Environmental Sound Classification for Flood Event Detection

Bipendra Basnyat[1], Nirmalya Roy[1], Aryya Gangopadhyay[1], Adrienne Raglin[2]
Department of Information Systems, University of Maryland Baltimore County[1],
DEVCOM Army Research Laboratory[2]
Email: b125@umbc.edu, nroy@umbc.edu, gangopad@umbc.edu, adrienne.raglin2.civ@army.mil

2022 18th International Conference on Intelligent Environments (IE) | 978-1-6654-6934-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/IE54923.2022.9826766

***Abstract*—Flood is one of the common natural disasters that can severely affect human life and properties. Early detection, therefore, is of paramount importance to provide help through an emergency response team. Robust flood detection techniques so far have been based on computer vision using images either from cameras, satellite imagery, remote sensing, or radar-based images. However, sound signal-based flood event detection has not been widely explored. In this work, we design an end-to-end architecture for a deep learning-based flood-related sound event detection model. We employ Mel-Spectrogram-based auditory signal analysis and deep learning models for sound event detection (SED). We evaluated four deep learning models under the following two categories: *(i)* Binary classification Flood/No Flood, vs. Windy vs. Non-Windy, and *(ii)* Multi-classification for more granular flood and wind events. The experimental results performed in these settings on the datasets collected from real deployment showed an accuracy of around 78%.**

***Index Terms*—Sound event detection, Deep Learning, Computer Vision, Acoustic Signal Processing, Mobile Computing**

I. Introduction

Sound signals can carry a large amount of information about the environment and surroundings. Humans can easily perceive the sounds associated with scenes such as forests, offices, kitchens, etc., and recognize sources of sounds that contribute primarily to the background, such as birds, people, utensils, etc. The ability to extract this information automatically has enormous potential in several applications, such as searching for multimedia based on its audio content, making context-aware mobile devices, robots, cars, etc. They can be further enhanced into an intelligent monitoring system to recognize activities in their environments using acoustic information [9]. Intelligent monitoring can be crucial during a natural disaster, including floods, earthquakes, hurricanes, snowstorms, and severe thunderstorms.

Flooding is one of the most severe yet common forms of natural disaster. Therefore, there is a great deal of interest in preventing the damages caused by the flood. An early accurate estimation of the flood can lead rescue teams to help the people affected areas mitigate the damage of flooding. Sound-based intelligent monitoring could be one such tool for early detection. In this work, we describe one such system developed to extract information in the context of a flooding stream. We show how information extraction from the surrounding can detect the environmental conditions related to the flood event.

Historically, flood detection has been done using hydrology and hydraulic model from civil engineering. Such models are site-specific, resource-intensive, and lack generalization. Automated processes are quick, cost-effective, and monitored from a distance compared to traditional methods, which require labor, in-person visits, and longer processing time [11]. We further argue that the detection model proposed in this work is compact, economical, and scalable. We foresee these systems being widely applicable not just in disasters but also in many other applications.

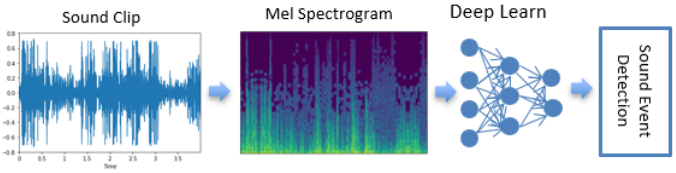

Fig. 1: Sound Event Detection-Approach

A broader goal of this work is to produce a context-aware computer system capable of processing real-world sound scenes of moderate complexity. The system should prepare an abstract representation of the sources in the sound and classify unseen data. Simply put, the systems should perform Sound event detection (SED). The objective of SED is to automatically identify the sound events, i.e., to associate distinctive concepts or label for sound segments. SED aims to detect each sound event's onset and offset times in an audio recording and associate a label with each of these events [7]. Sound events are good descriptors for an auditory scene as they help describe and understand human and social activities.

SED systems like ours are usually designed for specific tasks or environments. There are several challenges in extending the detection system to handle multiple settings and a large set of events. The most prominent challenge is the presence of multiple sound signals embedded within the event (e.g., heavy wind and rain). Therefore, we propose reducing the search space by providing the context in the sound event detection in the same manner as humans do [14]. Therefore, we have limited our experimental design to a given context of adverse weather such as flooding or detecting the wind conditions in a natural environment.

Unlike humans, computers cannot readily perform SED without prior knowledge and pattern recognition capabilities. To that end, we transform the raw sound signals into their time-frequency representation (Mel-Spectogram) then employ deep learning models to learn the mapping between this

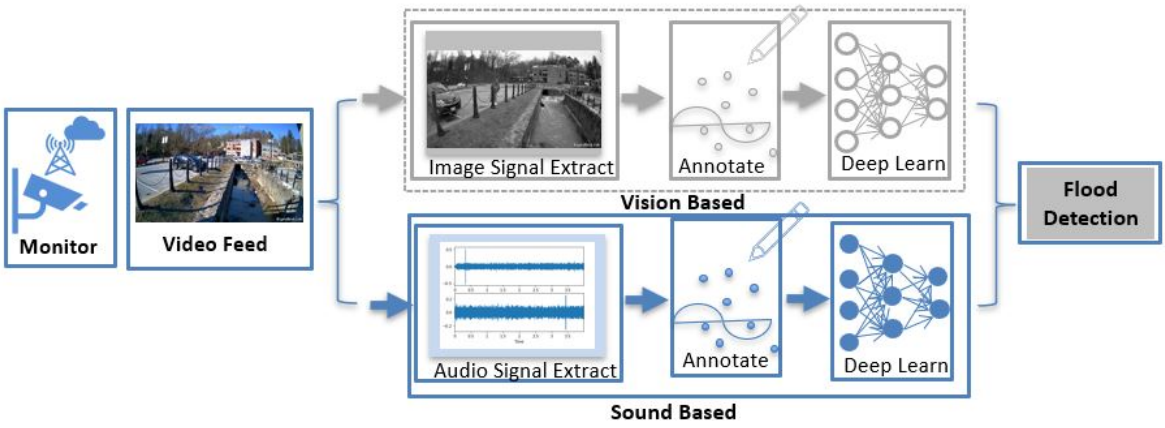

Fig. 2: Flood Detection Pipeline

representation and the target sound events. The high-level approach to sound event detection is shown in Figure 1. We envision SED-based models as auxiliary systems deployed as a trigger to more resource-intensive high accuracy computer vision based models.

We call the integrated end to end system 'FloodBot'. FloodBot is a network of a distributed system consisting of cameras, networking devices, and solar-powered field deployed sensors. Floodbot aims to observe the current situation of potential flood areas, infer relevant contextual information, and detect the potential risk. The major components and the flood detection pipeline are shown in Figure 2. The top part of the pipeline diagram in Figure 2 shows the vision-based flow, and the bottom pipeline depicts the work described in this paper. This work extends some of previously deployed Flood detection System using computer vision techniques [3, 5, 6, 4, 20]. While there have been great success with vision-based flood detection, the main challenge arises when the FloodBot loses visibility. Low illumination adverse weather conditions can cause visibility loss, often precursory to heavy rain and flood events. Therefore, we seek to improve the flood detection probability by exploiting sound as a signal in this work.

Key Contributions

The main contributions of this paper are:

- *Novel ESC Paradigm for Flood Detection*: We propose a novel design and algorithm that can be used as an alternative signal for flood Detection where traditional flood detection techniques may fail.
- *Annotated Natural Sound Data Set*: We use a weakly supervised labeling technique to label a large corpus of a natural sound event and release them in the public domain.
- *Enrichment for SED Research Area*: Our literature search did not find data pertinent to natural disasters or flood detection sound database. To address this gap and foster further research in this field, we create and publish novel Flood detection datasets for SED collected from urban environments.
- *Natural Sound Detection Model Architecture(s)*: Furthermore, we conduct baseline results on the datasets in a hazardous weather context and show their results with proposed deep neural network architectures.

In Section II, we present state-of-the-artwork that inspired us. In Section III, we discuss our framework and formalize the research problem. Section IV details our experimental data setup in depth. We then discuss our results in Section V. In Section VI, we present our findings and observations. In Section VII, we conclude by providing future directions in VIII.

II. Related Work

In this work, we try to enable devices to make sense of their environment through the analysis of sounds; this work belongs to a broader investigation area from machine listening related to computational auditory scene analysis [7, 2]. Machine listening systems perform analogous processing tasks to the human auditory system, which are part of a broader research theme linking fields such as machine learning, robotics, and artificial intelligence [1].

A. Acoustic scene classification

Acoustic scene classification (ASC) refers to the task of associating a semantic label to an audio stream that identifies the environment in which it has been produced [1]. Sawhney [18] proposed the first reported work in the ASC area in a 1997 technical report from the MIT Media Lab. Sawhney et al. recorded a dataset from a set of classes including 'people', 'voices', 'subway', 'traffic', and 'other'. Their best model based on recurrent neural networks and a K-nearest neighbor was able to achieve an overall classification accuracy of 68%.

ASC is also related to psychoacoustic/psychological studies aimed at understanding the human cognitive processes that enable humans to understand acoustic scenes [12]. However, the context of our work and this paper is focused on computational auditory scene analysis (CASA) [7]. A more focused algorithmic approach tries to identify the sources of acoustic events in a sound clipping and is closely related to event detection, and classification techniques [8]. The methods aim to identify and label temporal regions containing single events of a specific class. They have been employed, for example, in surveillance systems [17], and speech analysis through

segmentation of acoustic scenes [10]. This area and work are often categorized under Sound Event Detection (SED).

B. Sound Event Detection

Sound event detection (SED) aims to automatically identify the occurrence of target sound events, such as vacuum cleaner, thunderstorm, birds sound, or a dog barking, within an audio signal capturing an acoustic scene. The interest around increasing SED has increased lately, seen by the number of publications in the Detection and Classification of Acoustic Scenes and Events [9]. Since its first competition in 2013, the area has seen novel research ideas focused primarily on deep learning-based ASC algorithms.

Most SED algorithms build upon deep neural networks, specifically the convolutional neural network (CNN). Some work expands (CNN) to the convolutional recurrent neural network (CRNN) based architectures. They both employ convolutional front-ends, where multiple convolutional layers are trained to learn sound-specific features. As input to the network, either fixed two-dimensional signal transformations such as Mel spectrograms [22], or raw one-dimensional audio samples are used (end-to-end learning) [16]. Until the recent uprise in deep learning, traditional methods Gaussian mixture model (GMM) [21] was used to classify/detect acoustic events. Other conventional methods were proposed by hidden Markov model [13] and support vector machine [16].

C. Environmental sound classification

Within SED, further refined area branches into environmental sound classification (ESC). Environmental sound classification (ESC) is an emerging research area dedicated to identifying specific sound events, such as dog barking, gunshots, and air conditioning sound [15]. The availability of easy compute resources and data abundance has spurred much practical application for environmental sound classification. However, unlike speech and music, sound events are more diverse with a wide range of frequencies and often less well-defined, making ESC tasks more difficult than ASR and MIR. Hence, ESC still faces critical design issues in performance and accuracy improvement.

ESC has already been used in many practical applications [15], such as robotic hearing, smart home , audio monitoring systems, soundscape assessment, etc. ESC is a more challenging research problem than the regular musical note detection or speech recognition problem [19]. Additional challenges are caused because sounds generated by natural phenomena are not structured. Since environmental sound has neither static time patterns like melodies or rhythms nor semantic sequences like phonemes, it is difficult to find universal features representing temporal patterns. Moreover, the noise and parallel unrelated sound elements often found in nature can obscure the detection further by changing the composition structure with variability, diversity, and unstructured characteristics.

III. Framework

Large-scale weather-related sound recognition is essential for flood detection. It helps us quantify the possibility of flooding in the vicinity with ongoing weather patterns. Heavy flooding is accompanied by heavy sound event episodes such as thunderstorms, gusting winds, and the high-pitched sound of flowing water. The abundance of surveillance systems deployed in urban settings provides an extensive dataset that could be harnessed to build SED models. However, because recordings spanning hundreds of hours need to be carefully analyzed and categorized, flood detection and weather categorization remain almost an impossible task to be done manually. Therefore it seems natural to look at ways to automate the process. We collect video feed from the site, generate features to detect different flood-related sounds, and evaluate their performance.

A. Problem Formulation

We pose the Sound Detection Problem as a supervised classification problem where a set of training recordings $\mathcal{M}$ and labels: $\{c_m\}_{m=1}^{M}$ $\{s_m\}_{m=1}^{M}$ is provided. Let $\{\gamma_q\}_{q=1}^{Q} \in \mathcal{Q}$ possible categories $\in$ $\{$Low Wind,High Wind$\}$. Each label $c_m \subset \mathcal{Q}$, and we define a set $\Lambda_q = \{m : cm = \gamma_q\}$ that identifies the signals belonging to the q^{th} class. We train the model offline to learn the statistical properties of different classes and test them against unlabelled recordings s_new. Let $\mathcal{D}$ be the length of each frame, $s_{n,m} \in \mathbb{R}^D$ indicates the n^{th} frame of the m^{th} signal.

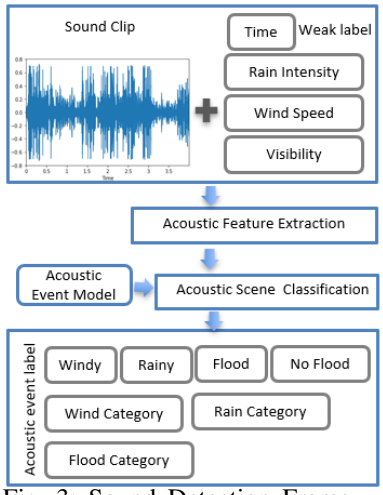

Fig. 3: Sound Detection Framework

Typically, $\mathcal{D}$ is chosen so that the frames duration is about 4 sec. Instead of using wave form in the time domain, we extract a sequence of features through a transform $\mathcal{T} : \mathcal{T}(s_n, m) = x_n, m$, where $x_{n,m} \in \mathbb{R}^K$ indicates a vector of features of dimension $\mathcal{K}$. Since $\mathcal{K} \ll \mathcal{D}$; $\mathcal{T}$ also causes dimensionality reduction. Let x_n, Λ_q indicate the features extracted from the signals belonging to the q^{th} category. The function $\mathcal{S}$: $\mathcal{S}(\{x_n, \Lambda_q\}) = \mathcal{M}$ learns the parameters of a statistical model $\mathcal{M}$ that describes the global properties of the training data. Once the training phase has been completed, and a model $\mathcal{M}$ has been learned, the transform $\mathcal{T}$ is applied in the test phase to a new unlabelled recording s_{new}, leading to a sequence of features x_{new}. A function $\mathcal{G}$: $\mathcal{G}(x_{new}, \mathcal{M}) = c_{new}$ is then employed to classify the signal, returning a label in set $\{\gamma_q\}_{q=1}^{Q}$. We show the graphical representations steps described above in Figure 3. The sound clips are labeled using weakly supervised techniques the passed through the deep learning models to categorize the sound and event association.

IV. Experiment Setup

This section describes the setup of our experiments, starting with the data collection, preprocessing, feature generation, and data annotation before describing the deep learning model.

A. Data Collection

We drive our process based on three data sources: the field camera's video, the real-time weather data API, and the manually labeled dataset for verification. Data is assimilated into their databases and made available for deep learning models, as described below. Figure 5 summarizes the total dataset and the observed weather pattern during the video recordings.

1) Flood Audio Database: The FloodBot implementation starts with real-time video acquisition from the deployment. These videos are transferred into the cloud for pre-processing. The Flood video database contains videos captured in various weather conditions for more than a year. Figure 4 shows the sample images captured during various ambient conditions from the testbed.

Fig. 4: Flood-Watch Recordings

2) Weather Database: We also collected the weather data from a public weather application interface during the same deployment time frame. We found the weather in the area when we captured those images. The weather data API allows us to extract weather at that particular location based on the latitude and longitude of our camera. We collect and store these records by minutes in our database.

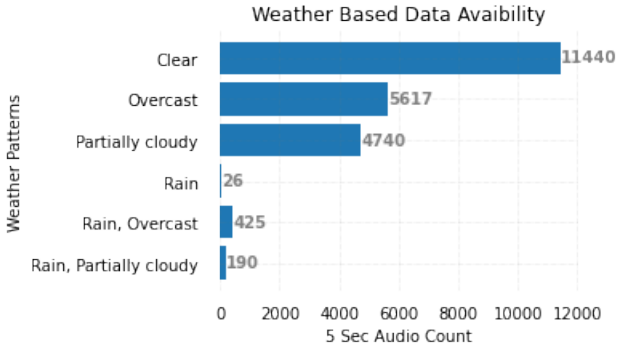

Fig. 5: Sample Data points & Weather Distribution

3) Temporal Sound Data: We establish a temporal join between a video captured time and weather timestamp to the weather condition of the site to our audio database. With a video to weather time tracking, we can visually observe the weather condition at the site (through video feed) and from data released by meteorological weather stations via their APIs. Both video feed and weather data have been rounded to their nearest five-minute time binning to capture and assign weather reading to as many videos as possible.

4) Pre-Processing: The Average video clipping size captured by our system is 30 seconds and hence the audio frame too. We first extract the sound from a video entirely and then shorten it to 4-sec frames each. Our preliminary experiments found that a 4-sec frame length is optimal from data storage volume and representation without cluttering them with uncorrelated sounds. We also maintain the splits and their source to locate them in different folds for cross-validations.

B. Feature Generation

Feature generation is essential in analyzing and finding relations between different things. The models cannot directly understand audio data provided to convert them into an understandable format feature extraction. It is a process that explains most of the data but is primarily understandable by the machine. We describe the feature generation steps in Algorithm 1.

Algorithm 1 Feature Generation

Input:Video Frames
Output:Sound Features, Sound Event Representation

1: Capture Video at $t_i; t_i \in \{$Natural Weather Occurrences$\}$
2: Extract Audio from Sample $\mathcal{Y}_i$
3: Split the sample into y_i where t_s = 4 sec
4: Compute Mel-Spectogram $\mathcal{M}_i$ for each y_i
5: Normalize $\mathcal{M}_i$
6: Persist Normalizes Image features for Deep Leaning

Therefore, feature extraction is precursory for any classification, prediction, and or recommendation algorithms. Creating a feature is crucial yet challenging because it involves numerous preprocessing steps. We discuss the steps involved in generating features in detail in this section. Audio data usually have complex features, so it is necessary to extract useful features to recognize the audio. The Mel-Spectrogram is one of the efficient methods for audio processing, and 8 kHz sampling is used for each audio sample. We used a Python package called Librosa for data processing with parameters (n fft =1024, hop length = 512, n mels = 128). Then we use Librosa's power to db function to convert the power spectrum (amplitude square) to decibel (DB) units.

1) Mel-Spectrogram: Most audio-based machine learning and deep learning, in general, are based on sound data represented in Mel spectrogram, Log -Mel spectrogram, and Mel Frequency Central Coefficients (MFCC) [17, 15, 19].

We use the log Mel spectrogram to train the deep neural network in our experiments. For other types of machine learning, such as Gaussian mixture models and hidden Markov models, an additional preprocessing step to generate Mel frequency central coefficients (MFCC) is performed. In Figure 6, we show some examples of Mel-spectrograms.The spectrograms depict how the sound pattern differs in different events. It is observed that the typical "No Flood" spectrograms event

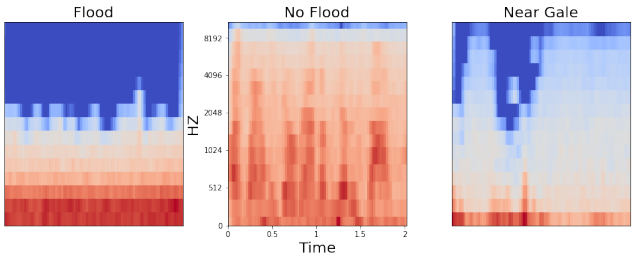

Fig. 6: Log Mel-Spectograms

seems normally distributed versus the Windy (Near Gale) sound spectrogram. We extract the audio features and transform them into feature images, so there are three channels like traditional color images. Spectrograms capture both time and the frequency associated with the raw sound wave for a better data representation based on short time for Fourier transform (STFT). STFT preserves the time domain data as a long vector computed using the wave's length(time), the sampling rate, and window size. Spectrograms are represented in hertz, which does not perceive the well human auditory system. We convert the linear frequency scale to the logarithmic Mel-scale and pass-through filter bank to get the eigenvector. These eigenvalues can be roughly expressed as signal energy distribution on the Mel-scale frequency. The output is saved as an image shown in Figure 6 to train the convolutional neural networks for their pattern recognition.

C. Data Augmentation

Data Augmentation is a compelling method used to represent the data better. The augmented data is expected to be a more comprehensive set of possible data points, thus minimizing the probability of disparity between the training and validation set. As seen from Figure 5, the data of our interest is quite limited. Naturally, adverse weather such as rain is comparatively less than the clear regular days.

Category	Fold	Count
Low Wind	7	2070
Low Wind	9	227
High Wind	1	640
High Wind	7	635
Low Wind	1	2102

TABLE I: Sample Data Distribution (Folds)

We show an example of a skewed ratio of majority to minority samples, e.g., 11K clear days versus 641 rain event-related data points. This is a classic example of class imbalance. We used data Augmentation techniques called an oversampling solution to alleviate the problem. We use a hybrid of over-sampling and under-sampling methods to achieve a balanced dataset for training and testing. During the over-sampling process, instances from the minority class (641 samples) were randomly duplicated (via replication). We also randomly remove sample data points from the majority class during the under-sampling process. Cross-Validation is a machine learning technique used when the available datasets are limited or not balanced like our case. Therefore, instead of training a fixed model only once with a train/test split, we iterated through several models on different data portions and validated the model on hold-outs. K-Fold is one common CV approach that we have implemented in our experiments.

First, we create a well-defined model is by normalizing data, selecting features, tuning parameters, as explained in the earlier sections. We split the clippings (4-sec sliding windows) into their folds from their full-length audio. A 40-Sec audio clip, for example, would be divided into ten 4-sec clippings, each clip belonging into its fold, ensuring that clipping from the same full-length sample would never be on the same fold. Sample data points and their cross-validation fold counts are shown in Table I.

D. Data Annotation

Supervised learning tasks such as classification and regression has achieved great success in various Machine learning problem. The success of supervised learning is due to the availability of many training examples, each corresponding to an event/object. Most successful techniques, such as deep learning, require ground-truth labels for an extensive training dataset. In many tasks, however, it can be challenging to attain

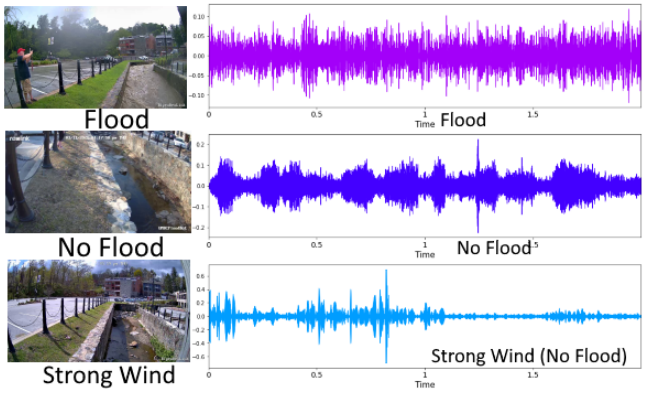

Fig. 7: Weak Label

substantial supervision information due to the high cost of the data labeling process. Thus, machine learning problems are often re-framed into weak supervision [23]. We use two types of Weak Supervision processes, incomplete supervision and inexact supervision.

1) Incomplete Supervision: For incomplete supervision, we use human annotation on a limited data set. This type of annotation is a sub-type of weak supervision called Incomplete Supervision. Incomplete supervision involves the situation where we have a small amount of labeled data and abundant unlabeled data. The labeled to unlabeled data ratio is insufficient to train a good learner.

Formally, the task is to learn $f : (\mathcal{X} \mapsto Y)$ from a training data set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_l, y_l), x_{l+1}, \ldots, x_m\}$, where there are l number of labeled training examples (i.e.those given with y_i) and $u = m - l$ number of unlabeled instances. For the experiment we manually labelled 20% of the video frames by replaying them at our lab.

2) Inexact Supervision: Inexact supervision is the process of annotating the dataset with imprecise certainty. Some supervision information is given for the inexact supervision but not as exact as desired. A typical scenario is when only

coarse-grained label information is available. For example,in the problem of sound detection, the objective is to build a model to predict the type of sound event such as heavy wind, rain, etc.

However, that same sound can also be associated with other sounds, such as a flowing river. One video frame captured by the system can have many sound event patterns, and the sound wave extracted could represent more than one event. Even with human annotation, there could be ambiguity and subjectivity. Formally, the task is to learn $f : (\mathcal{X} \mapsto Y)$ from a training data set $\mathcal{D} = \{(X_1, y_1), \ldots, (X_m, y_m)\}$ where $\mathcal{X}_i = \{\mathbf{x}_{i1}, \ldots, x_{i,m_i}\} \subseteq \mathcal{X}$ is called a bag, $x_{ij} \in \mathcal{X}\,(j \in \{1, \ldots, m_i\})$ is an instance, m_i is the number of instances in $\mathcal{X}_i$, and $y_i \in \mathcal{Y} = \{\mathcal{Y}, \mathcal{X}\}.\mathcal{X}_i$ is a positive bag, i.e. $y_i = \mathcal{Y}$ if there exists x_{ip} that is positive, while $p \in \{1, \ldots, m_i\}$ is unknown. The goal is to predict labels for unseen bags. This is called multi-instance learning [23].

We show sample output of Weak (Inexact Supervision) based data annotation example in Figure 7. We use the temporal join (time-based) in-exact supervision to bulk label our training dataset. One 4-sec frame can only belong to one sound event detection at a given time. Unable to detect that class during test/validation is regarded as a miss-classified data point.

E. Experiment Design

The experimental setup is shown in Table II. We first start with a baseline model where the objective is to detect the presence of "Gusting Wind Sound" in FloodBot captured videos. Once the audio is extracted and pre-processed as described earlier. We use the Weak Supervision to annotate the event.

TABLE II: Experiment Design

ID	Experiment	Criterion	Type
1	Wind Event	12 MPH Wind	Binary
2	Flood Event	24-H Cum.Rain	Binary
3	Wind Category	Beaufort Measures	MultiClass
4	Rain Category	24-H Cum.Rain Range	Multiclass

F. CNN Model Architecture

The network we used consists of ten convolutional layers and two fully connected layers. For hyperparameters, we chose the learning rate as 0.002, dropout as 0.5, and batch size as 16. Full CNN Layers and their input-output parameters are shown in Table III. Since we are focused on detecting the readiness and quality of data representation, we use the same CNN architecture for all experiments and focus more on experiment variations than model variations. We resize the mel-spectogram images to 128 x 431 in size and then pass the image to a CNN for image classification.

V. Results

We envision that the outcome of this model can be used to enhance flood detection capacity when other (vision-based) approaches either fail or have difficulty detecting the flood. We also argue that the model outputs could serve as a triggering event to detect or enable resource-intensive flood detection hardware and software. To that end, we designed four experiments and progressed their complexity, and evaluated their performance.

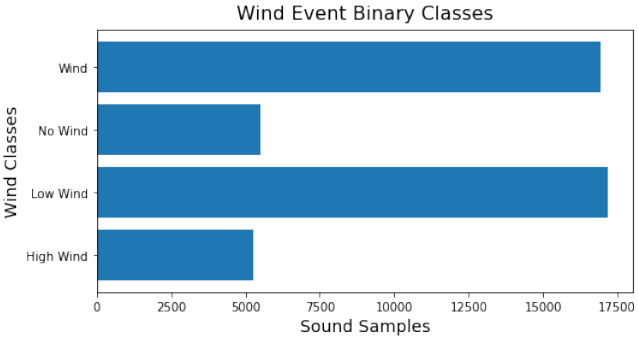

Fig. 8: Wind Event Classes

A. Binary Detection

We divided the experiments into two categories; Binary detection to detect the presence or absence of prevalence signal. We use heavy wind sound and cumulative rainfall amount to detect the wind and flood events under the binary classification experiments described below.

1) Wind Detection:

We train the model to detect high-speed wind sound in the video for this experiment. We define this experiment as a binary classification problem. The recorded wind speed varies from 0 MPH to 34.2 MPH. The mean wind speed was 7.9 while the 25% quartile is at 5 MPH and 75% (Q3) at 11.8 MPH. Therefore, we use the binary classification of wind versus no wind at a 5 MPH threshold (No Wind/Wind) to classify wind sound in the video. We also use 11 MPH as another threshold to detect the wind speed (High Wind/Low Wind). We show the classes and their distribution in Figure 8. Our goal with this experimental setup is to evaluate the preliminary natural sound event detection (wind speed) capability of our deep learning model(s), which is often directly associated with rainfall.

Layers	Output Size
Input: 1 x 128 x 431	
3×3,32,BN×2	1 × 128 x 431
3×3,32,BN×2	32 × 128 x 431
3×3,32,BN×2	64 × 64 × 215
3×3,32,BN×2	128 × 32 × 107
3×3,32,BN×2	256 x 16 x 53
Dense	500
Dropout-18	0.5
Dense	3
Output : 3	

TABLE III: CNN Architecture

We show that the model performed well in binary detection of the Wind Event. We are motivated by the fact that detection of Wind gust is an important first step in deploying the flood detection. It is also imperative that heavy rain events and thunderstorms often produces the heavy gusts and vice versa.

2) Flood Detection:

Flood Event Categorization is probably the most challenging setup from all experiments. Unlike wind speed, flooding is not an instantaneous event and often takes a prolonged period of rain to generate a flood. Therefore, we first computed the 24-hour cumulative rainfall from five-minute precipitation recorded by weather data. It is standard practice to estimate flooding events based on 24-hour cumulative rain. A threshold of 0.6 inches of recorded rainfall over 24-hour cumulative precipitation seems to be a reasonable threshold when the stream shows significant flow pattern changes to be classified into flooding versus not flooding stream. We show the results from the binary flood class in Figure 9. Next, we experiment with a binary flood detection problem. The faster the water flows, the larger the sound it makes. We correlated the sound intensity recorded to a water velocity and delineated our classes. Furthermore, the observed pattern of the sound wave during calm conditions is smooth, long, and continuous versus exhibiting high frequency and energy during fast flow flooding. The overall accuracy is 78% with a precision of 0.72% and recall 88% and F1 Score of 0.79.

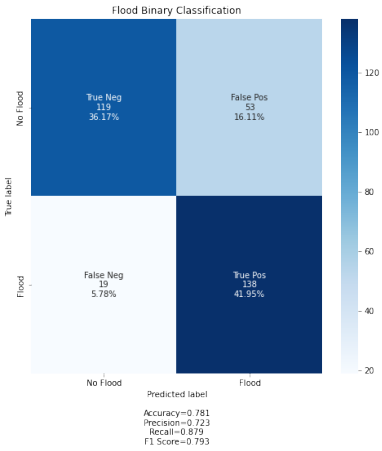

Fig. 9: Results: Binary Flood Classification

TABLE IV: Results

Experiment	Accuracy	Recall	Precision	F1 Score
Wind Event	0.72	0.76	0.78	0.77
Flood Event	0.78	0.72	0.80	0.79
Flood Multi Class	0.76	0.77	0.78	0.76
Wind Category	0.70	0.78	0.76	0.75

B. Multi class Detection

We further increase the complexity of our experiment by turning the binary classification into multi-class classification. We explore the model performance for three classes for Flood categories and four for Wind speed.

1) Multi class Flood Event:

We use similar computation and inexact supervision techniques to label our data and evaluate the flood detection models. After removing a few outliers of rain event data, we categorize the flood into three classes: Flood, Minor Flood, and No Flood. The threshold is based on 24 hours cumulative yielding no flood for less than 0.40 inches of rain, minor flood for above 0.40 inches but less than 2 inches. Above 2 inches of cumulative rain over 24 hours is categorized as flood-producing rain. After the bulk annotation, we manually verified a few hundred images and agreed with their distribution. . We

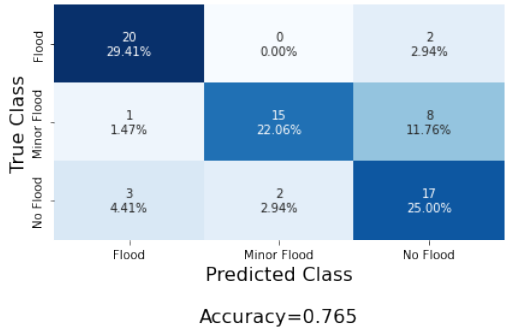

Fig. 10: Result:Flood Multi Classification

were able to achieve overall accuracy of 0.76 for three class flood detection problem and show the result in Figure 10.

Though the precision and recall are less important metrics in context of multi-class classification, we generalize the multi-class problems by summing over rows / columns of the confusion matrix. Specifically, Precision is computed as $\text{Precision}_{\ i} = \frac{M_{ii}}{\sum_j M_{ji}}$. Recall is computed as$\text{Recall}_{\ i} = \frac{M_{ii}}{\sum_j M_{ij}}$.Where i is each sample and j is total dataset.

2) Wind Speed Categorization:

Rather than identifying the windy or calm conditions, we try to classify the sound event into one of the Beaufort wind scale measures. The Beaufort wind scale measures wind speed according to the wind's impact on the land and sea. The measure developed by Sir Francis Beaufort in 1805 remains a widely used system to measure wind speed today. We experimented with all 12 Beaufort classes such as Calm, Light air, Light breeze, Gentle Breeze, etc. However, the classification was too granular for acceptable results. Therefore, we created four classes out of the range. The model achieved overall accuracy as shown in Table IV.

VI. Discussion

Table IV shows the mean classification accuracy achieved by the proposed CNN. In all experiments, proposed CNNs achieved a mean accuracy of above 0.70% and reached 0.78%, the highest for flood sound detection setup. We did not change the model architecture except reducing batch sizes if the sample sizes were smaller. We opted for the design choice to evaluate the readiness and usability of 2D representations of the audio signal as input. Our preliminary experiments find that the data set can predict the sound events on par with state-of-the-art models where the trend is still around 80% accuracy.

While there has been a great success with vision-based flood detection, the main challenge arises when the system loses visibility. Low illumination adverse weather conditions can cause visibility loss, often precursory to heavy rain and flood events. Therefore, we proposed this method to improve the flood detection probability by exploiting sound as a signal. Sound signals are also lighter in terms of data volume, thereby reducing the footprint of each deployment unit.

Motivated by the idea of Federated Learning, where data is allowed to remain on distributed devices while a central/global model is trained and shared from locally trained(on-device) model updates. Computer vision approaches and image-based solutions (Flood Detection) may have reached their limits in terms of its performance, but vision-based inference requires significant computing power, especially for real-time data-intensive applications. Flood detection is often done where there might be no internet or data transportation possibility. Cloud cost, network bandwidth, and detection time lag impede mission-critical deployment like ours.

However, running computer vision in the cloud heavily limits real-time computer vision applications. We argue that Sound Event detection can be one way to alleviate that problem without investing in computer vision deep learning resources. Furthermore, relying only on vision-based solutions restraints the detection capability to the camera angle, illumination, and image frame compositions. On the other hand, the sound-based signal is lightweight and more palatable for edge devices.

VII. Conclusion

We presented our work in environmental sound classification for flood event detection using CNN and a Mel-spectrogram. The experiment result and our field-collected real-time datasets show that the proposed experiments can achieve acceptable performance in sound event recognition. We present how the sound detection technique could be used as a supplemental trigger or precursory evaluation toolkit before deploying the relatively resource-intensive models such as vision models. The ultimate goal of our research is to deploy flood detection models into microcomputers, embedded systems and move towards edge computing. Federated learning provides us the means to reach that ultimate goal. Federated learning allows us to detect the flood on edge devices and compute the in-situ flooding condition without data transfer to the remote server or heavy computational unit.

VIII. Future Work

Our literature review found the spectrum of audio signals, particularly its temporal structure, and the two-dimensional audio representations such as the spectrogram is the most used to detect the sound event. In our future work, we shall re-evaluate the model's performance using 1-D representation or the regular wave plots. We also plan to integrate the audio model parameters into vision models and perform meta-learning to evaluate the individual data/model versus multi-modal setup, i.e., image only, audio (this paper), and image and audio together. We also limited our experiment to the same CNN model architecture. We plan to experiment with variations in model architecture to optimize their accuracy.

IX. Acknowledgment

This work is partially supported by the U.S. Army grant No. W911NF2120076.

References

[1] Daniele Barchiesi et al. "Acoustic Scene Classification". In: *CoRR* abs/1411.3715 (2014). arXiv: 1411.3715. URL: http://arxiv.org/abs/1411.3715.

[2] Jon Barker et al. "The PASCAL CHiME speech separation and recognition challenge". In: *Computer Speech Language* 27 (May 2013), pp. 621–633. DOI: 10.1016/j.csl.2012.10.004.

[3] Bipendra Basnyat, Nirmalya Roy, and Aryya Gangopadhyay. "A Flash Flood Categorization System using Scene-Text Recognition". In: *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE. 2018, pp. 147–154.

[4] Bipendra Basnyat, Nirmalya Roy, and Aryya Gangopadhyay. *Flood Detection using Semantic Segmentation and Multimodal Data Fusion*. IEEE, 2021.

[5] Bipendra Basnyat, Nirmalya Roy, and Aryya Gangopadhyay. *Towards AI Conversing: FloodBot using Deep Learning Model Stacks*. IEEE, 2020.

[6] Bipendra Basnyat et al. "Vision Powered Conversational AI for Easy Human Dialogue Systems". In: *IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS))*. IEEE. 2020.

[7] DeLiang Wang; Guy J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. 2006.

[8] Boris Defréville et al. "Automatic Recognition of Urban Sound Sources". In: vol. 2. May 2006.

[9] "Detection and Classification of Acoustic Scenes and Events: Outcome of the DCASE 2018 Challenge". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2018).

[10] Guoning Hu and DeLiang Wang. "Auditory Segmentation Based on Onset and Offset Analysis". In: *Audio, Speech, and Language Processing, IEEE Transactions on* 15 (Mar. 2007), pp. 396–405. DOI: 10.1109/TASL.2006.881700.

[11] Kazi Aminul Islam et al. "Flood Detection Using Multi-Modal and Multi-Temporal Images: A Comparative Study". In: *Remote Sensing* 12.15 (2020). ISSN: 2072-4292. URL: https://www.mdpi.com/2072-4292/12/15/2455.

[12] Stephen McAdams and Emmanuel Bigand. *Thinking in Sound: The Cognitive Psychology of Human Audition*. 1993.

[13] Annamaria Mesaros et al. "Acoustic event detection in real life recordings". In: (2010), pp. 1267–1271.

[14] Maria E Niessen, Leendert Van Maanen, and Tjeerd C Andringa. "Disambiguating sound through context". In: *International Journal of Semantic Computing* 2.03 (2008), pp. 327–341.

[15] Karol J. Piczak. "ESC: Dataset for Environmental Sound Classification". In: *Proceedings of the 23rd ACM International Conference on Multimedia*. 2015, pp. 1015–1018. ISBN: 9781450334594. DOI: 10.1145/2733373.2806390. URL: https://doi.org/10.1145/2733373.2806390.

[16] Alain Rakotomamonjy and Gilles Gasso. "Histogram of Gradients of Time–Frequency Representations for Audio Scene Classification". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1 (2015), pp. 142–153. DOI: 10.1109/TASLP.2014.2375575.

[17] Ajay Divakaran Regunathan Radhakrishnan and Paris Smaragdis. "AUDIO ANALYSIS FOR SURVEILLANCE APPLICATIONS". In: *Workshop on Applications of Signal Processing to Audio and Acoustics* (2005).

[18] Nitin Sawhney and Pattie Maes. "Situational Awareness from Environmental Sounds". In: (Dec. 1998).

[19] Jivitesh Sharma, Ole-Christoffer Granmo, and Morten Goodwin. *Environment Sound Classification using Multiple Feature Channels and Attention based Deep Convolutional Neural Network*. 2020. arXiv: 1908.11219 [cs.SD].

[20] Neha Singh et al. *Flood Detection Framework Fusing the Physical Sensing and Social Sensing*. IEEE, 2020.

[21] S. Yun et al. *"Discriminative training of GMM parameters for audio scene classification*. en. 2016.

[22] Zhichao Zhang et al. "Deep Convolutional Neural Network with Mixup for Environmental Sound Classification". In: Nov. 2018, pp. 356–367. ISBN: 978-3-030-03334-7. DOI: 10.1007/978-3-030-03335-4_31.

[23] Zhi-Hua Zhou. "A brief introduction to weakly supervised learning". In: *National Science Review* 5.1 (Aug. 2017), pp. 44–53. ISSN: 2095-5138. DOI: 10.1093/nsr/nwx106. eprint: https://academic.oup.com/nsr/article-pdf/5/1/44/31567770/nwx106.pdf. URL: https://doi.org/10.1093/nsr/nwx106.