An Automated Mapping Approach of Emergency Events and Locations Based on Object Detection and Social Networks

Khalid Alfalqi, Martine Bellaiche
Computer and Software Engineering Department, Ecole Polytechnique, Montreal, Canada

***Abstract*—The high prevalence of cellphones and social networking platforms such as Snapchat are obviously dissolving traditional barriers between information providers and end-users. It is certainly relevant in emergency events, as individuals on the site produce and exchange real-time information about the event. However, notwithstanding their demonstrated significance, obtaining event-related information from real-time streams of vast numbers of snaps is a significant challenge. To address this gap, this paper proposes an automated mapping approach of emergency events and locations based on object detection and social networks. Furthermore, employing object detection methods on social networks to detect emergency events will construct reliable, flexible and speedy approach by utilizing the Snapchat hotspot map as a reliable source to discover the exact location of emergency events. Moreover, the proposed approach aims to yields high accuracy by employing the state of the arts object detectors to achieve the objectives of this paper. Furthermore, this paper evaluates the performance of four object detection baseline models and the proposed ensemble approach to detect emergency events. Results show that the proposed approach achieved a very high accuracy of 96% for flood dataset and 94% for fire dataset.**

***Keywords*—Machine learning; deep learning; big data; social networks; object detection; emergency event detection; snapchat; hotspot map**

I. Introduction

The prevalence of emergency events has increased exponentially over the past half-century, disturbing since it causes infrastructure damage, casualties, and long-term socio-economic harm [1]. It is described as a significant instability of life of the community caused by hazardous incidents resulting in either one of the following: human, financial, and environmental damages, and consequences, according to the United Nations International Strategy for Disaster Reduction (UNISDR). [2] Emergency events fall into two categories: natural and human-made. Natural emergency events are further subdivided into geophysical, meteorological, biological, climatological, and extraterrestrial calamities. Human-caused emergency events can be further categorized into occupational incidents, traffic incidents, and other incidents. Natural disasters include landslides, earthquakes, floods, tsunamis, and wildfires, and human-made emergency events include explosions, large-scale building fires, toxic emissions, and aviation and railroad incidents [3]. Thus, many emergency events have serious economical consequences in the short term, but rare situations can result in longer economic loss. Detection and recognition of objects are extensively employed in a variety of areas. People live in the era of smart communities that provide a massive volume of information. Conventional learning techniques are incapable of dealing with massive amounts of data, but data can match the requirements of deep learning for a wide range of experimental datasets [4]. In addition, the computer power of contemporary hardware has substantially grown as technology has developed. Thus, object detection based on deep learning has improved remarkably well.

Several deep learning-based object detection and target recognition methods have recently been proposed. The deep convolutional neural network can automatically train and enhance its hyper-parameters using the supplied data set [5]. Deep learning object recognition techniques are categorized into two categories. The first type of algorithm is a two-stage object detection method, such as R-CNN, Fast R-CNN, Mask R-CNN, and Faster R-CNN [6]. Object detection is carried out in two steps by these methods. The region proposal network is performed first to retrieve candidate objects of interest and is followed by the detection network to predict and recognize the candidate object's region and class [7]. The second type of detection algorithm is a one-stage detection algorithm, which involves SSD, RefineDet, YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, and YOLOR. These methods bypass the region proposal network and generate location and class information concerning the objects straight throughout the network. As a consequence, the one-stage detection method recognizes objects faster [8].

The remainder of the paper is structured as follows: The problem definition is introduced in Section II. Section III describes the related works of two separate research topics covered in this study. Section IV "Background" provides a high-level overview of object detection models and CNN architectures. Section V "Methodology" describes the phases in the research methodology, as well as the approach and models employed in the experiments. Section VI "Evaluation approach", describes the specified case studies in depth, including the reasons for their selection and experimental environment construction, evaluation measures, performance analysis for our suggested approach, and discussion of study outcomes. Finally, Section VII "Conclusion and Discussion", highlights the findings and suggests future study avenues.

II. Problem Definition

Deep learning approaches have been used in previous studies to auto-link between emergency events and places using social networking information to analyze incident recognition and assess spatial geolocation data from geotagged social network content. However, georeferenced data makes up a relatively minor portion of the whole set of social network information, and it may not precisely correlate to events mentioned in the posts. Hence, location awareness and more detailed emergency event information could significantly improve social network content's usefulness, reliability, and compatibility [9].

The precision of an approximated position nowadays is to the level of a town or district when employing current approaches, and obtaining further relevant details like the name of a road or building remains challenging [10]. As a result, improved methods are required to significantly boost the accuracy and robustness of geolocation data collected from social networks.

This paper proposes an automated mapping approach between emergency events and locations based on object detection and social networks to address this gap. Furthermore, employing object detection methods on social networks to detect emergency events will construct a reliable, flexible, and speedy approach by utilizing the Snapchat hotspot map as a reliable source to discover the exact location of emergency events. Moreover, the proposed approach aims to yield high accuracy by employing the most current object detection methods to accomplish the objectives of this paper. More particularly, the contributions of this paper can be stated as follow:

- Detect the occurrence of emergency events across time and exact locations by utilizing the Snapchat map.
- Develop a novel ranking and selection model to rank the hotspot locations in order to prioritize which location needs urgent dealing and send it to the decision-makers.
- Develop a flexible, reliable and speedy object detection model to analyze images for the collected data from Snapchat API.
- Propose a combination of one-stage and two-stage object detection models to construct an ensemble model and evaluate the performance of our approach.

III. Related Work

This section outlines the literature review completed throughout this study on two separate fields of study. First, we highlight current research studies that address location detection methods for social networks, concerns, as well as existing and potential solutions. Second, we expand in a limited but adequate manner on current deep learning methods for object detection in emergency events.

A. Location Detection Methods for Social Networks

People usually are inquisitive about the location of catastrophes during or shortly after they occurred. Knowing the precise location is critical for decision-makers to respond promptly and make decisions accordingly [11]. We can get emergency event geolocation information from social network feeds. Location retrieval can be classified into three elements based on the methods used. The first element is the content analysis of words. Second, the analysis of different language modelling. Third, by guessing through social relations [12]. Many studies have been conducted in order to approximate the location of emergency events using material analysis in a specific outer data source depending on geo-related factors.

Singh et al. [13] presented a tweet categorization and location identification technique for spotting tweets from emergency affected seeking assistance as well as their locations. If the location is not stated in the previous tweets, the location was inferred utilising Markov Chain Stochastic approach.

Rodriquez et al. [14]. proposed Using a probabilistic technique that simultaneously predicts location designations and Twitter posts of users, the relationship network is represented graphically. In particular, the authors characterize the system with a Markov random field probability framework, and the training phase is based on a Markov Chain Monte Carlo simulator that estimates the posterior likelihood distribution of the sparse spatial user labels.

Duong-Trung et al. [15] developed a dynamic content-based regression approach using the matrix decomposition method to address the near real-time location estimation challenge. They demonstrated that real-time location estimation is feasible without combining users' tweets.

Gelernter et al. [16] applied Named Entity Recognition technique (NER) to recognize the names of locations in tweets, The findings were contrasted to words manually classified as locations by the researchers and revealed that the enormous number of abbreviations for landmarks is a significant concern for this technique.

Rout et al. [17] employed a Support Vector Machine algorithm to estimate the location of tweets. In addition, their dataset comprises relationship graph features and city metrics like demographic size and the amount of active Twitter accounts. The relationship graph features were adopted because most online social networking relationships begin in real life. They further contend that some elements of the towns are pertinent to the categorization phase.

Laylavi et al. [18] developed a technique for identifying relevant tweets regarding the storm incident. First, they defined event-related word categories utilizing term frequency analysis and the relationship scoring mechanism. Then, an incident-related score was assigned to each tweet. Finally, the suggested system's findings were matched to manually labelled data to measure performance. The recommended approach successfully categorized around eighty-seven incident-related tweets.

Table I illustrates the comparison of the related works in Section III-A. BD is denoted to big data and OD is denoted to object detection.

B. Deep Learning Methods for Object Detection in Emergency Events

The application of deep learning in emergency events detection is becoming more and more extensive. Significant advancements in computer vision have lately been accomplished using these technologies. Multiple detection strategies and techniques have demonstrated superior picture detection efficiency.

TABLE I. A COMPARISON AND SUMMARY OF THE LITERATURE REVIEW

Method	Ref	Dataset	results	BD use	OD use
Markov chain based	[13]	Twitter	87%	No	No
Markov Chain Monte Carlo	[14]	Twitter	78.99%	No	No
Matrix factorization technique	[15]	Twitter	79%	No	No
Named Entity Recognition	[16]	Twitter	90%	No	No
Support Vector Machine	[17]	Twitter	-	No	No
Term frequency analysis	[18]	Twitter	87%	No	No

Ji et al. [19] Used satellite images taken prior to and following the earthquake, a pre-trained VGG model was utilized to detect destroyed houses affected by the Haiti earthquake. The research findings reveal that the fine-tuned VGG model's accuracy level has risen from 83.38 to 85.19. In addition, the destroyed houses identification effect works better, with a production accuracy of 86.31 of earthquake-induced house damage from satellite utilizing a pre-trained CNN classifier, indicating that the CNN approach can successfully identify the characteristics of destroyed structures houses.

Miura et al. [20] employed CNN model to obtain the characteristic of the destroyed buildings' rooftops that wrapped with blue tarpaulin following the earthquakes. Leveraging the enhanced CNN network and satellite photos taken following the two events in Japan.

Qingjie Zhang et al. [21] proposed a novel deep learning method for detecting forest fires. They employed a cascaded method for detecting fire, with the global picture stage examining the whole picture initially and then a local patch encoder identifying the particular position of the determined fire. They presented a baseline for fire recognition, 178 pictures for the train set, and 59 pictures for the test set. They utilize the CIFAR 10 network for the initial phase but reduce the number of outcomes by two and add a drop-out layer to minimize the fitting problem. They employed the Caffe framework's 8-layer AlexNet for the final phase.

Ci et al. [22] proposed a novel CNN architecture integrated with a CNN data harvester, a unique loss function, and an arbitrary regression classifier to measure the severity of houses destruction due to earthquakes utilizing satellite images.

Pi et al. [23] exploited a pre-trained model to train several CNN classifiers based on You-Only-Look-Once (YOLO) in the accident areas to detect intact houses rooftops.

Wei Zhang et al. [24] developed a CNN-CAPSNet—an effective remote imagery categorization system that emphasizes the advantages of the CNN and CAPSNet algorithms. A CNN with partially coupled layers was used as the initial feature map generator. Furthermore, for feature extraction, they employed a pre-trained deep CNN classifier that was entirely trained on the ImageNet dataset. The findings demonstrated that the suggested strategy might outperform state-of-the-art approaches in classifier performance.

Pham et al. [25] proposed a YOLO-fine approach which is an enhanced object detection model from YOLOV3. The model was developed to identify tiny objects with high accuracy and efficiency, allowing real-time applications in realistic circumstances. In addition, they explored its resilience to the appearance of new contexts on the validation set deeper, tackling the critical barrier of domain adaptability in satellite imagery.

Yebes, J et al. [26] acquired and labelled images of real-world situations, and several object detection algorithms were fine-tuned to accomplish their experiment. Their highest results yielded an accuracy rate of 82, employing a hybrid of R-CNN and Resnet101.

Amit et al. [27] suggested a CNN-based paradigm for catastrophe identification in aerial pictures: two fully connected layers and three convolutional and max-pooling layers in the suggested CNN framework. A dataset is gathered for assessment purposes that spans many aerial picture patches from two different natural catastrophes, specifically landslides and flooding. Table II illustrates the comparison of the related works in Section III-B. BD is denoted to big data and OD is denoted to object detection.

IV. BACKGROUND

This section briefly describes important background concepts related to this study, including object detection methods and CNN architectures,

A. Object Detection

Objects detection methods fall into two min categories, one of which is one stage detectors and the other is two stage detectors [6]. Fig. 1 shows the stages of object detection models.

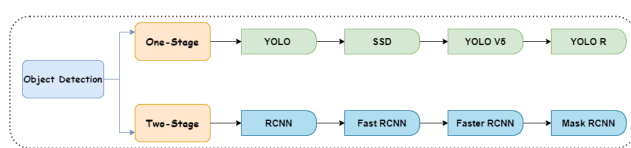

Fig. 1. Stages of object detection.

1) One stage detectors:

a) YOLOV5: is an upgraded version of the YOLOV3. Its implementation is comparable to that of YOLOV4 in that it integrates numerous approaches such as data augmentation and modifications to activation functions with data preprocessing

TABLE II. A COMPARISON AND SUMMARY OF THE DEEP LEARNING METHODS FOR OBJECT DETECTION IN EMERGENCY EVENTS

Method	Ref	Dataset	results	BD use	OD use
CNN	[19]	satellite imagery	87.6%	No	No
CNN	[20]	aerial images	95%	No	No
CNN	[21]	various online resources	90%	No	No
CNN	[22]	multiple aerial imagery	93.95%	No	No
CNN YOLO	[23]	in-house aerial video imagery	80.69%	yes	No
CNN-CapsNet	[24]	UC Merced Land-Use ,AID , NWPU-RESISC45	98.81%	No	No
CNN YOLO-fine	[25]	VEDAI ,MUNICH,XVIEW	84.34%	yes	No
CNN Faster R-CNN- SDD	[26]	images with pothole annotations	82%	yes	No
CNN	[27]	satellite imagery	80%	No	No

to the YOLO structure. Yet, YOLOv5 differs from YOLOV4 in the base; rather than Darknet, it uses CSPDarknet53 as its base. This base addresses the redundant gradient data in a massive backbone network. It incorporates gradient modification into the feature map, decreasing inference latency, boosting precision, and lowering model complexity by minimizing the parameters. In addition, it aggregates pictures for training and uses self-adversarial training (SAT) to guarantee faster prediction [28].

b) YOLOR: You Only Learn One Representative: The YOLOR model, which has an integrated network design, is intended to carry out several tasks at the same time. YOLOR is an upgraded version of the YOLO model that takes advantage of the data in the lower tiers of the CNN layers, specifically the attribute data. Furthermore, because it provides concurrent recognition, the YOLOR method is more effective than the existing YOLO methods [29].

YOLOR simultaneously utilizes implicit and explicit information to model training to acquire generalized representations and execute multiple jobs using these generalized representations. The implicit awareness recognizes deep layer characteristics, and annotated data is used to gain explicit information [29]. YOLOR has different versions with distinct specifications, so YOLOR-P6 is considered for the study.

2) Two stage detectors:

a) Faster R-CNN: Faster R-CNN is upgraded version of Fast R-CNN to overcome the limitations of Fast R-CNN. The main drawback of Fast R-CNN is that it leverages particular inquiry to generate Regions-Of-Interest (ROI), which is slower and requires the exact period to execute as the recognition system. Therefore, faster R-CNN is substituted by a unique RPN (region proposal network), a fully convolution network that can anticipate region suggestions using a broad variety of sizes and aspect ratios. Furthermore, since it combines full-image convolution characteristics and a similar set of convolution layers with the overall classifier, RPN speeds up the production of region suggestions [30].

b) Mask R-CNN: Mask R-CNN adds mask branch output on top of the prior Faster R-CNN base. The core layer gathers features at the beginning; then, the proposals are anticipated and optimized to infer the bounding boxes for object recognition and build segmentation masks. The mask offers pixel-level feature extraction for every potential object by performing instance segmentation [31]. In addition, it incorporates enhancements in Faster R-CNN and FCN (Fully Connected Network), which led to its popularity as a two-stage object detection model compared to the other models, as it offers both bounding box and segmentation [32].

B. Convolutional Neural Networks (CNN) Architecture

Convolutional neural networks are the most extensively used deep learning algorithms and the most popular type of neural network. It is a multi-layer neural network (NN) design comprising a convolutional layer(s) followed by a fully connected (FC) layer (s). The principal use of CNN is in databases, where the number of nodes and parameters is enormous [33].

1) Convolutional layer: This layer is the foundation of CNN models, defining linked inputs' outcomes. Such an outcome is accomplished by convolving kernels across the datasets' size and shape, determining the feed's and filter's dot product, and constructing a two-dimensional activation map with that filter. CNN efficiently understands which filters to activate when a certain kind of attribute is noticed at a specific spatial point in the input [34].

2) Non-linearity layer: Nonlinear functions are essential and retain a degree greater than one; when displayed, they exhibit a curve. The primary goal of this layer is to convert the incoming signal to the outgoing signal, which will be employed as an input in the subsequent layer. Non-linearity layers include sigmoid or logistic, Tanh, ReLU, PReLU, and ELU [35].

3) Pooling layer: CNNs can comprise internal or external subsampling layers that combine the results of one layer's neurons into an individual neuron in the subsequent layer. Its primary function is to reduce the spatial size of the depiction to decrease the size of the parameters and computations in the framework. It prevents overfitting and accelerates the computation. The max-pooling layer is the most frequent type of pooling layer [35].

4) Fully connected layer: FC layers are conventional deep NN layers that aim to create forecasts from activation for regression and classification. This layer obtains entire links to each activation in the preceding layer, and the activation may be determined by matrix multiplication coupled with a bias of sets [33].

V. Research Methodology

In this section, elucidation of the critical components of this article is required by outlining the steps needed. First, hotspot location in Snapchat map will be pinpointed. Second, collect the dataset from the Snapchat map API in the exact locations of the emergency events that were identified earlier without any prior knowledge. Third, data pre-processing and data augmentations methods are employed at this stage. Fourth, object detection models for emergency event detection are proposed. Fifth, we deploy an ensemble learning based for emergency events to conduct a performance evaluation of the proposed model. Finally, locations ranking and model selection is proposed to prioritize the hotspot locations. The steps of the proposed model is shown in Fig. 2.

A. Pinpoint Hotspot Locations in Snapchat Map

Pinpointing the precise location of emergency events is very important. Therefore, by leveraging the Snapchat map we can find the exact location of any emergency event. The snaps posted on the Snapchat Interactive Map is automatically geotagged and represent a particular hotspot location which can be a possible emergency event location [36]

B. Data Collection

Accordingly, after identifying the hotspot locations, a preliminary ranking is applied based on the highest number of snaps in each location to prioritize the locations. Then we will use the same method we used in our previous works wrapper for the Snapchat Map's internal API in order to construct a Node.js function that scans for snaps uploaded at precise coordinates (hotspot zones) [37] which will collect all the snaps shared at the exact location and marked as either emergency event-content or non-emergency event-content. The acquired snaps might be image or video, thus,the images they will be analyzed by utilizing object detection techniques.

C. Data Pre-processing

1) Standardize images: One fundamental limitation in several object detection methods is the requirement to scale the datasets to a consistent size. This means that before we feed the images to the training model, they need to be preprocessed and resized to have equal dimensions [38]. As a result, the collected images are resized to a length of 640*640.

2) Data augmentation: Another frequent preprocessing strategy is to augment the current dataset with different replicas of the original images. It is performed to expand the dataset and introduce the neural networks to a broad range of image permutations [38]. In addition, it will increase the likelihood that the model will detect objects in any configuration.

Data augmentation may successfully prevent fitting problems throughout the advanced training phase and can tremendously enhance the clarity of the data [39]. Several data augmentation methods were employed, including vertical and horizontal flipping, rotating to a specific angle (less than 20°), and raising or reducing brightness.

D. Applying Object Detection Models

After collecting the dataset from the Snapchat map API, we will apply the most common object detection algorithms which are : You Only Look Once V5 (YOLO v5), You Only Learn One Representation (YOLOR), Mask RCNN and Faster R-CNN. We will apply these algorithms to every hotspot location that have been identified earlier to detect emergency events. Then, we will evaluate the performance of these algorithms using evaluation metrics. Table III shows the hyper-parameters of the models. Algorithm 1 depicts training and evaluation steps.

E. Proposed Voting Ensemble Object Detection Model

Among the four object detection models mentioned above, we will propose to use of the ensemble learning. The precision of the object detection model may be boosted by merging several models into an ensemble model which can be effectively applied for emergency event detection. One of the primary factors driving the prevalence of ensemble learning is its capability to minimize the variance and bias of deep learning models [40].

Numerous object detection models have been introduced to ensemble learning for different scenarios; however, using an ensemble learning approach with object detection models in emergency event detection is still appealing. Both one-stage and two-stage models performed well for object detection, as discussed in the related works. Therefore, four baseline object detection models were chosen for the proposed ensemble approach in this work. We applied a combination of single-stage and two-stage models to conduct our proposed ensemble approach. The proposed voting ensemble object detection model is shown in Fig. 3.

F. Location Ranking and Selection Model

With the continuous search of hotspot zones in the Snapchat map, multiple locations will be found and processed. In this step we will rank these hotspot locations to prioritize them in terms of ranking criteria. Algorithm 2 explains the ranking model. This model consists of four criteria metrics:

a) Amount of snaps in the location: It counts how many snaps have been posted in this particular hotspot location. The overall score for this criteria is one point. Since we only have two locations of emergency events, the highest will get one point and the least will get zero.

b) Amount of snaps related to the emergency events: It counts only the snaps that are related to the emergency event in this particular hotspot location. This step is important to get rid of unrelated snaps. The overall score for this criteria is two points. Since we only have two locations of emergency events, the highest will get two points and the least will get zero.

c) mean Average Precision (mAP) score: A higher mAP score shows improved object detecting method performance. Since we have applied four object detection models to the dataset, we will get four different mAP score for each model. Therefore, we will calculate the mean of this score which the sum of all scores divided by the total number of models. The overall score for this criteria is three points. Since we only have two locations of emergency events, the highest

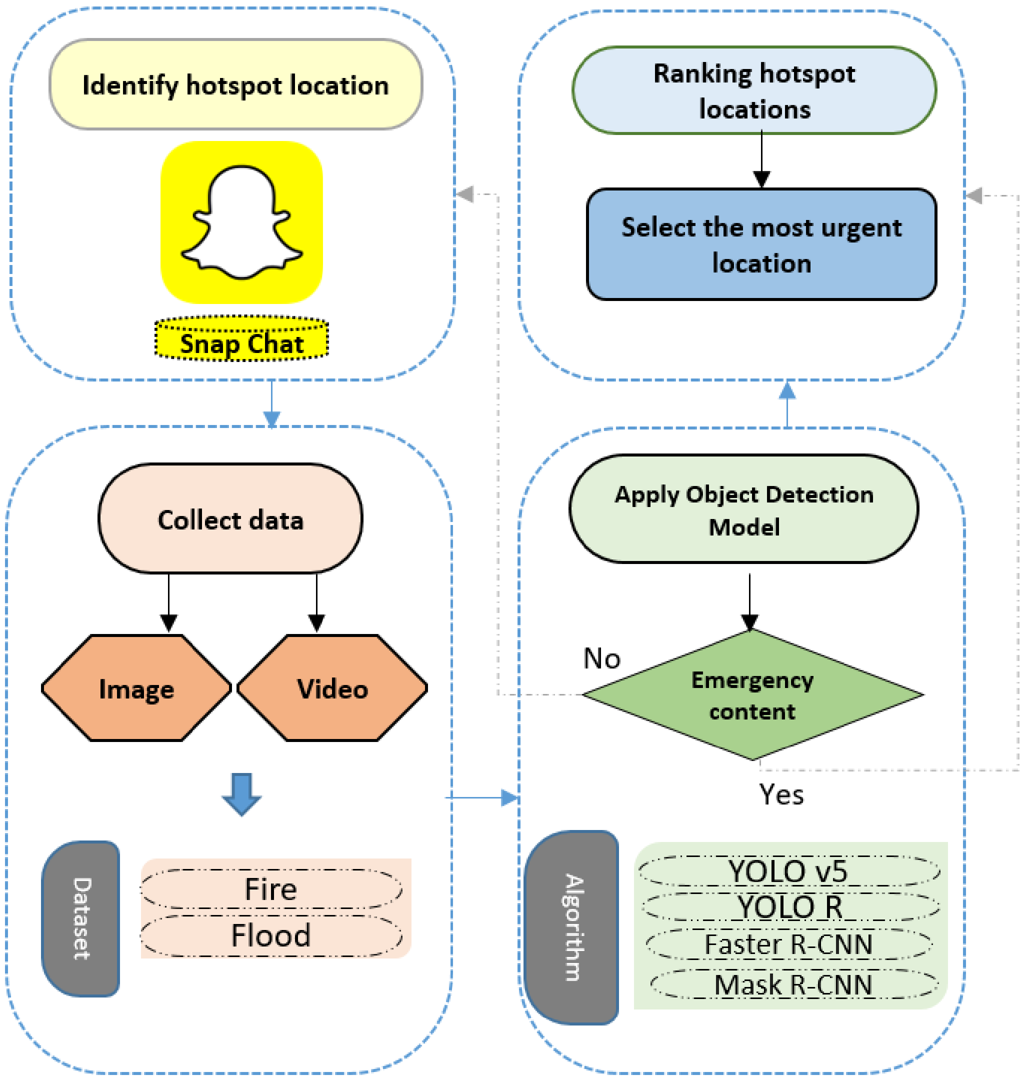

Fig. 2. The steps of the proposed model.

will get three points and the least will get zero. It can be calculated as:

$$Mean = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{a_1 + a_2 + \cdots + a_n}{n} \quad (1)$$

d) Throughput score (TP score): It refers to the quantity of outcomes delivered in a particular amount of time. After we evaluate the models performance using Spark, the model that will get the best performance for both case studies in regards to time processing will be selected. Then by matching this model to the corresponding case study (location), the corresponding location will get selected. The overall score for this criteria is four points. Since we only have two locations of emergency events, the lowest processing time model will get four points and the rest will get zero.

After that, we will calculate the summation of all scores, the location with a score of equal or greater than five will be sent in cluster A and will be selected as the most urgent location.

$$\text{if } Location(L) \geq 5 \text{ then } put\ it\ in\ Cluster(A) \quad (2)$$

If the location gets a score of equal or smaller than three, it will be sent to cluster B.

$$\text{if } Location(L) \geq 3 < 5 \text{ then } put\ it\ in\ Cluster(B) \quad (3)$$

Also, If a location get a score less than two, it will be sent to cluster C and will be disregarded.

$$\text{if } Location(L) \leqslant 2 \text{ then } put\ it\ in\ Cluster(C) \quad (4)$$

The ranking model is shown in Fig. 4.

TABLE III. HYPER-PARAMETERS OF THE MODELS

Models	Hyper-Parameters
YOLOV5	lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, epochs=100, batch_size=16
YOLOR	'lr0': 0.01, 'lrf': 0.2, 'momentum': 0.937, 'weight_decay': 0.0005, epochs=100, 'warmup_epochs': 3.0, 'warmup_momentum': 0.8, 'warmup_bias_lr': 0.1, batch_size=16
Faster R-CNN	lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, epochs=50, batch_size = 8
Mask R-CNN	lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, epochs=50, batch_size = 8

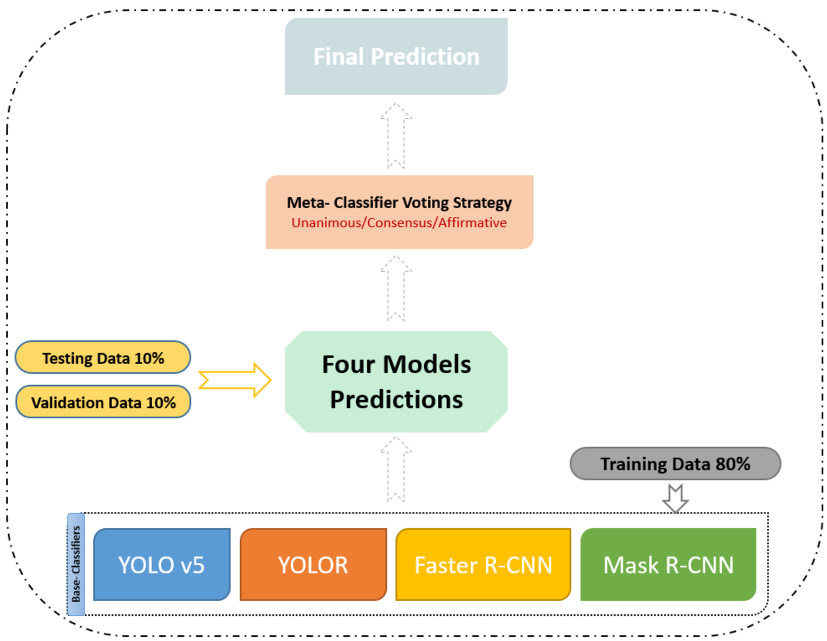

Fig. 3. Proposed voting ensemble object detection model.

Algorithm 1 Location Ranking and Selection

Input: *LocationList*; *RankingCriteria*; *ModelsList*;
Output:Ranked Locations

spark ← SparkConnector()
spark.load(Location List)
spark.load(Ranking Criteria)
for each location in (Location List) **do**
    Rank_score ← Amount of snaps in the location = 1 point
    Rank_score ← Amount of emergency related content = 2 points
    Rank_score ← Mean Average Precision(MAP) score = 3 points
    Rank_score ← Throughput score = 4 points
    Calculate the sum of Rank_score
    if the sum of Rank score ≥ 5 **then**
        Put this location in cluster (A)
    if the sum of Rank score ≥ 3 AND < 5 **then**
        Put this location in cluster (B)
    if the sum of Rank score ⩽ 2 **then**
        Put this location in cluster (C)
return *Ranked Locations* ← Location List

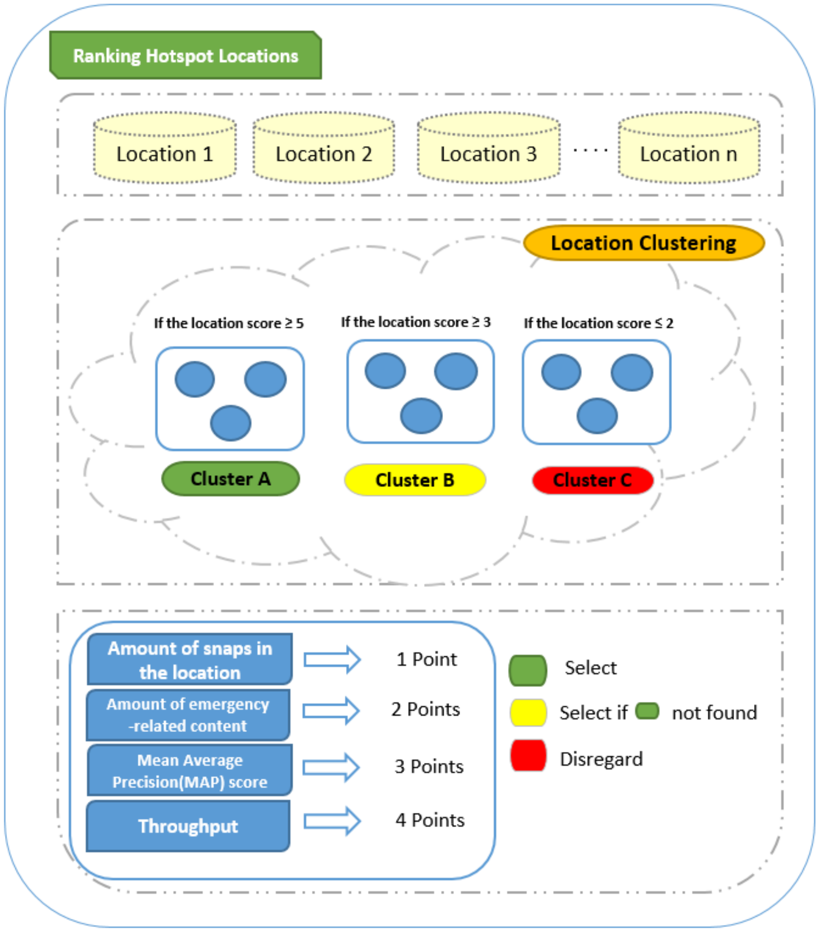

Fig. 4. Proposed location ranking and selection model.

VI. Evaluation Approach

This section describes the specified case studies in depth, including the reasons for their selection and experimental environment construction, evaluation measures, performance analysis for our suggested approach, and discussion of study outcomes.

A. Case Study

During the scan for the hotspot zones in the Snapchat map utilizing the constructed Node.js wrapper, we identified more than five hotspots location between March 2022 and May 2022 .However, according to the preliminary Snapchat data collection ranking criterion, the most hotspot location that contains the most amount of snaps at that moment were India building fire and South Africa flooding. Therefore, after identifying the exact location of these emergency events, We collected the dataset of each incident using Snapchat API. The total number of the collected images and videos are stated in Table IV.

1) India building fire: At least 27 individuals died, and 24 people were injured in a devastating fire in the Indian capital of New Delhi on Friday (May 13), according to rescue teams. The big fire burst in the mid-evening at a four-story residential building in west Delhi, although the cause was not intuitively known. Twenty-seven burnt remains were found from the building, also several of the residents jumped from the building during the fire and were hospitalized [41]. Fig. 5 shows the location of building on fire in India. Fig. 6 shows the detection results of building on fire in India.

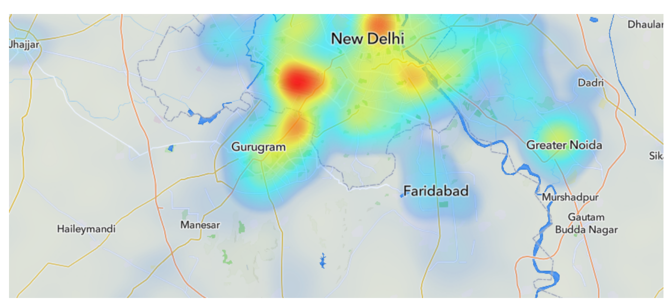

Fig. 5. Location of building on fire in India.

2) South Africa flooding: Heavy rainfall caused catastrophic floods and landslides in southern and south-eastern South Africa, notably in the provinces of KwaZulu-Natal and the Eastern Cape, from April 11 to 13. According to national authorities, 443 people have died in KwaZulu-Natal, while over 40,000 remain missing. Also, over 40,000 people have been evacuated, while 4,000 homes have been demolished or

Algorithm 2 Training and Evaluation

Input: *LocationList*; *Datasetslist*; *ModelsList*;
Output:Trained models list

spark ← Spark Connector()
spark.load(Location List)
spark.load(Models List)
spark.load(Datasets List)
for *each* Dataset in (Datasets List) **do**
    Trained_models ← Train YOLOv5
    Trained_models ← Train YOLOR
    Trained_models ← Train Faster R-CNN
    Trained_models ← Train Mask R-CNN
    for *each* model in (Trained_models) **do**
        Evaluation_score ← Calculate Recall
        Evaluation_score ← Calculate Precision
        Evaluation_score ← Calculate IOU
        Evaluation_score ← Calculate AP
        Evaluation_score ← Calculate mAP
        Evaluation_score ← Calculate Throughput
    return Trained models list ← Trained_models

TABLE IV. AMOUNT OF DATASET

Location	All data	Data related to emergency events
India fire	780 images	546
South Africa flooding	759 images	524

Fig. 6. Detection results of building on fire in India.

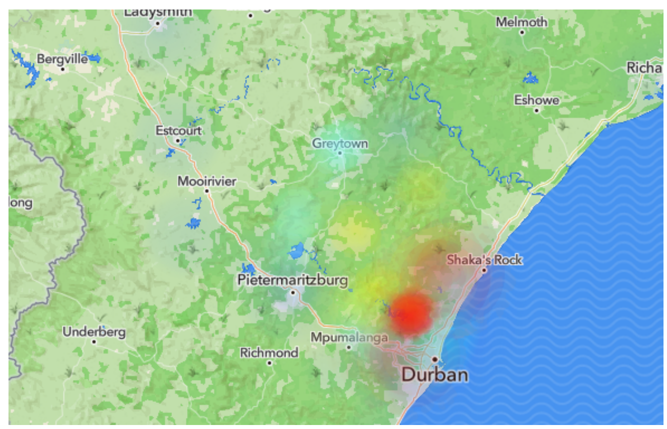

Fig. 7. Location of flooding in South Africa model.

severely damaged, mainly in Durban and its nearby regions. Due to the severe flooding, a National State of Disaster has been announced. Emergency crews have been dispatched to the areas impacted to give immediate aid to people affected by disasters [42]. Fig. 7 shows the location of flooding in South Africa. Fig. 8 shows the detection results of flooding in South Africa.

B. Experimental Setup

We designed our models with different python packages. The core libraries that we used were Pytorch, numpy, opencv, matplotlib, Scikit-Learn, Keras, and native TensorFlow. The hardware used for this training and testing of models was Nvidia Tesla P100 (16 GB) offered by Google Colab Pro. High Ram offered by Google Colab Pro was used to increase the IO speed from google drive datasets.

To evaluate the performance of our approach, we first Identified the hotspot locations from the Snapchat map using the developed Node.js Google function wrapper in our previous paper. Then we collected the dataset from the Snapchat map API in the exact locations of the emergency events that were identified earlier without any prior knowledge.

The total amount of dataset collected from both locations are shown in Table IV. In our first case study (the location of South Africa flooding) we were able to collect 780 images. However after we manually analyzed the dataset, we found out that there were some images that were unrelated to the case

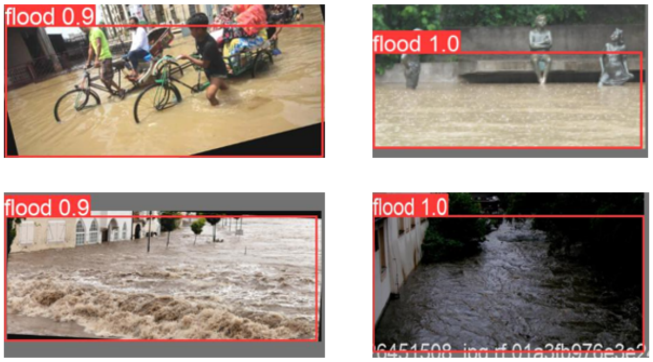

Fig. 8. Detection results of flooding in South Africa.

study and we got rid of them. The total amount of dataset in that location dropped to 546 images. Also, we were able to collect 759 images for the second case study (the location of flooding in South Africa). However, there were some images that unrelated to the case study which we got rid of them. After deleting the noise images, the total number of images became 524 images.

After cleaning our dataset, we used Roboflow online annotation tool that manually labels our dataset, generates boundary boxes, and classifies them. To guarantee that the dataset is spread evenly while considering the correlation between labels and data, the dataset is arbitrarily separated into three sets: training, validation, and test. according to the ratio of %80, %10, and %10. We created three folders to tore the dataset accordingly: training; testing; and validation [43]. Training data are included in the data and enable the deep learning model in producing a prediction. Validation data indicate if the model is competent of accurately detecting new instances or not, and they usually comprise pictures that the model employs to assess and monitor its learning. The images in the testing folder can be used to predict model correctness. The final dataset is saved in the XML dataset format to maintain the same experimental configuration.

Then, we applied the pre-processing steps to the collected images after getting rid of images that were unrelated to the case studies and annotate the related images. After that, due to the fact that object detection models require as much data as possible, our dataset were very low and we should find ways to enlarge them. Therefore, we applied some data augmentation methods to increase the dataset and to ensure the effectiveness of the training. Furthermore, due to the training dataset we used in our research is relatively small even after employing data augmentation methods,We employed transfer learning to overcome this challenge by training our models on the MS-COCO dataset and obtaining a pre-training model as MS-COCO is the baseline standard for validating and testing object detection methods. The pre-trained weights were loaded, and the training began with them as a starting point.

The pre-training algorithm can retrieve the generic characteristics of all objects from the standard datasets. We may employ the matching architecture and weights by leveraging the pre-training framework. Despite the limited dataset, the model may modify the parameters to an optimal form based on the pre-training model. IOU (Intersection over Union) threshold is set to 0.65 in the experiment. After that, we built the object detection models using the trained dataset. Finally, we integrated the trained models into the Apache Spark streaming and used the test dataset to evaluate the performance.

C. Performance Evaluation

During the training phase, the evaluation parameters play an important role in achieving the targeted object detection accuracy. Furthermore, gathering appropriate assessment parameters is a critical component in the differentiation and development of the perfect model. After applying the object detection models to our case studies, We need to test the suggested model's accuracy using several performance assessment indicators such as [44]:

1) Precision (PR): It is utilized to subtract the number of accurately predicted positive patterns from the total number of expected positive patterns in a positive class [45]. Precision can be calculated using the following equation.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

2) Recall (RE): It is used to determine the percentage of correctly categorized positive patterns [45]. Recall can be calculated using the following equation:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

True Positive (TP): It corresponds to the quantity of cases correctly recognized by the classifier [46].

False Positive (FP) It represents the amount of negative incidents that were incorrectly classified as positive cases [46].

False Negative (FN): It relates to the quantity of positive incidents that were incorrectly labelled as negative cases. [46].

True Negative (TN): The number of negative cases successfully categorised by the model is denoted by the true negative values [46].

3) F-Score (FS): Also recognized as the F1-score, this is a statistic used to assess data correctness. In addition, it's employed to investigate binary classification methods that classify data as "positive" or "negative". [45]. The F1-score can be calculated using the following equation.

$$F1 = 2 \times \frac{(precision \times recall)}{precision + recall} \tag{7}$$

4) Intersection over Union (IOU): IoU is calculated by dividing the overlapping region between detection and ground truth by their union region [47]. Fig. 9 illustrates the concept of IoU.

When IoU is 100%, projection boxes and ground truth boundaries perfectly coincide, and the prediction is the highest. Despite 100% IoU is practically impractical to acquire (because of the constraints of present Convolutional networks), an IoU score of 50% to 90% is widely employed in numerous computer vision tasks [48].

In this work, a detection is deemed successful if IoU $\geq$ 65%.

IoU can be calculated using the following equation:

$$IoU(A, B) = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{A \cap B}{A \cup B} \tag{8}$$

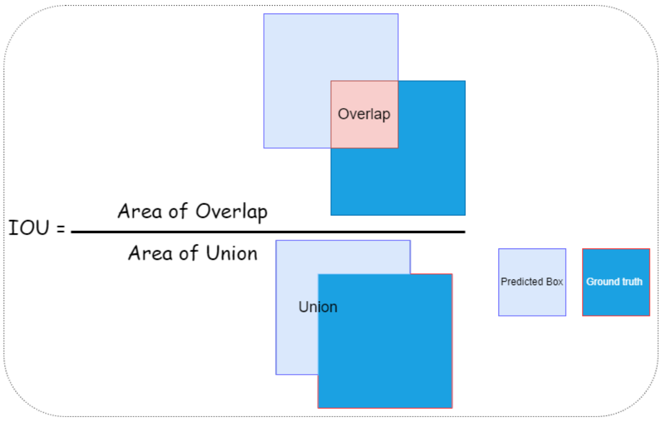

Fig. 9. Intersection over Union (IOU).

5) Average Precision (AP): The Average Precision (AP) is intended to describe the Precision-Recall Curve by summing accuracy over all recall scores ranging from 0 to 1. It's the region underneath the Precision-Recall curve [49]. (AP) can be calculated using the following equation.

$$\mathrm{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{9}$$

6) Mean Average Precision (mAP): It is the average of all specified categories of precision. A more excellent mAP implies that an object detection technique performs well in terms of precision and resilience [49]. (mAP) can be calculated using the following equation:

$$\mathrm{mAP} = \frac{1}{N} \sum AP \tag{10}$$

7) Throughput (TP): It refers to the quantity of outcomes delivered in a particular amount of time.

D. Results

We conducted the experiments locally on the environment set up described in the previous subsection to evaluate our approach.

As shown in Table V, which contains six evaluation metrics named Precision(PR), Recall(RE), IOU, F-Score, mAP and Throughput(TP) for the flood case study after applying data augmentation.

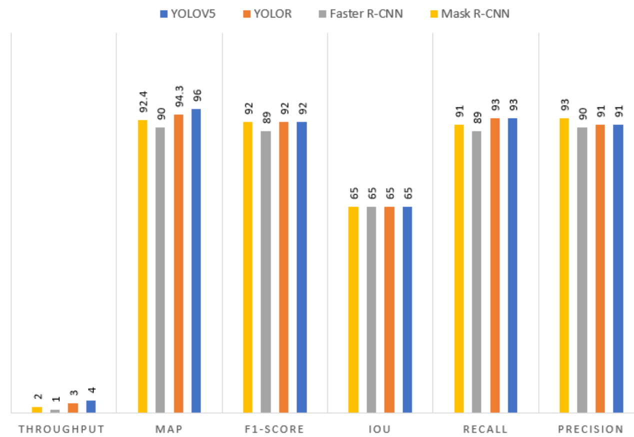

Fig. 10. Results of flood.

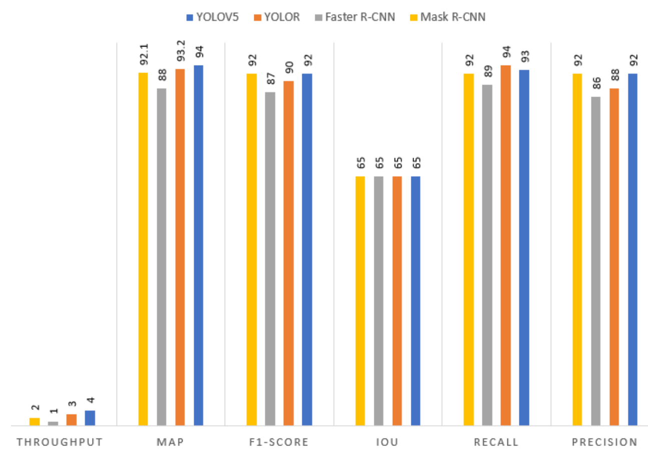

Fig. 11. Results of fire.

Table V illustrates that the results of the one stage models Yolov5 and YoloR are the same for Precision, Recall and F-Measure. However, they vary in terms of mAP score, Yolov5 achieved a very high accuracy of %96 while YoloR achieved %94.3 score. Moreover, the difference in the performance of the two stage models Faster R-CNN and Mask R-CNN is very minimal. However, Mask R-CNN outperform Faster R-CNN in all metrics. Faster R-CNN yields %90 of mAP score while Mask R-CNN yields %92.4.

Table VI shows the performance of the models for the flood case study before applying data augmentation. It can be noted that all the model didn't perform well before applying data augmentation.

Table VII shows the models' performance for the fire case study after applying data augmentation. It can be noted that Yolov5 outperforms all the models in all evaluation metrics except for Recall where YoloR achieved %94 while Yolov5 achieved %93. However, the difference between them is negligible in regards to mAP score, Yolov5 yields %94 while YoloR yields %93.2. On the other hand, the difference between the two stage model Faster R-CNN and Mask R-CNN is noticeable. Mask R-CNN surpass Faster R-CNN in all metrics. Also, Mask R-CNN yields %92.1 of mAP score

TABLE V. RESULTS OF MODELS PERFORMANCE FOR FLOOD CASE STUDY WITH DATA AUGMENTATION

Model	PR	RE	IOU	F-Score	mAP	TP
Yolov5	0.91	0.93	0.65	0.92	0.96	4
YOLOR	0.91	0.93	0.65	0.92	0.943	3
Faster R-CNN	0.90	0.89	0.65	0.89	0.90	1
Mask R-CNN	0.93	0.91	0.65	0.92	0.924	2

TABLE VI. RESULTS OF MODELS PERFORMANCE FOR FLOOD CASE STUDY WITHOUT DATA AUGMENTATION

Model	PR	RE	IOU	F-Score	mAP
Yolov5	0.75	0.77	0.65	0.76	0.82
YOLOR	0.61	0.81	0.65	0.70	0.80
Faster R-CNN	0.66	0.69	0.65	0.67	0.70
Mask R-CNN	0.70	0.72	0.65	0.71	0.79

TABLE VII. RESULTS OF MODELS PERFORMANCE FOR FIRE CASE STUDY WITH DATA AUGMENTATION

Model	PR	RE	IOU	F-Score	mAP	TP
Yolov5	0.92	0.93	0.65	0.92	0.94	4
YOLOR	0.88	0.94	0.65	0.90	0.932	3
Faster R-CNN	0.86	0.89	0.65	0.87	0.88	1
Mask R-CNN	0.92	0.92	0.65	0.92	0.921	2

TABLE VIII. RESULTS OF MODELS PERFORMANCE FOR FIRE CASE STUDY WITHOUT DATA AUGMENTATION

Model	Precision	Recall	IOU	F-Score	mAP
Yolov5	0.74	0.75	0.65	0.74	0.80
YOLOR	0.60	0.80	0.65	0.69	0.78
Faster R-CNN	0.67	0.68	0.65	0.67	0.70
Mask R-CNN	0.68	0.70	0.65	0.69	0.73

which is very minimal to YoloR.

Moreover, as it can be noted from Table VIII, it shows the performance of the models for the fire case study before applying data augmentation. It's obvious that all the model didn't perform well before applying data augmentation.

Fig. 10 shows the performance evaluation of each model for flood dataset model separately while Fig. 11 illustrates the performance evaluation of each model for flood dataset model separately.

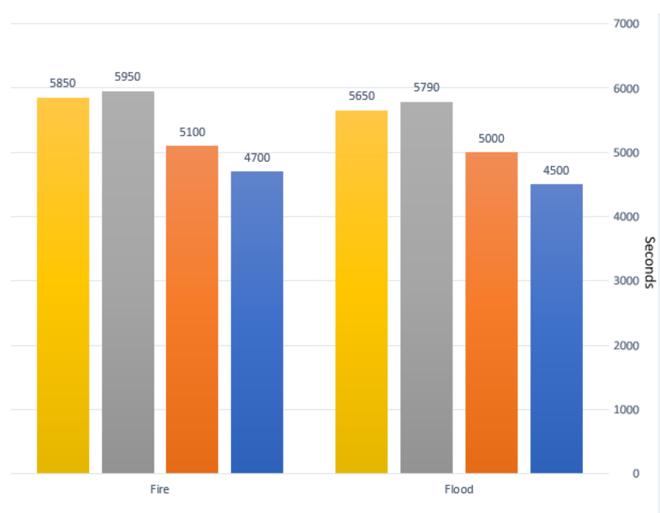

Fig. 12. Models performance using Spark.

Regrading the Throughput score, we applied and run a ranking algorithm to get the scores. After we measured the performance of the models using the test dataset on Spark, we sort out the score results of the models from Spark in ascending order in regards to processing time. Then, the first model (with the least processing time) will get a score of four points which is the highest and the second will get three and the third will get two and the last one will get only one point.

Therefore, Fig. 12 illustrates that Yolov5 model outperform the rest of the models in terms of throughput in both case studies and score four points because it needed less time to process the dataset.

Fig. 12 shows the performance of all the model using Spark.

Table IX shows the results of our location ranking and selection model.

It can be noted that the second case study of South Africa flooding (Location 2) scored seven points in contrast to the first case study building on fire in India (Location 1) which scored only three points. Therefore, the second location was selected as the most urgent location. All Snaps is denoted to all the collected snaps and its score, Emergency is denoted to all the snaps related to emergency cases.

Fig. 13 and 14 depict the the evaluation results of Yolov5 for flood dataset and fire dataset, respectively. In addition, Fig. 15 represents the accuracy score of Mask R-CNN for both datasets. While Fig. 16 shows the accuracy result of YoloR for flood and fire datasets.

TABLE IX. Results of Location Ranking and Selection Algorithm

Location	All Snaps/ Score	Emergency/ Score	mAP/ Score	TP Score	Sum
L1(Fire)	780/ 1	546/ 2	0.918/ 0	0	3
L2(Flood)	759/ 0	524/ 0	0.931/ 3	4	7

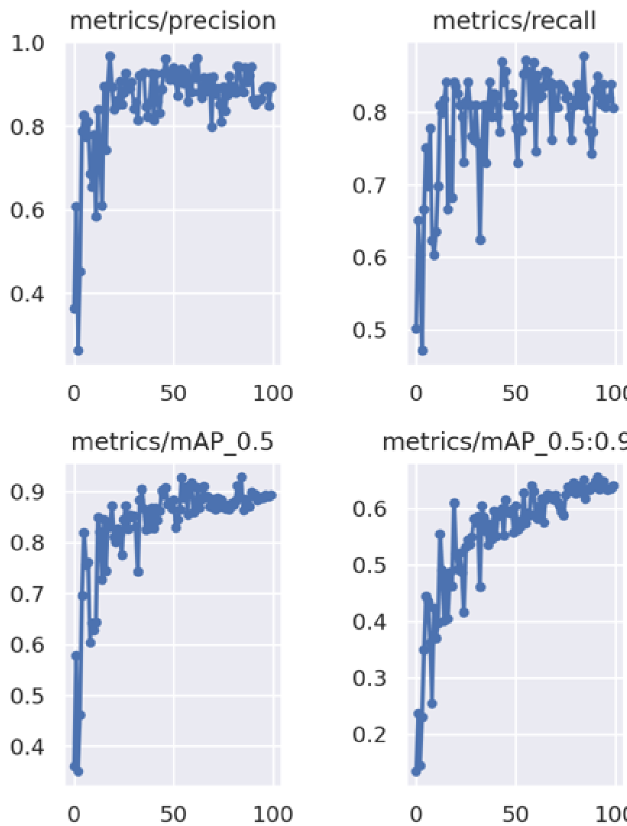

Fig. 13. Yolov5 flood results.

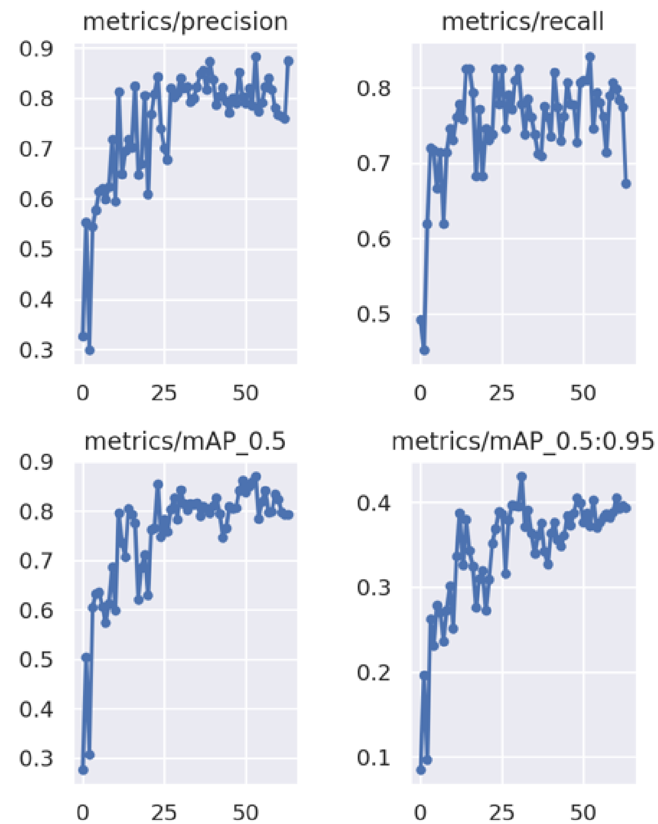

Fig. 14. Yolov5 fire results.

VII. Discussion

The high prevalence of cellphones and social networking platforms such as Snapchat are obviously dissolving traditional barriers between information providers and end-users. It is certainly relevant in emergency events, as individuals on the site produce and exchange real-time information about the event.

However, notwithstanding their demonstrated significance, obtaining event-related information from real-time streams of vast numbers of snaps is a significant challenge.

To address this gap, this paper proposes an automated mapping approach of emergency events and locations based on object detection and social networks. Furthermore, employing object detection methods on social networks to detect emergency events will construct reliable, flexible and speedy approach by utilizing the Snapchat hotspot map as a reliable source to discover the exact location of emergency events.

Moreover, the proposed approach aims to yields high accuracy by employing the state of the arts object detectors to achieve the objectives of this paper. Results show that the proposed approach achieved a very high accuracy of 96% for flood dataset and 94% for fire dataset.

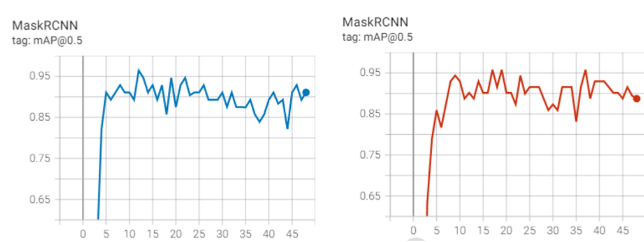

Fig. 15. MaskRCNN.

Among the evaluated models, Yolov5 exhibited the highest performance, proving to be a reliable option for emergency event detection, aligning with previous research indicating that Yolov5 achieves superior accuracy in real-time object detection compared to traditional CNN-based models [50].

Mask R-CNN also demonstrated promising results, particu-

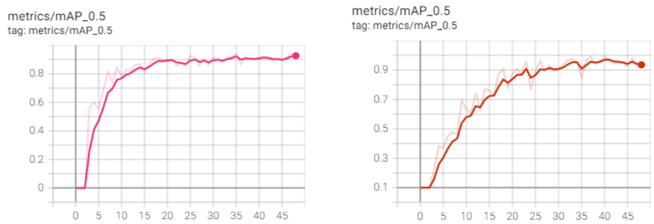

Fig. 16. YOLOR.

larly in terms of learning stability, supporting findings from He et al. [51], which highlight its effectiveness in detecting objects with high precision. However, YoloR struggled with detecting fragmented objects, and Faster R-CNN was the least effective model in this study, consistent with prior work noting Faster R-CNN's computational inefficiency for real-time detection [52] .

VIII. Conclusion

Based on experimentation and testing, the YoloR model did not perform well without applying data augmentation techniques on the fire and flood datasets because the dataset was too small. Without data augmentation, the recall was as high as 0.8, but the precision was merely 0.6.

The model kept predicting numerous small and large boxes, and in doing so, it got many erroneous boxes alongside the ground truth; as a result, it gave many incorrect outputs. Its precision was very low, but since a few of those boxes were the actual boxes, the recall was 0.8. The recall refers to all the actual boxes being detected (irrespective of how many erroneous ones you get in addition). We applied brightness data augmentation where the image was made brighter by 20% or darker by 20% which makes it more stable in different lighting conditions.

By applying data augmentation, the precision increased to 0.88 while recall increased to 0.94, with a total mAP of 0.932 on the fire dataset. A similar augmentation technique was applied to the flood dataset. We noticed that YoloR works better for continuous objects; when we have objects that break or distance between each instance, it does not work very well. Because instead of detecting the whole object, it detects numerous smaller objects. Technically it is correct, but in terms of the mAP score, this could consider a downside.

Moreover, Mask R-CNN performed very well, it was a stable model, and the predictions were similar to other models in terms of accuracy. Mask R-CNN's fascinating side is that its learning curve is better than the other models. If we had more data, Mask R-CNN would do better than other models.

Yolov5 yielded the highest accuracy for both datasets compared to the rest of the models. Even before applying the data augmentation techniques, both datasets' accuracy scores were acceptable for Yolov5. Yolov5 is available in various variants: Yolov5x, Yolov5m, Yolov5l, and Yolov5x, each with deeper layers than the others. We used Yolov5l for both flood and fire datasets in this work.

Faster R-CNN was the least model in terms of performance compared to other models. Before applying data augmentation techniques, its performance was almost the same, and the difference was negligible. However, after applying data augmentation techniques, the performance of Faster R-CNN for the flood dataset was better than that of the fire dataset; it yielded 0.90 and 0.88, respectively.

Future work expected that the performance would improve if more datasets and additional object detection models were applied to the approach. Furthermore, our system shows promising results when combining the Snapchat hotspot map and computer vision approach to detect the location of emergency events.

Declarations

Authors' Contributions

Alfalqi Khalid: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing original draft preparation, writing review and editing, visualization, Bellaiche Martine: discussion, review, supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Competing Interests

The authors declare no conflict of interest.

References

[1] Z. Hu, G. Wu, H. Wu, and L. Zhang, "Cross-sectoral preparedness and mitigation for networked typhoon disasters with cascading effects," *Urban Climate*, vol. 42, p. 101140, 2022.

[2] M. T. Chaudhary and A. Piracha, "Natural disasters—origins, impacts, management," *Encyclopedia*, vol. 1, no. 4, pp. 1101–1131, 2021.

[3] H. J. Caldera and S. Wirasinghe, "A universal severity classification for natural disasters," *Natural hazards*, vol. 111, no. 2, pp. 1533–1573, 2022.

[4] L. Sun and F. You, "Machine learning and data-driven techniques for the control of smart power generation systems: An uncertainty handling perspective," *Engineering*, vol. 7, no. 9, pp. 1239–1247, 2021.

[5] M. Suriya, V. Chandran, and M. Sumithra, "Enhanced deep convolutional neural network for malarial parasite classification," *International Journal of Computers and Applications*, pp. 1–10, 2019.

[6] J. Zhou, X. Tan, Z. Shao, and L. Ma, "Fvnet: 3d front-view proposal generation for real-time object detection from point clouds," in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2019, pp. 1–8.

[7] Y. Song, L. Gao, X. Li, and W. Shen, "A novel robotic grasp detection method based on region proposal networks," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101963, 2020.

[8] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude uav datasets: A survey," *Image and Vision Computing*, vol. 104, p. 104046, 2020.

[9] S. Khatoon, A. Asif, M. M. Hasan, and M. Alshamari, "Social media-based intelligence for disaster response and management in smart cities," in *Artificial Intelligence, Machine Learning, and Optimization Tools for Smart Cities*. Springer, 2022, pp. 211–235.

[10] A. Kumar and J. P. Singh, "Location reference identification from tweets during emergencies: A deep learning approach," *International journal of disaster risk reduction*, vol. 33, pp. 365–375, 2019.

[11] A. Fokaefs and K. Sapountzaki, "Crisis communication after earthquakes in greece and japan: Effects on seismic disaster management," *Sustainability*, vol. 13, no. 16, p. 9257, 2021.

[12] C. Xu, A. S. Ding, and K. Zhao, "A novel poi recommendation method based on trust relationship and spatial–temporal factors," *Electronic Commerce Research and Applications*, vol. 48, p. 101060, 2021.

[13] J. P. Singh, Y. K. Dwivedi, N. P. Rana, A. Kumar, and K. K. Kapoor, "Event classification and location prediction from tweets during disasters," *Annals of Operations Research*, vol. 283, no. 1, pp. 737–757, 2019.

[14] E. Rodrigues, R. Assunção, G. L. Pappa, D. Renno, and W. Meira Jr, "Exploring multiple evidence to infer users' location in twitter," *Neurocomputing*, vol. 171, pp. 30–38, 2016.

[15] N. Duong-Trung, N. Schilling, and L. Schmidt-Thieme, "Near real-time geolocation prediction in twitter streams via matrix factorization based regression," in *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 1973–1976.

[16] J. Gelernter and S. Balaji, "An algorithm for local geoparsing of microtext," *GeoInformatica*, vol. 17, no. 4, pp. 635–667, 2013.

[17] D. Rout, K. Bontcheva, D. Preoţiuc-Pietro, and T. Cohn, "Where's@ wally? a classification approach to geolocating users based on their social ties," in *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, 2013, pp. 11–20.

[18] F. Laylavi, A. Rajabifard, and M. Kalantari, "Event relatedness assessment of twitter messages for emergency response," *Information processing & management*, vol. 53, no. 1, pp. 266–280, 2017.

[19] M. Ji, L. Liu, R. Du, and M. F. Buchroithner, "A comparative study of texture and convolutional neural network features for detecting collapsed buildings after earthquakes using pre-and post-event satellite imagery," *Remote Sensing*, vol. 11, no. 10, p. 1202, 2019.

[20] H. Miura, T. Aridome, and M. Matsuoka, "Deep learning-based identification of collapsed, non-collapsed and blue tarp-covered buildings from post-disaster aerial images," *Remote Sensing*, vol. 12, no. 12, p. 1924, 2020.

[21] Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection," in *2016 International Forum on Management, Education and Information Technology Application*. Atlantis Press, 2016, pp. 568–575.

[22] T. Ci, Z. Liu, and Y. Wang, "Assessment of the degree of building damage caused by disaster using convolutional neural networks in combination with ordinal regression," *Remote Sensing*, vol. 11, no. 23, p. 2858, 2019.

[23] Y. Pi, N. D. Nath, and A. H. Behzadan, "Convolutional neural networks for object detection in aerial imagery for disaster response and recovery," *Advanced Engineering Informatics*, vol. 43, p. 101009, 2020.

[24] W. Zhang, P. Tang, and L. Zhao, "Remote sensing image scene classification using cnn-capsnet," *Remote Sensing*, vol. 11, no. 5, p. 494, 2019.

[25] M.-T. Pham, L. Courtrai, C. Friguet, S. Lefèvre, and A. Baussard, "Yolo-fine: One-stage detector of small objects under various backgrounds in remote sensing images," *Remote Sensing*, vol. 12, no. 15, p. 2501, 2020.

[26] J. J. Yebes, D. Montero, and I. Arriola, "Learning to automatically catch potholes in worldwide road scene images," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 3, pp. 192–205, 2020.

[27] S. N. K. B. Amit, S. Shiraishi, T. Inoshita, and Y. Aoki, "Analysis of satellite images for disaster detection," in *2016 IEEE International geoscience and remote sensing symposium (IGARSS)*. IEEE, 2016, pp. 5189–5192.

[28] U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, no. 2, p. 464, 2022.

[29] E. Kizilay and İ. Aydin, "A yolor based visual detection of amateur drones," in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*. IEEE, 2022, pp. 1446–1449.

[30] B. Liu, J. Luo, and H. Huang, "Toward automatic quantification of knee osteoarthritis severity using improved faster r-cnn," *International journal of computer assisted radiology and surgery*, vol. 15, no. 3, pp. 457–466, 2020.

[31] X. Xu, M. Zhao, P. Shi, R. Ren, X. He, X. Wei, and H. Yang, "Crack detection and comparison study based on faster r-cnn and mask r-cnn," *Sensors*, vol. 22, no. 3, p. 1215, 2022.

[32] M. Liu, J. Dong, X. Dong, H. Yu, and L. Qi, "Segmentation of lung nodule in ct images based on mask r-cnn," in *2018 9th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2018, pp. 1–6.

[33] J.-H. Lee, D.-h. Kim, S.-N. Jeong, and S.-H. Choi, "Diagnosis and prediction of periodontally compromised teeth using a deep learning-based convolutional neural network algorithm," *Journal of periodontal & implant science*, vol. 48, no. 2, pp. 114–123, 2018.

[34] D. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary intelligence*, pp. 1–22, 2021.

[35] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021.

[36] H. Lamba, S. Srikanth, D. R. Pailla, S. Singh, K. S. Juneja, and P. Kumaraguru, "Driving the last mile: Characterizing and understanding distracted driving posts on social networks," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 393–404.

[37] K. Alfalqi and M. Bellaiche, "An emergency event detection ensemble model based on big data," *Big Data and Cognitive Computing*, vol. 6, no. 2, p. 42, 2022.

[38] R. Walambe, A. Marathe, and K. Kotecha, "Multiscale object detection from drone imagery using ensemble transfer learning," *Drones*, vol. 5, no. 3, p. 66, 2021.

[39] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, pp. 1–27, 2021.

[40] J. Lee, W. Wang, F. Harrou, and Y. Sun, "Reliable solar irradiance prediction using ensemble learning-based models: A comparative study," *Energy Conversion and Management*, vol. 208, p. 112582, 2020.

[41] https://www.reuters.com/world/india/building-fire-kills-27-new-delhi\protect\penalty-\@M-police-arrest-company-owners-2022-05-14/, accessed: 2025-2-2.

[42] D. floods, "Durban floods: South Africa floods kill more than 300," *BBC News*, Apr. 2022. [Online]. Available: https://www.bbc.com/news/world-africa-61092334

[43] O. L. F. De Carvalho, O. A. de Carvalho Junior, C. R. e. Silva, A. O. de Albuquerque, N. C. Santana, D. L. Borges, R. A. T. Gomes, and R. F. Guimarães, "Panoptic segmentation meets remote sensing," vol. 14, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/4/965

[44] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[45] J. Miguel, S. Caballé, F. Xhafa, and J. Prieto, "A massive data processing approach for effective trustworthiness in online learning groups," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 8, pp. 1988–2003, 2015.

[46] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview. arxiv preprint. 2020: 1–17," 2022.

[47] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/3/279

[48] Y. Pi, N. D. Nath, and A. H. Behzadan, "Convolutional neural networks for object detection in aerial imagery for disaster response and recovery," *Advanced Engineering Informatics*, vol. 43, p. 101009, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034619305828

[49] N. Ottakath, O. Elharrouss, N. Almaadeed, S. Al-Maadeed, A. Mohamed, T. Khattab, and K. Abualsaud, "Vidmask dataset for face mask detection with social distance measurement," *Displays*, vol. 73, p. 102235, 2022.

[50] R. Khanam and M. Hussain, "What is yolov5: A deep look into the internal features of the popular object detector," 2024. [Online]. Available: https://arxiv.org/abs/2407.20892

[51] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[52] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” 2016. [Online]. Available: https://arxiv.org/abs/1506.01497