

RESEARCH ARTICLE

DX-FloodLine: End-To-End Deep Explainable Pipeline for Real Time Flood Scene Object Detection From Multimedia Images

NUSHRAT HUMAIRA¹, VIDYA S. SAMADI², AND NINA C. HUBIG³¹School of Computing, Clemson University, Clemson, SC 29634, USA²Department of Agricultural Sciences, Clemson University, Clemson, SC 29634, USA³School of Computing, Clemson University, Charleston, SC 29407, USA

Corresponding author: Nushrat Humaira (nhumair@clemson.edu)

ABSTRACT In recent years, an alarming number of natural disasters, specifically hurricanes and floods have affected the United States. Flooding is a global crisis, however we still do not have an automated approach for real-time flood risk detection and mitigation. Detecting inundation in urban areas and road segments is crucial for vehicle routing and traffic management. Unlike remote sensing images that have been used in past studies, it is better to mine social media data and geo-tagged images or videos for flood estimation in near real time. In this paper, multimedia content was collected from social media streaming services, primarily Twitter for analysis. We propose DX-FloodLine, an interpretable, intelligent, multi-stage, end-to-end, deep neural network-based pipeline to classify near real time emerging flood occurrence from images and detect submerged objects and pedestrians around flooded regions. We introduce a novel hybrid neural network for flood detection, a VGG16(Visual Geometry Group 16-layer model)-LSTM (Long short term memory) ensemble. Our novel ensemble architecture recognizes flood in images in the first stage of pipeline, to be later passed on to second stage for object detection. We applied interpretable models to flood classifiers to identify model shortcomings and then incrementally train for continuous improvement. DX-FloodLine was deployed and tested on unseen and near real time streamed flood images. Our VGG16-LSTM flood recognition model achieved around 90% validation accuracy on multiple benchmark studies and surpassed other competitors by a good margin.

INDEX TERMS Flood detection, model uncertainty, object detection, social media data, VGG16-LSTM.

I. INTRODUCTION

Flooding events can cause loss of life and significant destruction to properties and critical infrastructure. These effects are often easily recognized and frequently reported in the media. Recently, natural flood surveillance and detection has been a prominent research interest. Efficient and accurate flood occurrence detection is critical for authorities to control floods and manage flooding impacts. Furthermore, it is crucial to alert authorities and first responders about at-risk areas to flooding for real time decision making and planning. Various methods have been used to monitor and estimate flood magnitude, and a recently popularized

approach for flood monitoring and assessment is the use of crowd sourced data. Multimedia images and videos from news outlets, and social media posts with geo-location tags, known as Volunteered Geographic Information (VGI), can provide valuable insights into flood situations occurring in real time. Extracting publicly shared content and user metadata, neighborhoods can be narrowed down to precise locations where the inundation is impending. However, labeling and extracting the voluminous and unfiltered images retrieved continuously through online sources, reminds of the defining factors of working with big data, i.e volume, velocity and variety. Deep Learning (DL) models such as the Convolutional Neural Network (CNN) [1] have proved to be proficient in extracting meaningful features and detecting flood labels accurately from the massive number of social

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo¹.

media data generated in real time or after the event. As there are different combinations of layers and structure of neural networks for different tasks, it makes sense to incorporate the best group of networks in a pipeline to serve the purpose. An automated pipeline that is equipped with big data analytics, data engineering and deep neural network to assess risk concerning locations and livelihood based on tweets collected near real time flood situations, certainly would benefit both stakeholders and civilians. We identified few research gaps in prior flood detection pipelines that are relevant to the objective of this paper. Donratanapat et al. [2] proposed flood informatics pipeline named FAIS, Flood analytics information system, which analyzes data collected during historical and real time events, through web platform, river level and discharge from USGS (United states geological survey) and tweets from Twitter, and visualizes flood impacted areas in a web application. The FAIS application detected roads, bridges from flooded areas and provided flood frequency metrics with peak prediction. Techniques proposed by FAIS acted as starting point to develop our pipeline where we sourced images from web and social media platforms to perform image analysis to assess flood risks. Unlike FAIS, we split the flood recognition task and flooded scene object detection task into two separate modules, and provide near real time inference with model incrementally trained with incoming data. To improve the model performance, we added interpretable attribute visualization methods in the pipeline. In [3], Baker et al. presents similar pipeline linking real-time river level information and social media to aid national level flood management for Britain. The major difference between this pipeline and DX-Floodline is that they focused on analyzing and classifying tweets with natural language processing tools and did not utilize multimedia images for visual assessment of flood situations. Focusing on these shortcomings, this paper proposes DX-FloodLine, an explainable, deep learning based multi-stage, end-to-end pipeline to estimate real time emerging flood severity and detect objects submerged in flooded areas. Multiple modules embedded in the proposed pipeline are designated for specific tasks that attempt to address several limitations of above mentioned prior works. Overall, the objectives of this paper are to: (i) present an end-to-end flood monitoring pipeline to deliver near real-time response, (ii) propose a novel hybrid ensemble method called the VGG16-LSTM for flood image classification, (ii) construct a custom labeled image database consisting more than 9000 images in addition to using MNIST [4], CIFAR-10 [5] and STL-10 [6] for evaluation, (iii) implement an ensemble of object detection models to detect various categories of roadside and urban objects in the images such as vehicles, pedestrians, buildings, bridges, etc., (iv) offer clarity and explainability using attribute-based methods applied on the flood image classification model when the model outcomes veer off the expected direction. This paper is organized as follows: Section II describes related work. In section III, a detailed description of the

end-to-end pipeline, its workflow, and the structure of individual modules are discussed. Section IV presents the performance evaluation of our flood event detection architecture against benchmark datasets, urban flooded scene object detection, interpretable visualization of classifier attributes, and uncertainty assessment of trained models. Finally, the discussion and future work of this study is presented in section V.

II. RELATED WORK

In order to create a baseline of systems closest to the desired pipeline, we referenced publications focused on urban area flooding [7], near real time solutions [8], social media data analysis for flood informatics [9], flood water level estimation [10], [11], [12] or pipeline processes including retrieval and detection [2], [13]. We included references that was used as baseline and state-of-the-art methods in the development of DX-Floodline. As our pipeline has multiple modules, it was crucial to look into frameworks for social media data based flood analytics, flood recognition from image attributes, textual and geographical information. Moutmzidou et al. [7] proposed a framework that streams real time social media data for relevant contents using which the model analyzes both visual and textual features to detect flooding events. The support vector machine classifier learned visual features whereas the random forest classifier was trained on textual features. Classifier results were evaluated in dual mode, once sequentially and then once simultaneously. In the first case, textual classifier results were examined after visual classifier results to determine event relevancy, while in the second case both classifiers produced the same outcome at the same time. Instead of visual features, Bruijin et al. [9] extracted contextual hydrological information such as precipitation, rainfall from the timestamp and location of the tweets and both hydrological and tweet text features were passed through a multi-modal neural network to finally reach a uniform decision. Fitriannah et al. [8] extracted spatial information from user updates on a flood event and put the location coordinates on map near real time to communicate developing flood situations to users. Lin et al. [14] presented image-based volunteered geographic information (VGI) that were obtained from smartphone cameras to determine the flood water levels based on photogrammetric principles. Random forest classifier model was applied to highlight flooding regions and Canny edge detection method was used to detect the flooding line of the classified image based VGI. Both Feng et al. [15] and Quan et al. [16] predicted flood level by estimating human activity on water related scenes crawled from social media. Human pose information was gleaned using a binary mask processed through Mask-RCNN by He et al. [17]. Chawdhury et al. [18] applied a similar mask based detection approach by tracking submerged portion of objects in images taken from social media in real time. Features were extracted by combining water level regression with a relative ranking of image as pairs. Water level regression involved supervised learning while pairwise

water level ranking was weakly supervised. Singh et al. [19] classified localized flood related tweets using natural language model based transfer learning. Opella et al. [20] primarily used high resolution topological spatial imagery and other geographic information such as wetness index, contour details, soil quality to map potential flood prone areas using a combination of convolutional neural network and support vector machine. Lohumi et al. [21] focused on gathering short videos that captured flooding incidents and categorized them based on flood severity and damage inflicted on local infrastructure. Base CNN model extracts fixed length representation from video frames and a GRU (Gated Recurrent Unit) encodes the dependencies from the sequences of those representations at different time steps. Final prediction determines the severity of the flood.

III. DX-FLOODLINE

In the proposed pipeline there are two main phases, one for training and the other for creating the inference in real time. Within training phase, there are two sub-task modules, one for flood event classification and another for scene object detection. For the flood event classification task, we proposed a hybrid VGG16-LSTM ensemble (VGG 16-Long Short-Term Memory) model to classify sequences of frames from archived images and videos on the relevance of flood event. After this task, images would be classified as relevant or not relevant to developing flood situations and propagated as input to the object detection task. Object detection models predict the chosen categories of objects found in the image along with prediction confidence score. We focused on detecting pedestrians and common objects such as traffic signs and vehicles or houses found along roads and highways. Further augmentation on the detection model segments irregular shaped object such as bridge and flyover from the background and estimates bounding area surrounding those objects. The inference module is set to infer predictions on unseen flood images by deploying saved models from the training phase. For inference purpose, multimedia images and videos obtained through any web streaming application programming interface (API) or social media for particular locations were propagated through the pipeline. Final prediction results are displayed to the user with a bounding box around objects and corresponding confidence score on the predicted object label. Fig. 1 illustrated a schematic workflow of our proposed DX-FloodLine. As shown, images were collected and aggregated from multiple source and fed into the pipeline. We evaluated modeling performances on data collected from major hurricane events in the Southern USA, i.e., Hurricanes Harvey and Florence, that occurred in 2017 and 2018.

A. FLOOD EVENT CLASSIFICATION

Two categories of ensemble neural architecture with different predictor network were created for the flood classification task. In the first category, pre-trained convolutional networks were utilized as feature extractors and multi-layer fully

connected network as predictors. For the second category, same feature extractor was connected to the LSTM subnetwork so it can interpret those features as a sequence across consecutive time steps. VGG-Net [22] achieved state-of-the-art image classification results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [23]. In this research, we selected a 16 layers version of VGG-net pre-trained on the ImageNet dataset as feature extractor. Fine tuning of pre-trained network needs to be balanced properly to avoid overfitting or missing useful information. The lower layers in convolutional network extract generalized features while upper layers extract specific features. From the convolutional base of the VGG-16 net, we kept most lower layers up to the final maxpooling layers frozen and trained upper layers. We maintained a small learning rate to decrease the risk of losing knowledge previously learned by the model. Predictor networks for first category are listed below:

- **Fully Connected dense layers:** We implemented a stack of fully connected layers, followed by a softmax activated layer [24] to output the probability distribution over class labels and classifies the image to the most probable class.
- **Global average pooling:** We replaced the dense layers with a global average pooling (GAP) layer and kept the softmax activated layer as it is. GAP layers are similar to max pooling layer, but their usefulness lies in their ability to reduce spatial dimensions and number of parameters.

The second architecture was a hybrid combination where we used a convolutional neural network as base for feature extraction and recurrent upper layers. We started off with a simple, two block convolutional base to see how it works in tandem with the recurrent network. Each convolutional block was followed by ReLu activation [25] and a max pooling layer. ReLu activation has been used by many state-of-the-art such as [26]. Output of max pooling aggregates the feature maps to be reshaped, and then transferred it to the LSTM sub-network for modeling sequential variation of image properties over time. Distinct characteristics of LSTM network enables learning particular pattern observed over successive periods of time. So, when all images in the set are repeatedly processed and reshaped to flattened form, resultant is set of spatial features over time which is to be handled by the LSTM network. For this model we used 64 nodes for the LSTM layer construction. Alternatively we considered GRU (Gated recurrent unit) for sub-network as well, however the performance difference was nominal so we kept LSTM network for simplicity. Finally, a flatten layer and softmax layer ends as last layer to output classification labels. We replaced above mentioned shallower convolutional base with VGG-16 pre-trained on ImageNet while keeping the recurrent LSTM subnetwork same. The pretrained VGG16 base captured patterns in sequences of images from developing flood videos which was learned through recurrent LSTM layers over consecutive time steps. Proposed hybrid

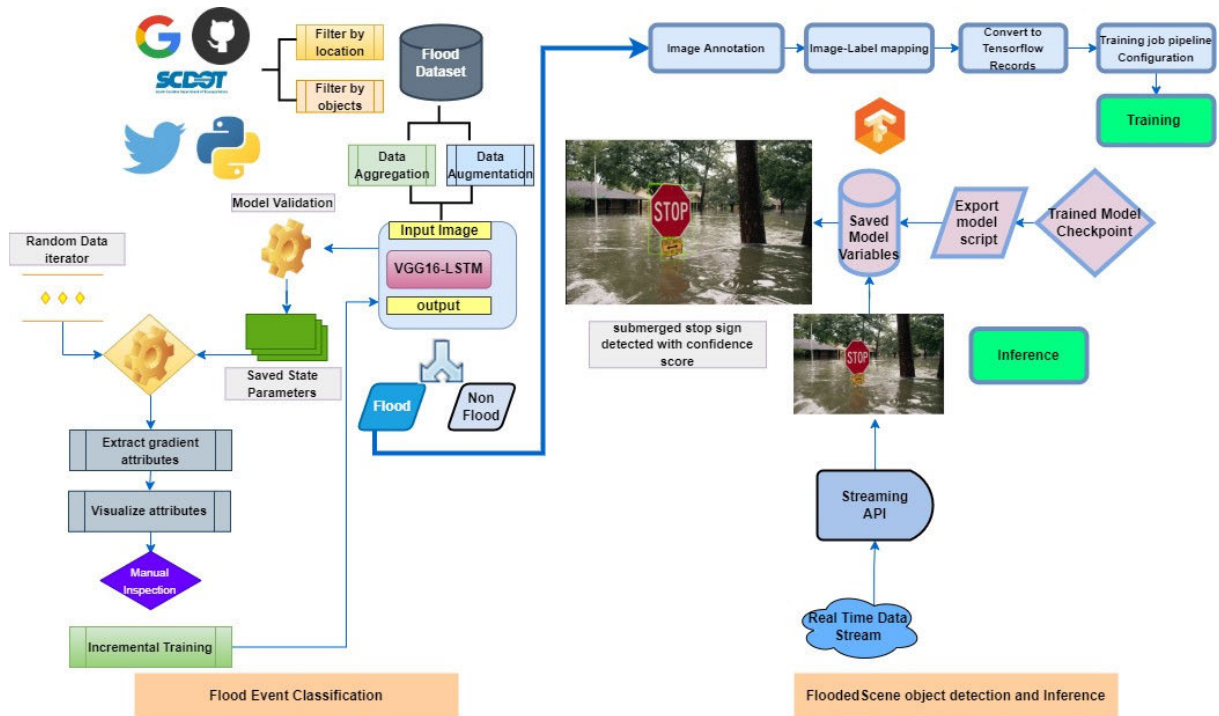


FIGURE 1. DX-FloodLine: near real time end-to-end flood detection pipeline with explainer module that discovers evidence of flooding and submerged objects from images collected primarily from Twitter and web sources. Images are pooled based on their geographic location. Training module is separated into flood event classification and flooded scene object detection task with intermediate explainer module.

VGG16-LSTM model is completely illustrated in Fig. 2. We kept VGG-16 network layers frozen and only trained LSTM layers.

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} \quad (1)$$

$$\mathcal{L}(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{box}} \sum_i p_i^* \cdot L_1^{smooth}(t_i - t_i^*) \quad (2)$$

B. ROADSIDE SCENE OBJECT DETECTION

Next, we discuss the scene object detection task from flood images which was developed using the Tensorflow object detection API. We explored two state-of-the-art CNN based framework for object detection, YOLO [27] and Region proposal classification networks. While RPN performs detection in two stages, extraction of the region proposal and prediction on the region proposal on various regions of the image, YOLO completes the prediction in one pass. YOLO is considered to be faster and more accurate which will benefit real time performance. Despite those benefits, we employed RPN models for few reasons. Our primary goal is to detect objects common in urban areas e.g, vehicles, pedestrians, trees and properties like house, bridge. Differences in object shapes, structures and properties required specific consideration thus we chose two types

of models, the faster R-CNN [28] and the mask R-CNN [17]. Both models use region proposal network for the first stage to extract candidate bounding boxes around regions of interest for each object in the image. Thus it was possible to utilize RoI extraction phase between two RCNN models in our pipeline to cover for all objects as an additional benefit and reduced computational load. With Mask R-CNN, it was possible to detect polygonal shaped objects like bridges and buildings. In future, we plan to use shapes to estimate the depth of submersion for the infrastructure. TensorFlow API offers building blocks to create custom object detection models, with the key notion of objects being recognized as regional bounding box out of the whole image. Thus, input images need to have a corresponding annotation file that have information on respective location of object, category and so on. Each object category is given a label to be an integer encoded. All images, annotations, mapped category label are serialized into TensorFlow record format. TensorFlow requires training jobs to be configured properly and satisfy some existing conditions. Those requirements include choice of a pre-trained model to fine tune a model selected from available models, options to resize images, regularize and set batch size to match the memory available for execution. Tensorflow object detection model zoo provides a wide collection of pretrained R-CNN models with different ResNet backbone trained on COCO 2017 dataset of images in varying resolution. This factor also solidified our decision for employing R-CNN based object detection models. Model

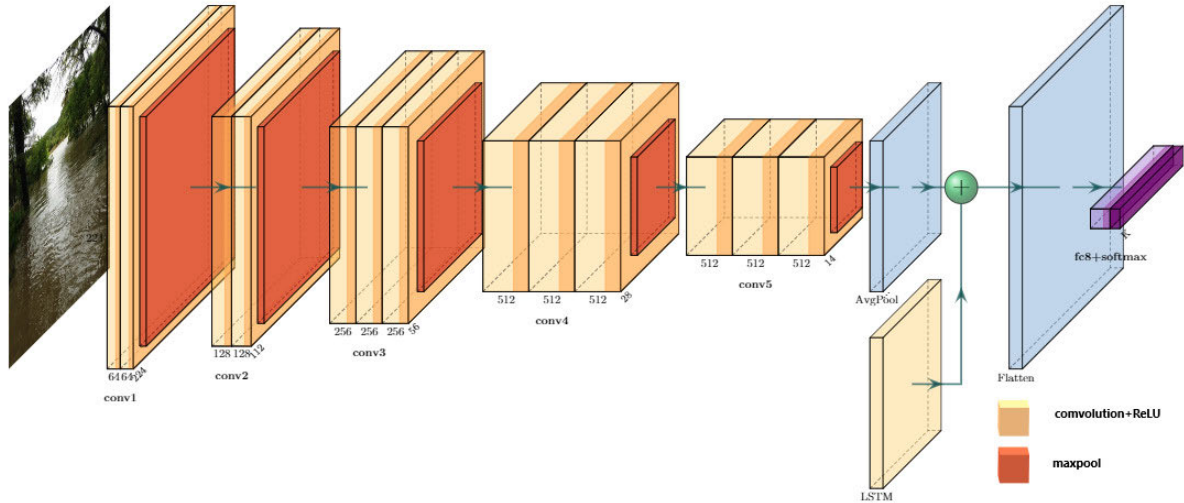


FIGURE 2. Proposed hybrid VGG16-LSTM architecture for flood image classification. Light orange rectangular blocks in the lower layers shows convolutional blocks with increasing kernel size (64,128,256,512), darker orange blocks are max-pooling layers. Final max-pooling layer of the extraction base is connected to average pooling layer, then passed on to the LSTM layers and final classification layer.

results can be evaluated using detection evaluation metrics used by the COCO object detection challenge [29]. Finally, when training job was complete and evaluation seems satisfactory, we exported the newly trained inference graph, which was later used to perform the object detection during run-time inference on unseen images. Ensemble of models we applied for scene object detection are described in more details as follows.

1) FASTER R-CNN:

Faster R-CNN [28] network belongs to region proposal-based object detection algorithm family. This network can be considered as an object detection pipeline where first stage, region proposal network (RPN) generates candidate variable sized bounding boxes or object RoI (region of interest) proposals to be passed on to Fast R-CNN [30] for final detection results in the second stage. Object bounding boxes produced by RPN are also known as Anchors and they are ranked by their likelihood of containing certain objects. RPN attempts to correct bounding box prediction error by localized regression which minimizes SSE loss as $\mathcal{L}_{reg} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2$ where predicted bounding box coordinates $\mathbf{p} = (p_x, p_y, p_w, p_h)$. Regressor learns correction function $d_i(\mathbf{p})$ where $i \in \{x, y, w, h\}$ and target function $t_x = (g_x - p_x)/p_w, t_y = (g_y - p_y)/p_h, t_w = \log(g_w/p_w), t_h = \log(g_h/p_h)$. Fast-RCNN improves training convergence speed by aggregating feature vectors from all region proposals into one shared matrix. Now Faster RCNN uses RPN for region proposal task and fast R-CNN for object detection using those proposals. Optimization problem changes to multi-task loss function as combination of the losses of classification and bounding box regression as defined in (1) and (2). Here \mathcal{L}_{cls} is the log loss function

over two classes, as we can easily translate a multi-class classification into a binary classification by predicting a sample being a target object versus not. L_1^{smooth} is the smooth L1 loss. \mathcal{L}_{cls} is defined in (3):

$$\mathcal{L}_{cls}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i) \quad (3)$$

2) MASK R-CNN:

This algorithm by He et al. [17] extends Faster R-CNN by adding parallel module for segmenting different object instance as a mask from background while keeping the original bounding box and class prediction module as it is. Segmentation module matches objects by its mask shape and other factors by pixel-to-pixel after region of interests (ROI) have been identified by the bounding box regression module. As there are multiple tasks being executed in parallel for the second stage of Mask R-CNN, a multitask loss $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask}$. \mathcal{L}_{mask} is the new loss which can be a typical binary cross entropy loss. Mask segmentation module encodes K number of binary masks, each of size m by m for K number of classes to represent object’s spatial layout. This module outputs of Km^2 size and a sigmoid function applied on top of pixels of this output.

C. INTERPRETATION OF FLOOD IMAGE CLASSIFICATION THROUGH ATTRIBUTE VISUALIZATIONS

Deep neural models continue to be more complex and less transparent behind the scenes and the outcome they produce. Thus, model understanding became an active area of innovation for practical intelligent systems deployed across many industries. Attempt to explain a model’s decision can be achieved by investigating input features that contribute to certain decision. In our pipeline, we added interpretability methods as intermediate phase after the flood event

classifier and before the scene object detector. With attribute visualization methods for interpretability, we discovered image pixels around houses or cars submerged in water, often influenced the model to classify it as flood relevant image. In this phase, mis-classification were caught early on and helped for model improvement intuitively. With Captum [31] library for Pytorch, we used gradient based attribution algorithms to understand and explain flood image classification. Attribution algorithms used for this paper were Saliency map, integrated gradients [32], Deep Lift (Deep Lift Important Features) [33]. It can be noted that assigning attribution scores to input features is something all algorithms have in common. Sign of the attribution score, positive and negative means whether a particular input element contributed positively or negatively to the final model prediction.

IV. EXPERIMENTS AND EVALUATION

As we built DX-FloodLine module by module, we continued to run performance evaluation experiments for each module in the pipeline. Integrated pipeline was also tested to understand how all modules together perform for near real time inference. In this section, we showcase the conducted experiments, starting from data pre-processing and collection to final object detection.

A. DATA COLLECTION

We sourced flooding images from Disaster Image Retrieval Social Media (DIRSM) dataset that was published for proceedings by MediaEval Benchmarking Initiative for Multimedia Evaluation [34]. We also used Central European Flood 2013 dataset [35], mined multimedia tweets, image search engines (Google), and Github repositories to provide additional images from past flooding events. Publicly available traffic cam images from SCDOT (South Carolina Department of Transportation) were also included in the set. These images were labeled manually as flood or non-flood, based on discriminative features often present in scenarios related to urban and inland flooding. The criteria to differentiate between floods and heavy rains was decided based on certain features such as water clogged areas, submerged roads and buildings, cars, and humans. Tweets were streamed using the *Tweepy* python library, API version 1.1. We used developer credentials to authenticate and access the tweets. Since the time of data collection, Twitter API version 2.0 has been released while updating the data access policy for academic researchers and developers. The flexibility of our pipeline design ensures that we can replace the tweepy data collection module scripts with updated API methods without compromising the rest of the modules' stability. Twitter API queries were filtered with geocode (i.e., latitude/longitude) of places within the US with past and frequent flood occurrences. Gathering live tweets real-time during flood or hurricane was difficult at times due to the limited availability of historical tweets from only past 7 days. Also only about 10-15% of the collected



FIGURE 3. Demonstration of image augmentation as (a) randomly cropped and (b) random black patches overlaid.

tweets had images attached to them. Regular twitter users shared inundated indoor scenes along with outdoor scenes, while media photographers shared outdoor scenes. Thus we broadened the search to archived tweets and grouped them by flood or hurricane names or hashtags associated with disaster such as road/bridge damage, shelter, emergency, evacuation. Minimum of 1000 tweets each from several flood events were collected which were filtered manually based on the inclusion of certain objects such as houses, cars, or trees, in water bodies and inundation areas. To have balanced training set, we included random non-flood images as well. Merging all collected images, we have built more than 9000 flood event images.

B. DATA PRE-PROCESSING AND AUGMENTATION

As images and videos were collected from various sources as shown before, image enhancement methods were performed to create uniform image feature distribution. Contrast of the images and extracted video frames were improved with adaptive gamma correction algorithm by Rahman et al. [36], Cao et al. [37]. By varying the gamma value we modulated the contrast for uneven brightness in images and countered white exposure. We performed geometric translation to images such as translation, rotation, and affine transformation to incorporate variety. Other augmentations were applied in the later stage of object detection model training. Using Tensorflow object detection API, we applied enhancements such as crop, padding, overlapping black patches, horizontal flip, and rotation all in a randomized manner. Fig. 3 demonstrates augmentation by crop in (a) and overlaid black patches in (b). Doing so increased the number of training images and prevented overfitting. As mentioned in previous section, images for first phase of flood recognition task were labeled as flood and non-flood, manually. Object detection models based on region proposal network rely on predetermined bounding boxes around objects of specific categories to learn the features of objects and quality of annotations. With incredible number of images in our dataset, manual annotations were both time consuming and required more attention to achieve precision since the number of instances of objects in the images is quite large. Specially, for polygonal shaped objects like bridges that needed to be annotated for Mask R-CNN to train, the level of precision required was higher. To lessen the workload, we shifted

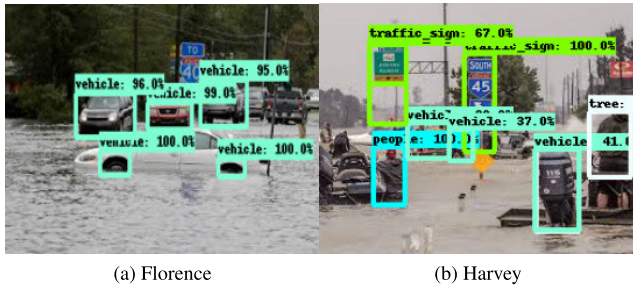


FIGURE 4. Left image(a) is taken during Hurricane Florence, North Carolina and the right(b) during Hurricane Harvey, Texas. Both images contain people, vehicle, traffic sign, tree that are detected using DX-FloodLine along with the confidence score of detection.

TABLE 1. Performance comparison: Flood event classification.

Model name	Performance Criterion ⁺			
	Accuracy(%)	F1-score	ROC-AUC score	NMI
VGG16-Dense	89	91.827	0.92	0.56
VGG16-GAP	85.29	80.157	0.90	0.5
CNN-LSTM	86.23627	89.265	0.87	0.5
VGG16-LSTM	90.2	87.815	0.93	0.73
ViT	79.91	82.35	0.82	0.33

⁺ best performances are highlighted in bold

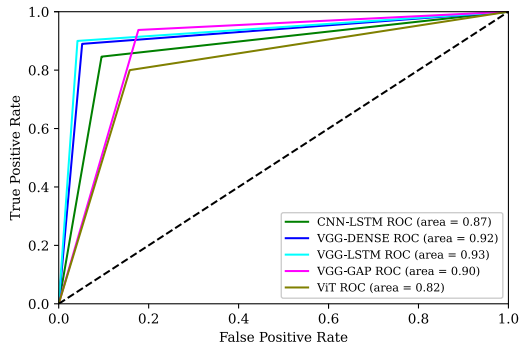


FIGURE 5. ROC-AUC curve comparison on all flood image classification models, legend shows the area under each curve which puts VGG-LSTM ahead of the rest of models.

some portion of object annotation task to pre-trained object detection model from Tensorflow model zoo.

C. FLOOD CASE STUDIES

Two case studies are paramount for our study a) hurricane Harvey and b) Hurricane Florence - both of them occurring in the USA. Hurricane Harvey made landfall in Texas on Aug. 25, 2017, as a Category 4 hurricane, that caused unprecedented flooding around Houston, Texas. Around September 14, 2018, Hurricane Florence made landfall near Wilmington, North Carolina which led to record floods in North and South Carolina. We used Hurricanes Harvey and Florence driven flooding images for segmentation and flood label detection. In fig. 4 we can see many objects detected around flooded area caused by Hurricanes Florence (a) and Harvey (b), respectively.

D. PREDICTION EVALUATION

The baseline of classification models for the flood event classification task on images and corresponding performance

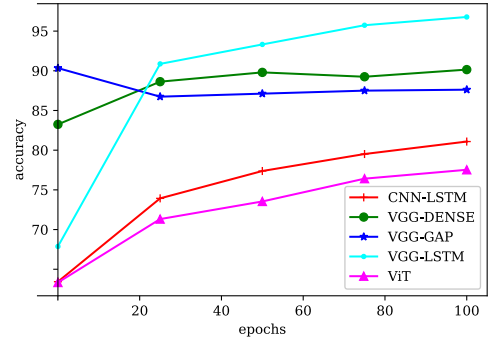


FIGURE 6. Comparative analysis of average validation accuracy of four flood image classification models proposed in this paper and vision-transformer. As noted, VGG16-LSTM outperformed other four models, including state-of-the-art ViT model in terms of validation accuracy by good margin. For this task of flood classification from sequence of images, VGG16-LSTM model offered the best fit.

metrics is shown in Table. 1. We used four classification quality metrics for comparison to ensure different angles of the method: accuracy, F1-score - the harmonic mean of precision and recall, ROC-AUC for visualization quality score [38] and Normalized Mutual Information (NMI) [39] an information-theoretic metric, for evaluation. Among our four novel ensemble models that were invented and implemented in this paper, three models use VGG-16 as the base feature extraction layer and different combinations of upper layers for classification, thus they were named, accordingly as VGG-Dense, VGG-GAP and VGG-LSTM. Dense indicates a two-layer dense neural network while GAP is a global average pooling layer. The fourth model, CNN-LSTM is a combination of convolutional ANN and a LSTM network which was implemented to showcase how a shallower, convolutional feature extraction network fare in comparison to deeper networks. To deliver a thorough performance evaluation, we brought a state-of-the-art, Vision Transformer (ViT) model [40], [41] against the four models previously mentioned. Inspired by the Transformer architecture [42] for Natural Language Processing (NLP) tasks, Vision Transformer was created to give attention to specific pixels in images, to break the limitation of standard convolutional models and learn sparsely distributed features of images. All models were trained for 100 epochs with learning rate of $1e-4$ and batch size of 64. The train-test split was done in a 70 – 30 ratio. Training and test set images were normalized before being fed into the network. ROC-AUC curve for the VGG16-LSTM architecture and all competitor models mentioned before are displayed in Fig. 5, with Area Under Curve (auc) score for each curve. Overall, VGG16-LSTM reached our expectation by achieving the highest accuracy, ROC-AUC score and Normalized Mutual Information (NMI) score.

E. EXECUTION LATENCY EVALUATION

We conducted several experiments to quantify the effectiveness and latency of flood event classification models during

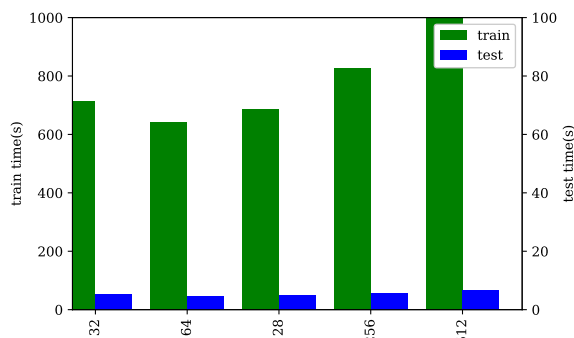


FIGURE 7. Execution time for training and testing/inference phase of flood recognition model. Time was measured in seconds (y-axis) plotted against images in different scale (x-axis). We performed this experiment to test whether image size affects execution time of model as it may increase latency of inference module for near real time inference. Results show no significant change in execution time and ensures quick response during training and inference both.

inference. We calculated average validation accuracy over every 25th epoch for each of the models and visualized them in Fig. 6 to understand the differences in performances. All models were trained and tested on Palmetto supercomputer at Clemson University. Multiple NVIDIA GPU with 56 gbps infiniband interconnect, 32 cores and 126 GB RAM were selected at each complete run of the pipeline. Models were synchronized on the GPU and we measured execution time for train and test or inference phase as shown in plot in Fig. 7. Image sizes were varied starting from 32 by 32 pixels and going exponentially up to 512 by 512 pixels. Time during inference was significantly lower and low-resolution images took less time in both phases without any negative impact on accuracy. Again we observed that VGG16-LSTM maintained consistent performance over others and thus we demonstrate latency evaluation only for this model. VGG16-LSTM achieved 90% average validation accuracy on our flood image dataset and three other benchmark datasets, namely MNIST, CIFAR-10 and STL-10. MNIST [4] has a training set of 60,000 images and test set of 10,000 images. CIFAR-10 dataset [5] consists of 60000 colored images in 10 categories and 32 by 32 pixel size, with 6000 images per category. There are 50000 training images and 10000 test images. STL-10 [6] is modified from CIFAR-10 to introduce large number of unlabeled images and higher resolution. Dataset contains 500 training images (that has 10 pre-defined folds), 800 test images for each of the 10 classes and 100000 unlabeled images.

F. BENCHMARKING ARCHITECTURE ON MULTIPLE IMAGE DATASETS

We conducted a benchmark study to compare the performance of the proposed VGG16-LSTM architecture for classification over our customized flood image dataset and other publicly available datasets such as CIFAR10, MNIST, STL10. We compared VGG16-LSTM against models that

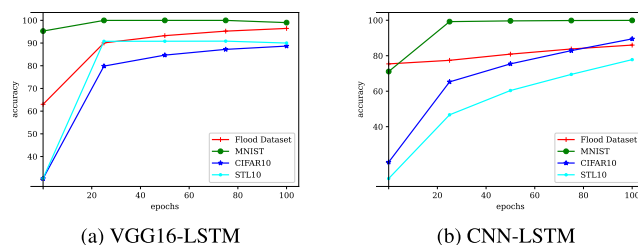


FIGURE 8. Comparative benchmark results of VGG16-LSTM (left) and CNN-LSTM architecture (right). We maintained same experimental configuration for both architecture and ran validation on four datasets, our collected flood image dataset, MNIST, CIFAR10, STL10. As evident from figures side by side, deeper feature extraction base in VGG16-LSTM gives better accuracy.

we built prior to VGG16-LSTM, namely VGG16-DENSE, VGG-GAP, CNN-LSTM and state-of-the-art, Vision Transformer, ViT. All datasets were partially downsampled to bring down the dataset size as to deliver fair results. Benchmark results are illustrated in Fig. 8. To demonstrate the difference of using shallower vs. deeper convolutional feature extraction base of classification model, we put both CNN-LSTM and VGG16-LSTM benchmark results side by side in (a) and (b). It is apparent that both models perform best using the MNIST dataset. However, keeping moderate and consistent performance for our custom flood dataset, we can declare with strong conviction that our VGG16-LSTM model with deeper feature extraction base and pretrained weights are able to yield better result compared to the CNN-LSTM across all 4 datasets.

G. INTERPRETABLE VISUALIZATION FOR FLOOD IMAGE CLASSIFIER

Gradient attributes derived from three attribution models are illustrated in Fig. 9, in the clockwise order. We demonstrate the original image in (a), then the image gradients extracted and visualized by gradient magnitudes in (b), the integrated gradients in (c), and DeepLift in (d). Visualizations highlight the gradients VGG16-LSTM model used to decide image being related to flood. As we can see, the model focused learning pixels around the water boundaries and in few places of the blue skyline.

H. FLOOD SCENE ROADSIDE OBJECT DETECTION

With the help of Tenorflow object detection API, we implemented custom object detection models for roadside object detection. Ensembles of two models, one based on the Faster R-CNN and one based on the Mask R-CNN were custom implemented and installed in the object detection module of DX-FloodLine. Images labeled as flood related by the VGG16-LSTM and corresponding annotations for objects inside those images are fed into both models. Faster R-CNN being a region proposal network-based model required rectangular shaped bounding box annotation of objects while Mask RCNN works with custom polygonal shape binary mask of objects.

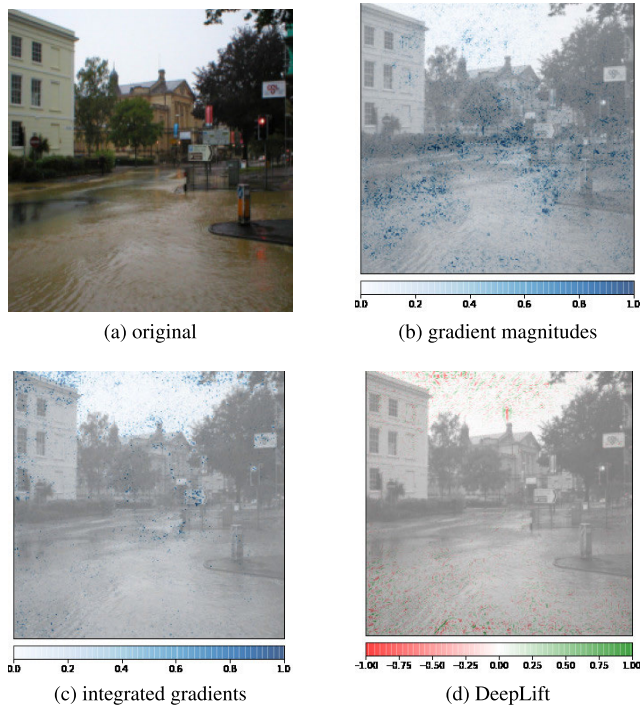


FIGURE 9. (a) Original image with houses submerged in water (b) image overlaid with gradient magnitudes to show gradients around submerged region (c) Image overlaid with attributed visualized by integrated gradients and smoothgrad squared method (d) Image overlaid with attribute visualized by DeepLift method.

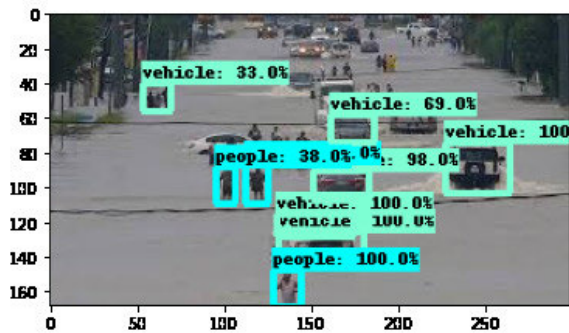


FIGURE 10. Different vehicles and pedestrians detected with bounding box with estimated confidence score.

1) COMMON SHAPE ROADSIDE OBJECT DETECTION WITH FASTER RCNN

As illustrated in Fig. 10, vehicles and people were detected within a rectangular bounding box with the confidence score on the precision of the detected object category. We observed that clearer objects have higher confidence scores while some of the object being detected incorrectly albeit with a significantly lower confidence score to indicate that those should be disregarded for final inference. The detection scores varied among different objects depending on the clarity of the objects and the depth of flood. For example, the model detected people object with a score ranging from 38% to 100% and vehicles with average score of 68%.

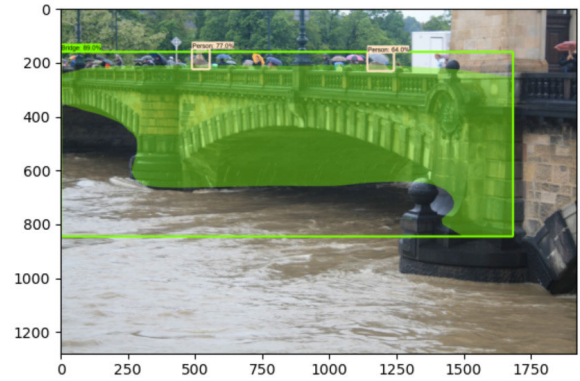


FIGURE 11. Bridge shape estimated with green colored polygonal shape using Mask R-CNN and displayed with confidence score on the estimation. The image also displays the predicted bounding box around persons detected over the bridge.

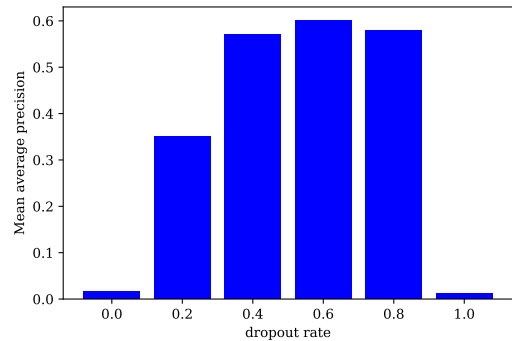


FIGURE 12. Mean average precision of the Faster RCNN progression with the Monte Carlo-dropout applied at the testing period.

2) ROADSIDE OBJECT DETECTION WITH MASK RCNN

Since bridges vary in shapes and structures, we manually created custom polygonal shaped annotations to segment bridges that have common physical attributes like uncovered top, arched pillars, etc. Training images included bridges from different locations, angles, and distance to incorporate variation into the dataset and made it more robust and generalize better during inference. Annotations and the corresponding mask images were fed into pre-trained Mask RCNN with Inception V2 as feature extractor. Fig. 11 demonstrates bridge shape detection results with filled polygon showing the regions of structure being detected using Mask RCNN.

I. UNCERTAINTY ASSESSMENT OF ROADSIDE OBJECT DETECTION MODEL

Capturing uncertainty in the proposed object detection module can provide insights into the error and uncertainty in detection tasks [43], [44]. Bayesian models lay the groundwork for moderately reasoning on how a model would behave in production, albeit the associated computational cost may dissuade the developer from proceeding. Introducing Bayesian uncertainty into neural networks involves randomness in weights and for the object detection model

this technique helps with estimating the uncertainty in bounding box regressor output. Bayesian methods captured the epistemic uncertainty which often occur due to lack of good observational data. For the nonlinear neural network, dropout in all its variant forms can be strategically applied before layers which helps to approximate probabilistic Gaussian process in otherwise ordinary neural network [45], [46]. In this research, uncertainty was introduced in the form of dropout during training of a two-stage region proposal network (RPN) base of scene object detection model. By comparison, Monte Carlo-dropout is applied during testing while Monte Carlo sampling was used to drop out neurons from models [47]. Mean average precision is conventionally formulated to make use of Precision and Intersection over Union (IoU). IoU is the ratio of area of intersection and union for predicted and ground truth bounding box. IoU has the task to figure out whether predicted bounding box is TP (True positive), FP (False Positive) or FN (False Negative). As long as the model achieves IoU over a certain threshold we can consider the prediction is TP. We used mean average precision of the model against increasing dropout rates as a metric for uncertainty measurement. Fig. 12 shows mean average precision of the Faster RCNN progression with Monte Carlo dropout applied during testing period. Results revealed that a range of 0.4-0.6 drop-out rate was sufficient to improve the modeling precision.

V. CONCLUSION AND FUTURE WORK

Imminent disaster response systems have continued to advance with plethora of technological innovations over the years to counter ongoing climate and landscape across the globe. Live monitoring video cameras, satellites, and social media, all contribute to the collection of real time data and advances in computer science makes timely, precise response possible. A system that encompasses multi-faceted detection features without compromising the detection accuracy and response time is continuously sought after by the researchers. In this paper, we proposed DX-FloodLine, an explainable, robust, end-to-end flood detection pipeline deployed to take action at near real time of a flooding event. We built our own dataset with custom annotations to account for historic events and wide range of multimedia image features, such as continuous changes in roadside objects occurred over the duration of flooding event. During urban flooding, risks to human and damages to house, transportation, and landscape are of utmost concern and thus we implemented a robust object detection model to capture object state in urban setting. DX-FloodLine has several modules embedded with neural networks for different tasks. We gradually built and experimented with multiple, stacked combinations of feature extraction and classification layers to find the best ensemble, named VGG16-LSTM to identify images with flood evidence, such as a house submerged in water or underwater road partially obstructed in view. We compared all proposed models with state-of-the-art model on the same

benchmark and VGG16-LSTM came on top. As evident by consistent good scores of accuracy, F1-score, and NMI, it is clear that the VGG16-LSTM works best for image feature extraction compared to the simple convolutional network. For upper network layers to connect extracted features for classification, LSTM works well as middle layer to connect extracted features and classification layer, although a dense or a global average pooling did the same job with lesser accuracy. our knowledge, the VGG16-LSTM ensemble is unique for large scale flood image classification. DX-FloodLine can be adapted to suit any disaster monitoring task where large amount of multimedia images is available. One of the pitfalls of disaster monitoring is the high execution latency caused from progressing through data collection to disaster prediction. DX-FloodLine lowers latency by using trained model inference to provide trustworthy prediction near real time. DX-FloodLine is robust to size inconsistency of gathered images as models were trained invariant to image resolution. When we consider the fact that impending hurricane and subsequent floods loom around the corner every couple of months, models need to consider the most recent urban scenery changes. We incrementally trained models on latest images and utilized interpretable attribute visualizations to improve the model's future performance and stability for near real time inference. The object detection module were able to find objects visible or partially hidden around the flooded scene. The Faster RCNN detected standard objects found in urban areas with high confidence score while the Mask RCNN worked better for irregular and larger shaped objects such as bridges and buildings. By including variations of object detection models, the proposed pipeline can provide a well-round narrative of the casualties around a developing flood scene. Uncertainty of the flood scene object detection model were captured and explained by Bayesian neural network approximation that involved adding dropout in the region proposal network contained by the Faster-RCNN model. Dropout were used as a regularization during training and test time as well with varying values of dropout probabilities. Because we faced challenges in detecting objects from partially obscured or poor-quality images, we wish to tackle the hidden shape estimation of submerged projects as our next task. We are expanding the category of objects to annotate to provide better assessment of damaged surroundings. Detectors also suffered from mis-classification errors introduced by noisy annotations. To rectify this, we propose a joint objective detection framework that detects objects and refines the bounding box annotations based on detection error margin. We are planning to add attention mechanism to the backbone network of Mask RCNN for bridge shape and instance segmentation, which will also help detecting multiple instances from the same sample. Emergency flood related decision-making system often fall under the criticism of unfair decision due to the hasty nature of response delivery in the face of high input load. We aim to thoroughly evaluate our flood detection pipeline on its ability to deliver fair prediction in near real

time regardless of input data frequency. In future, we plan to modify and adapt DX-FloodLine so it can be utilized for other image data domain where prompt response is crucial. We are working towards better scalability and stability of the pipeline when dealing with large volume of images in near real time. With distributed training and inference, it is possible to achieve better scalability. By persisting the incrementally trained models on cloud service, we can ensure stable execution. We plan to containerize our application in the future so it works in any system.

ACKNOWLEDGMENT

The authors would like to thank Clemson University for generous allotment of GPU units and compute time on the Palmetto cluster.

REFERENCES

- [1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.
- [2] N. Donratanapat, S. Samadi, J. M. Vidal, and S. S. Tabas, "A national scale big data analytics pipeline to assess the potential impacts of flooding on critical infrastructures and communities," *Environ. Model. Softw.*, vol. 133, Nov. 2020, Art. no. 104828.
- [3] J. L. P. Barker and C. J. A. Macleod, "Development of a national-scale real-time Twitter data mining pipeline for social geodata on the potential impacts of flooding on communities," *Environ. Model. Softw.*, vol. 115, pp. 213–227, May 2019.
- [4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Mar. 1998.
- [5] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.
- [6] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
- [7] A. Moumtzidou, S. Andreadis, I. Gialampoukidis, A. Karakostas, S. Vrochidis, and I. Kompatsiaris, "Flood relevance estimation from visual and textual content in social media streams," in *Proc. Companion Web Conf. Web Conf.*, 2018, pp. 1621–1627.
- [8] D. Fitriannah, D. Jatikusumo, and I. Nurhaida, "D-Loc apps: A location detection application based on social media platform in the event of a flood disaster," in *Proc. 2nd Asia Pacific Inf. Technol. Conf.*, Jan. 2020, pp. 41–45.
- [9] J. A. de Bruijn, H. de Moel, A. H. Weerts, M. C. de Ruiter, E. Basar, D. Eilander, and J. C. J. H. Aerts, "Improving the classification of flood tweets with contextual hydrological information in a multimodal neural network," *Comput. Geosci.*, vol. 140, Jul. 2020, Art. no. 104485.
- [10] R. J. Pally and S. Samadi, "Application of image processing and convolutional neural networks for flood image classification and semantic segmentation," *Environ. Model. Softw.*, vol. 148, Feb. 2022, Art. no. 105285.
- [11] L. Windheuser, R. Karanjit, R. Pally, S. Samadi, and N. C. Hubig, "An end-to-end flood stage prediction system using deep neural networks," *Earth Space Sci.*, vol. 10, no. 1, Jan. 2023, Art. no. e2022EA002385.
- [12] N. Humaira, S. S. Tabas, S. Samadi, and N. C. Hubig, "FlowDyn: A dynamic web-based streamflow prediction pipeline," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 757–767.
- [13] N. Hubig, P. Fengler, A. Züfle, R. Yang, and S. Günemann, "Detection and prediction of natural hazards using large-scale environmental data," in *Advances in Spatial and Temporal Databases*. Cham, Switzerland: Springer, 2017, pp. 300–316.
- [14] Y.-T. Lin, M.-D. Yang, J.-Y. Han, Y.-F. Su, and J.-H. Jang, "Quantifying flood water levels using image-based volunteered geographic information," *Remote Sens.*, vol. 12, no. 4, p. 706, Feb. 2020.
- [15] Y. Feng, C. Brenner, and M. Sester, "Flood severity mapping from volunteered geographic information by interpreting water level from images containing people: A case study of hurricane Harvey," *ISPRS J. Photogramm. Remote Sens.*, vol. 169, pp. 301–319, Nov. 2020.
- [16] K.-A.-C. Quan, V.-T. Nguyen, T.-C. Nguyen, T. V. Nguyen, and M.-T. Tran, "Flood level prediction via human pose estimation from social media images," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 479–485.
- [17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [18] P. Chaudhary, S. D'Arconco, J. P. Leitão, K. Schindler, and J. D. Wegner, "Water level prediction from social media images with a multi-task ranking approach," *ISPRS J. Photogramm. Remote Sens.*, vol. 167, pp. 252–262, Sep. 2020.
- [19] N. Singh, N. Roy, and A. Gangopadhyay, "Localized flood detection with minimal labeled social media data using transfer learning," 2020, *arXiv:2003.04973*.
- [20] J. M. A. Opella and A. A. Hernandez, "Developing a flood risk assessment using support vector machine and convolutional neural network: A conceptual framework," in *Proc. IEEE 15th Int. Colloq. Signal Process. Appl. (CSPA)*, Mar. 2019, pp. 260–265.
- [21] K. Lohumi and S. Roy, "Automatic detection of flood severity level from flood videos using deep learning models," in *Proc. 5th Int. Conf. Inf. Commun. Technol. Disaster Manage. (ICT-DM)*, Dec. 2018, pp. 1–7.
- [22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [25] J. Brownlee, *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning Machine Learning Mastery*. Accessed: Jul. 2017. [Online]. Available: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning>
- [26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
- [30] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [31] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for PyTorch," 2020, *arXiv:2009.07896*.
- [32] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.
- [33] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.
- [34] G. Gravier, B. Bischke, C.-H. Demarty, M. Zaharieva, M. Riegler, E. Dellandrea, D. Bogdanov, R. Sutcliffe, G. Jones, and M. Larson, Eds. "Working notes proceedings of the MediaEval 2017 workshop co-located with the conference and labs of the evaluation forum (clef 2017), Ireland, September 13–15, 2017," in *Proc. CEUR Workshop*, vol. 1984, 2017. [Online]. Available: <https://ceur-ws.org/Vol-1984>
- [35] B. Barz, K. Schröter, M. Münch, B. Yang, A. Unger, D. Dransch, and J. Denzler, "Enhancing flood impact analysis using interactive retrieval of social media images," 2019, *arXiv:1908.03361*.
- [36] S. Rahman, M. M. Rahman, M. Abdullah-Al-Wadud, G. D. Al-Quaderi, and M. Shoyaib, "An adaptive gamma correction for image enhancement," *EURASIP J. Image Video Process.*, vol. 2016, no. 1, pp. 1–13, Dec. 2016.
- [37] G. Cao, L. Huang, H. Tian, X. Huang, Y. Wang, and R. Zhi, "Contrast enhancement of brightness-distorted images by improved adaptive gamma correction," *Comput. Electr. Eng.*, vol. 66, pp. 569–582, Feb. 2018.

- [38] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process.*, vol. 5, no. 2, pp. 1–11, Mar. 2015.
- [39] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Jan. 2003.
- [40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [41] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," 2020, *arXiv:2006.03677*.
- [42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [43] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for LiDAR 3D vehicle detection," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3266–3273.
- [44] S. Wirges, M. Reith-Braun, M. Lauer, and C. Stiller, "Capturing object detection uncertainty in multi-layer grid maps," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1520–1526.
- [45] A. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Artificial Intelligence and Statistics*. PMLR, vol. 31, 2013, pp. 207–215.
- [46] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
- [47] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3243–3249.



NUSHRAT HUMAIRA received the bachelor's degree in computer science and engineering from the Bangladesh University of Engineering and Technology, Bangladesh, and the master's degree in computer science from Clemson University, USA, in 2020, where she is currently pursuing the Ph.D. degree in computer science with the School of Computing. Her research interests include data science and explainable machine learning for large scale datasets. She has previously published with

IEEE conferences as a first author.



VIDYA S. SAMADI is currently an Assistant Professor and the Director of Hydrosystem and Hydroinformatics Research (HHR) Group, Department of Agricultural Sciences, Clemson University, USA. Prior to joining Clemson University, in Spring 2020, she was a Research Assistant Professor with the Department of Civil and Environmental Engineering, University of South Carolina, and a Postdoctoral Scholar with the NOAA Center for the Carolina Integrated Sciences and Assessments and the Department of Architectural, Civil, and Environmental Engineering, School of Engineering, Cardiff University, U.K. She has authored/coauthored more than 65 journal articles and peer reviewed conference proceedings. Her research interests include leveraging advances in hydroinformatics and cyber-physical systems to address problems and challenges associated with water resources modeling systems.



NINA C. HUBIG received the Ph.D. degree in machine learning from the Technical University of Munich. She is currently an Assistant Professor with the School of Computing, Clemson University. Prior to Clemson University, she was with BMW as a Senior Data Scientist. Her research interests include explainability in artificial intelligence and other ML topics. In regard to environmental data science, she has worked with NASA and IBM research on large weather data and nature catastrophe prediction.

...