the DCASE 2017 challenge.

B. Evaluation Metrics

We use event-based error rate(ER) and F-score to evaluate our method [14]. These criteria can be calculated based on counting the true positives(TP), false positives(FP) and false negatives(FN). In the event-based metrics, there is not any meaningful true negatives(TN) [14]. Based on the description of the task 2 of DCASE 2017 challenge [1], the challenge considers a detected event to be a TP, if its detected onset time would be within 500 ms collar of the actual onset time.

Error-rate and F-score are defined as:

$$ER = \frac{FN + FP}{N} \quad \text{and} \quad F = \frac{2TP}{2TP + FP + FN}$$

We have used the sed-eval toolbox provided by the challenge organizer to calculate these metrics [14].

C. Results and Discussion

The synthesizer for DCASE 2017 task 2 puts events inside 30 seconds long audio files with specific start and finish times for the event. We use these start and finish times to create rectangular bounding boxes for the purpose of training a Mask R-CNN model for them. The height of the bounding boxes is fixed and is equal to the number of mel-bands we used to create mel-spectrograms and their width is equal to the length of the event. For each event type, we train a Mask R-CNN model with ResNet-101 backbone for 50 epochs and save the model at each epoch. We have modified the RPN module in the Mask R-CNN to only generate rectangular regions with the fixed height of 128 to cover all of the frequency bins and variant width of t. We choose the model with the least error on the validation set and run it on the test set to generate a list of regions that potentially contain the target event with a confidence value of 0.6 or higher. We have chosen 0.6 as the detection threshold because we found that almost all of the events in the development set, are detectable by Mask R-CNN at this threshold level. Gunshot is the only event type that Mask R-CNN is unable to detect all the true instances for it. We observed that lowering the confidence threshold on the development set did not lead into proposing more true regions by the Mask R-CNN and it rather increased the number of falsely detected regions.

We also train a frame-level classifier for each event type. To prepare the inputs for training these classifiers, we only consider frames of the event and frames in their vicinity. In training step, we select a segment of audio with 200 frames where the specific event happens at some point inside it. We randomly choose the start of these segments so the event might happen at the start, middle or end of it. In the test step, we convert the proposed event-regions by Mask R-CNN to segments with the fixed length of 200 frames, so if the proposed region is larger than 200 frames, it will lose some of its frames before being analyzed by the frame-level classifier and if the proposed region is smaller than 200 frames, we would add some neighbor frames to it. We choose 200 frames as the segment size, because it covers 4.6 seconds of each audio and based on our calculations, this is enough to cover almost all of the events and gives us a balanced data set of audio segments with almost equal numbers of overall positive and negative frames. To label each frame in the segment, binary values are used based on whether that frame belongs to the event or not. As a result, for a training set of 5000 audios that is generated using the DCASE synthesizer with event being present in all of them, we create a new data set of 1000 audio segments of 200 frames where each segment has a label vector of 200 binary values.

The frame-level classifier analyzes each frame in the segment and assigns a probability value of belonging to the specific event to each frame. So the output of the classifier is a vector of 200 probability values.

We have conducted a grid search to determine the best filter size, number of filters, batch size, learning rate and dropout

TABLE I
AUDIMASK'S PERFORMANCE WITH DIFFERENT λ VALUES ON DCASE 2017 DEVELOPMENT SET

	babycry		glassbreak		gunshot		Average	
	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER
$\lambda = 0$	90.8	0.18	76.6	0.51	62.8	0.86	76.7	0.52
$\lambda = 0.25$	91.2	0.17	87.2	0.26	70.2	0.66	82.9	0.36
$\lambda = 0.5$	92.5	0.14	87.2	0.26	72.0	0.55	83.9	0.32
$\lambda = 0.75$	92.4	0.15	88.2	0.24	73.5	0.50	84.7	0.30
$\lambda = 1$	91.6	0.16	83.8	0.32	72.3	0.51	82.6	0.33

TABLE II
PERFORMANCE COMPARISON OF OUR MODELS VERSUS BEST NON-ENSEMBLE METHODS ON DCASE 2017 EVALUATION SET

	babycry		glassbreak		gunshot		Average	
	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER
AudioMask (Our method)	90.2	0.19	91.2	0.18	76.4	0.46	85.9	0.28
DNN/CNN [16]	85.7	0.28	88.8	0.22	81.6	0.33	85.3	0.28
SLR-NMF[6]	91.4	0.17	89.1	0.22	72.0	0.55	84.2	0.31
R-FCN [10]	87.3	0.26	91.5	0.16	67.2	0.53	82.0	0.32
Baseline [1]	70.7	0.57	81.0	0.36	66.0	0.57	72.6	0.50

rate for the frame-level classifier using the development set. Our model is trained using ADAM optimizer with a batch size of 150 and learning rate of 0.001 for 100 epochs. The rest of the hyperparameters are shown in Fig 4. We use early stopping with minimum improvement of 0.0001 on validation accuracy and patience of 10 epochs to prevent over-fitting the model on the training set.

In the test stage, log-mel values for the audio are fed to Mask R-CNN to detect the regions potentially containing an event. We extract a segment of 200 frames from each of these proposed regions ,using the method that was explained above, and feed them to the frame-level classifiers. In the post-processing step, we calculate the average of probability values generated by the frame-level classifiers and use formula 1 to detect the true event out of the proposed ones. We conducted experiments with different values for λ to study the effect of the frame-level analysis on the final results as shown in Table 1. When $\lambda = 0$, only the confidence values from the Mask R-CNN is used and when $lambda = 1$, we are only using the probability values from the classifier to detect the true events out of the proposed regions by the Mask R-CNN. It is obvious from the table that different values of λ does not affect our model's performance on 'babycry'. This is due to more distinctive shape of this event in the mel-spectrograms that allows Mask R-CNN to successfully detect them with high confidence. Fig 5.a, shows a 'babycry' event-region detected by Mask R-CNN, that is clearly distinctive from its background noise.

Frame-level analysis plays more important role on the detection of true 'glassbreak' and 'gunshot' events. Because of the shorter and less identifiable shapes of these events, Mask R-CNN proposes more regions as the potential event. Parts b and c of Fig 5, shows examples of the generated bounding boxes by Mask R-CNN for these events along with their confidence values. Our results show that event-detection in audio by only employing object-detection models from the computer vision can not yield the best results for the shorter, less distinctive events and a finer analysis is required. As shown in the table, by increasing the weight of the frame-level classifier from 0 to 0.75, the F-score on the 'glassbreak' event increases significantly from 76.6% to 88.2%. This increase in λ value has a non-trivial impact on our model's performance over the 'gunshot' event too, where its F-score increases about 11.7%. We choose 0.75 as the λ value since it leads to the best performance on the development set with average F-score of 84.7% and ER of 0.30 over all three events.

Overall, Our model shows outstanding results in detecting 'babycry' and 'glassbreak' events, which is due to the detectable patterns of these events in the mel-spectrograms. The 'gunshot' events can be very short and there are some background sounds in the audio that look very similar to the patterns of the 'gunshot' event in the mel-spectrograms. Our model has its lowest performance on this event compared to the other two events. It achieves an F-score of 73.5% on the development set for the 'gunshot' event. This comparatively lower performance is mostly due the fact that Mask R-CNN is an object detection model developed to detect objects with clear shapes or patterns in the images, and its performance drops in detecting boundaries for the 'gunshot' events with non-regular shapes in their log-mel spectrograms. Our method shows much better performance with events like 'babycry' and 'glassbreak' which have more regular and detectable shapes in the mel-spectrograms.

In Table 2, we compare the average F-scores and ERs achieved by our model with four non-ensemble sound event detection methods that had the highest performance at DCASE 2017 challenge. The evaluation is done over the DCASE 2017 evaluation data set. DNN/CNN model as reported in [16] has

TABLE III
AUDIOMASK'S PERFORMANCE ON TWO EXTRA TEST SETS GENERATED USING DCASE 2017 SYNTHESIZER

	babycry		glassbreak		gunshot		Average	
	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER	F-score(%)	ER
test-set-1	95.6	0.09	93.7	0.13	88.7	0.22	92.7	0.15
test-set-2	93.5	0.13	96.4	0.07	87.1	0.26	92.3	0.15

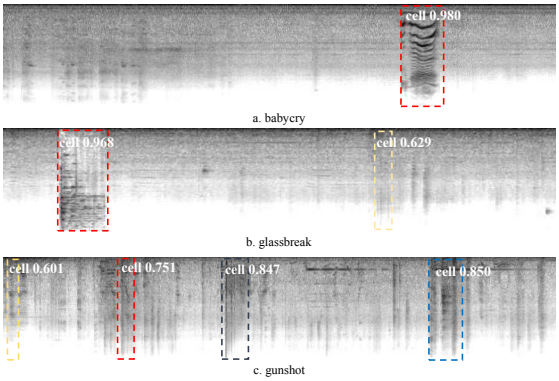

Fig. 5. Regions detected by Mask R-CNN for a.babycry, b.glassbreak and c.gunshot events. The red regions are the actual events

the best performance among all non-ensemble methods in DCASE 2017 and was ranked at third place in the challenge (The top two methods [8], [12], use sophisticated ensemble methods). The DNN/CNN model uses frame-level analysis to detect the onset of events. In Table 2, we cited the F-scores and ERs calculated by the challenge's official website for this method, which is slightly less than what authors reported in their paper. Our method achieves a slightly higher average F-score of 85.9% compared to 85.3% of DNN/CNN. Also we improved the F-score by 4.5% and ER by 0.09 for 'babycry' events. The improvement for 'glassbreak' events is 1.4% in F-score and 0.04 in ER. The only event that our method shows worse performance compared to DNN/CNN is 'gunshot' for which they use a network based on fully connected layers to detect this event.

We also compare our model against SLR-NMF method which was ranked 4th in the challenge and uses non-negative matrix factorization to filter out the noise and detect the event [6]. Although, this method has better performance on 'babycry' with F-score of 91.4%, its overall average F-score across all three events is 84.2% which is 1.7% less that AudioMask's overall average F-score.

Our model also outperforms R-FCN [10], a region-based method that is inspired by an object detection model. Our AudioMask algorithm improved their average F-score by 3.9% and the average ER by 0.04 over the evaluation set. Our algorithm shows relatively similar results over 'glassbreak' events but increased the F-scores by 2.9% and 9.2% on 'babycry' and 'gunshot' events respectively.

Furthermore, our model in average has 13.3% higher F-score than the baseline model and performs significantly better in detecting all of the three events. Overall, AudioMask performs better than all of the challenge's non-ensemble methods on the evaluation set of DCASE 2017.

To ensure the strength of our model, we used DCASE 2017's data synthesizer to create two more data sets using random seeds. This synthesizer can generate different data sets by changing the seed values. The values of 32 and 57 were randomly chosen as the seed values to create test-set-1 and test-set-2. We follow the same settings as the evaluation set of the DCASE 2017 challenge with the probability of event presence of 0.5 and 500 generated audios for each event. Table 3 shows the performance of AudioMask on these data sets. We achieve an average F-score of 92.7% and 92.3% on these sets using the exact same models as we used on the evaluation set. On these sets, AudioMask shows a better performance on 'babycry' and 'glassbreak' compared to 'gunshot' which confirms our assumption that our model has better performance on detecting events that have more distinguishable shape on the mel-spectrogram.

IV. CONCLUSION

We proposed AudioMask, a novel sound event detection algorithm based on the Mask R-CNN framework and frame-level audio analysis. Our algorithm exploits the power of Mask R-CNN to create bounding boxes around candidate events, which are then fed as input to a frame-level classifier for finer analysis to determine if the candidate regions can be categorized as the target event or not. By comparing to the top non-ensemble algorithms of the DCASE 2017 task 2, we show that our method achieves higher performance.

We believe our AudioMask algorithm can be generalized to detect other audio events of interest in the environment, especially those that are relatively longer and have a specific shape in the mel-spectrogram representation.

REFERENCES

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), pp. 85-92, November 2017.

[2] S. Chu, S. Narayanan, and C.-C. Jay Kuo, "Environmental sound recognition with timefrequency audio features", IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, no. 6, pp. 1142-1158, 2009.

[3] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection", Proceedings of the 25th European Signal Processing Conference (EUSIPCO), pp. 1744-1748, August 2017.

[4] A. Harma, M. F. McKinney, and J. Skowronek, (2005, July). "Automatic surveillance of the acoustic activity in our living environment", Proceedings of the IEEE International Conference on Multimedia and Expo, 2005.

[5] Y. Wang, L. Neves, and F. Metze,(2016, March). "Audio-based multimedia event detection using deep recurrent neural networks", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2742-2746, 2016.
[6] Q. Zhou and Z. Feng, "Robust sound event detection through noise estimation and source separation using NMF", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), pp. 138-142, November 2017.
[7] L. Wyse, "Audio spectrogram representations for processing with convolutional neural networks", Proceedings of the First International Conference on Deep Learning and Music, pp. 37-41, May 2017.
[8] H. Lim, J. Park, K. Lee, and Y. Han, "Rare sound event detection using 1D convolutional recurrent neural networks", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), pp. 80-84, November 2017.
[9] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn", Proceedings of the IEEE international conference on computer vision, pp. 2961-2969, 2017.
[10] K. Wang, L. Yang, and B. Yang, "Audio Event Detection and classification using extended R-FCN Approach", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), pp. 128-132, November 2017.
[11] C. C. Kao, W. Wang, M. Sun, and C. Wang, "R-CRNN: Region-based Convolutional Recurrent Neural Network for Audio Event Detection", Proceedings of the Interspeech conference, pp. 1358-1362, 2018.
[12] E. Cakr and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), pp. 27-31, November 2017.
[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", Proceedings of the Advances in neural information processing systems, pp. 91-99, 2015.
[14] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection", Applied Sciences, vol. 6, no. 6, pp. 162, 2016.
[15] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), pp. 3236, November 2017.
[16] H. Phan, M. Krawczyk-Becker, T. Gerkmann, and A. Mertins, "DNN and CNN with weighted and multi-task loss functions for audio event detection", DCASE2017 Challenge, Tech. Rep., 2017.
[17] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, I. McLoughlin, and A. Mertins, "What makes audio event detection harder than classification?", Proceedings of the 25th European Signal Processing Conference (EUSIPCO), pp. 2739-2743, August 2017.
[18] K. J. Piczak, "Environmental sound classification with convolutional neural networks", Proceedings of IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1-6, 2015.
[19] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, "Convolutional neural networks for speech recognition", IEEE/ACM Transactions on audio, speech, and language processing, vol. 22, no. 10, pp. 1533-1545, 2014.
[20] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick "Microsoft coco: Common objects in context", in European conference on computer vision (ECCV), pp. 740-755, 2014.
[21] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks", Advances in neural information processing systems, pp. 379-387, 2016.