UGEN: UAV-based GAN-aided Ensemble Network for Efficient Post Disaster Survivor Detection

***Abstract*—Post-disaster scene understanding frameworks are becoming increasingly crucial in search and rescue operations and damage assessment initiatives. The use of Unmanned Aerial Vehicles (UAVs) provides an efficient method to complete the task of scene understanding. However, complex environments in post-disaster scenarios make it difficult for UAVs to accurately detect humans or objects. Moreover, inefficient object detection mechanisms lead to low accuracy and a long time for object detection tasks. Hence, to mitigate these issues, we propose a UAV-based GAN-aided Ensemble Network (UGEN) for efficient post-disaster survivor detection, involving a semantic segmentation mechanism. This approach deploys a Context-Conditional Generative Adversarial Network (CCGAN)-based model to remove occlusion in the images obtained from the UAVs. The framework classifies entities present in the visual scope of the UAV using a semantic segmentation framework deployed on the CCGAN-enhanced images, resulting in a pixel-level prediction of entities present in the post-disaster images. Furthermore, an ensemble network comprising a combination of single-stage and multi-stage detectors detects survivors present in the post-disaster scenario, thereby combining the benefits of both architectures, resulting in a reduced false negative rate and improved performance. The proposed model achieved a survivor detection accuracy of 96.4%.**

***Index Terms*—Unmanned Aerial Vehicles, Generative Adversarial Networks, Semantic Segmentation, Ensemble Learning**

I. INTRODUCTION

Post-disaster scene understanding frameworks are becoming increasingly crucial in search and rescue operations and damage assessment initiatives [1]. As the number of natural disasters continues to rise, the importance of efficient and accurate disaster response has become paramount [2]. The use of Unmanned Aerial Vehicles (UAVs) provides an efficient and cost-effective method to complete the task of scene understanding [3]. Moreover, many deep learning strategies have been deployed to effectively execute visual detection from camera sensors mounted on UAVs [4]. However, the complex environments present in post-disaster scenarios make it difficult for UAVs to accurately detect humans or objects. Additionally, inefficient object detection mechanisms lead to low accuracy and a long inference time for object detection tasks, which can be particularly problematic in urgent search and rescue situations. Survivors being small objects in post-disaster UAV images makes the task of survivor detection using traditional techniques daunting [5]. Furthermore, survivors who are occluded in the image due to debris or damaged buildings covering them will make the survivor detection task further challenging.

The main objective of the proposed UGEN system is to mitigate the aforementioned issues using a UAV-based scene understanding scheme involving a GAN-aided semantic segmentation mechanism. This approach classifies objects present in the visual scope of the UAV by removing occlusion present in the images and executing pixel-level prediction to classify entities present in the images. By leveraging the power of GANs, the proposed system can better handle the challenges of post-disaster environments and improve the accuracy of object detection. A Context-Conditional GAN (CCGAN)-based denoiser results in images having lower occlusion and optimal brightness, thereby highlighting the important features of the object. Furthermore, using GAN improves the detection of small and dense objects [6], which is the case of survivors in images obtained from a UAV, resulting in the visual regeneration of occluded survivors. Semantic segmentation on the CCGAN-enhanced images leads to a pixel-level prediction of various entities or objects present in the image, thereby generating a corresponding color coding for each entity. The ensemble model, a hybrid architecture consisting of single-stage and multi-stage detectors, is to be implemented to detect the presence of survivors. The network will overcome the disadvantages of both single-stage and multi-stage frameworks. The envisioned framework deploys an ensemble network comprising the YOLOv5 and Faster R-CNN frameworks, thereby improving the performance of survivor detection while decreasing the false negative rate. Deploying the proposed UGEN framework increases the accuracy and efficiency of the survivor detection task, thereby significantly enhancing the effectiveness of post-disaster scene understanding and resulting in successful Search And Rescue operations.

The key contributions of this paper include:

1) A CCGAN-based denoiser and occlusion remover mechanism will be implemented to improve the detection of survivors using post-disaster UAV images.
2) A semantic segmentation mechanism will be deployed on the CCGAN-enhanced images to classify various entities, thereby improving survivor detection.
3) A hybrid ensemble network comprising single-stage and multi-stage detectors will be developed for survivor detection using the color coding and classification of semantic entities. This will result in the decrease of the high false negative rate of the multi-stage mechanism and the improvement in the performance of the single-stage detector.

The remainder of this paper is organized as follows. Section II consists of a summary of the related works. Section III, IV, and V describe the proposed work and its components. Section VI evaluates the hybrid ensemble network-based model's survivor detection performance and outlines the results of the overall preprocessing module comprising the CCGAN-based occlusion remover and semantic segmentation. Section VII brings the paper to a close by concluding the proposed work.

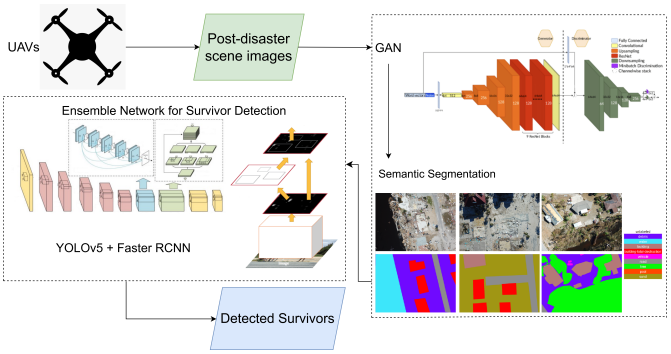


Fig. 1: High-Level Architecture

II. RELATED WORKS

Post-disaster survivor detection is an important task in search and rescue operations. However, survivors being hard to detect objects, especially from sensors mounted on UAVs, require specialized techniques to be effectively detected. Various Convolutional Neural Network (CNN)-based models have been developed for efficient object detection. With the goal of lowering the high false negative rate of multi-stage detectors and improving the quality of the single-stage detector proposals, [7], [8] propose different ensemble networks that combines a multi-stage method with a single-stage detector through ensembling for effective object detection. But according to the investigation, detecting objects in drone images is more challenging than detecting them in images that were taken from the ground. Hence, the accuracy of the model trained on UAV images is still low compared to models trained on ground images. Several other deep learning techniques used for the detection of objects from UAVs have been studied in [9], wherein various deep learning architectures have been outlined, including generative adversarial networks, autoencoders, recurrent neural networks, and convolutional neural networks, and their contributions to the challenge of improving vehicle detection. However, videos captured in the UAVs are sent to on-ground workstations or to the cloud for processing rather than being implemented on the UAV itself, thereby leading to the absence of a lightweight system for efficient real-time detection.

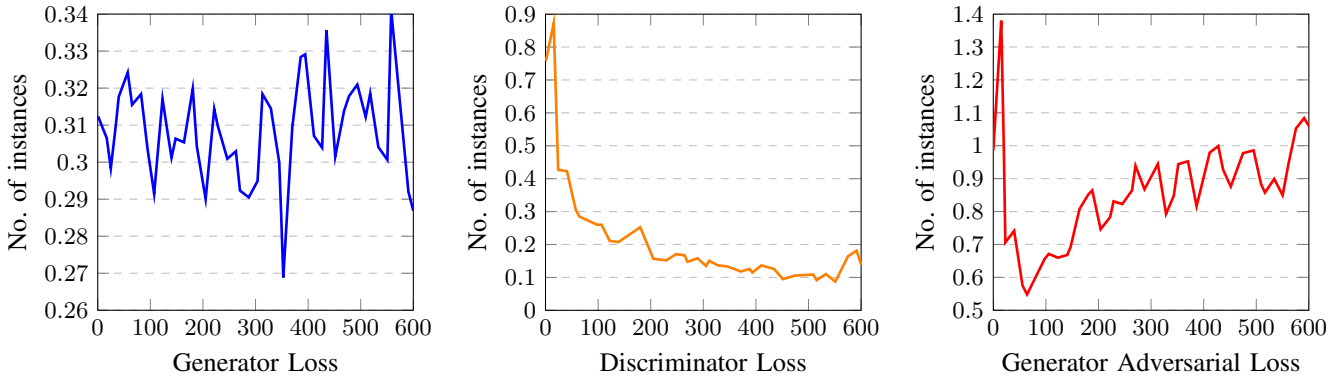
GANs are being increasingly deployed in many modern detection algorithms due to their extensive applications, like the removal of occlusion present in images. [10], [11] discuss denoising and occlusion-removal strategies empowered by GAN-based systems for better image recognition, wherein image regeneration with efficient removal of unnecessary entities present in the image leads to better object detection performance. Many UAV-driven and synthetic datasets have been generated for analyzing efficient methods for survivor detection using UAVs. [12] introduces a high-resolution post-disaster UAV dataset named RescueNet, which contains comprehensive pixel-level annotation of images for semantic segmentation to assess the damage and detect survivors after a natural disaster. However, smaller objects like “vehicles” and “pools” make it difficult to get a good segmentation compared to larger objects like buildings and roads. [13] proposes the UAV-Human dataset for understanding human action, pose, and behavior. The proposed UAV-Human contains 67,428 multi-modal video sequences, 119 subjects for action recognition, 22,476 frames for pose estimation, 41,290 frames, 1,144 identities for person re-identification, and 22,263 frames for attribute recognition which encourages the exploration and deployment of various data-intensive learning models for UAV-based human behavior understanding. However, The UAV-Human dataset poses a limitation for attribute recognition because the dataset is captured over a relatively long period of time.

Semantic segmentation classifies all entities present in an object through pixel-level prediction. This enables the detection model to easily recognize objects present in an image. The authors of [14], [15] propose and evaluate self-attention segmentation models on new high-resolution datasets, namely HRUD and UAVid. However, HRUD is a very challenging dataset due to its variable-sized classes along with similar textures among different classes. Debris, textures of debris, sand, and building with destruction damage make a great impact on the segmentation performance of the evaluated network models. Furthermore, [16], [17] discus in detail various semantic segmentation frameworks used for entity separation and classification in UAV-driven images. 3D rendering of images is also being done in an extensive manner for better detection of objects in a scene. [18], [19] implement 3D imaging mechanisms using 2D images obtained from a swarm of UAVs, wherein a 3D imaging of a scene by the usage of 2D images obtained from several UAVs present in the swarm is implemented at different perspectives with a few points of overlap. But a considerable amount of data must be transmitted from the UAV swarm, as images obtained from each node in the swarm are used to produce the 3D rendering. Multiple UAVs also need to exchange information in order to efficiently collect data on the scenario.

Hence, to mitigate the aforementioned limitations of currently existing systems for survivor detection, we propose an efficient post-disaster survivor detection framework using UAVs that encompasses a GAN-based occlusion removal mechanism to improve the detection of survivors, a semantic segmentation mechanism on the 3D model that classifies various entities in the scene to improve survivor detection, and a hybrid ensemble network comprising single-stage and multi-stage detectors for survivor detection using the color coding and classification of semantic entities. This will result in the decrease of the high false negative rate of the multi-stage mechanism and the improvement in the performance of the single-stage detector, which in turn leads to an efficient survivor detection model for Search and Rescue operations.

III. GAN-BASED OCCLUSION REMOVAL MECHANISM

The proposed mechanism aims to serve as an efficient methodology to detect the presence of survivors in post-disaster scenes, thereby aiding Search-And-Rescue operations. The overall mechanism incorporates a Context Conditional GAN (CCGAN)-based occlusion remover that regenerates occluded survivors present in the post-disaster scene from images



(a) (b) (c)

Fig. 2: GAN-based Occlusion Removal Mechanism: (a) Generator Loss curve, (b) Discriminator Loss curve, (c) Generator Adversarial Loss curve

Algorithm 1 Occlusion Removal and Entity Separation

Input: Images (λ) of post-disaster scene obtained from UAV
Output: Images with classified entities (Φ)

procedure GAN_DENOISING(λ)
  Pre-processing of UAV images λ
  $\lambda_{\text{epochs}} \leftarrow$ Number of iterations to train GAN
  $\lambda_{\text{batchsize}} \leftarrow$ Number of images to train per epoch
  for β `in` λ **do**
    $\vartheta \leftarrow TrainGAN(\beta, \lambda_{\text{epochs}}, \lambda_{\text{batchsize}})$
  end for
  $\varpi \leftarrow GAN\_denoiser(\vartheta)$
  $\varepsilon \leftarrow CCGAN(\varpi)$
  return ε
end procedure
procedure SEMANTIC_SEGMENTATION(ϖ)
  $\Phi[] \leftarrow$ entities present in ϖ
  $\nu \leftarrow 0$
  for α `in` ϖ **do**
    $\Phi[\nu] = Classify(\alpha)$
    $\nu = \nu + 1$
  end for
  Embed Φ in λ
  return λ
end procedure

captured from a swarm of UAVs. Furthermore, the model implements semantic segmentation to produce and entity-wise color coding for efficient human classification and reduced ambiguity for survivor detection. On top of that, we propose a hybrid single-stage and multi-stage-based ensemble network for efficient survivor detection. Fig. 1 describes the workflow of the proposed framework for survivor detection.

Post-disaster images taken from a UAV comprise several issues, namely noise, distortion, and poor clarity. However, the most crucial problem to be eliminated from UAV images of post-disaster scenes is object occlusion. Occlusion is the phenomenon in which objects of interest are covered or masked by other objects, noise, or other characteristics present in the image. Hence, occlusion removal is essential for efficient survivor detection in post-disaster scenarios to be able to accurately detect occluded survivors. To be able to remove occlusion present on human targets present in an image, we propose a CCGAN-based occlusion removal mechanism. The GAN-based occlusion removal framework is deployed to be able to regenerate occluded human targets, thereby resulting in better survivor detection. The GAN model can also be used to denoise images used to train the survivor detection model. Algorithm 1 discusses the usage of GAN for the particular use case. Furthermore, pre-processing images to be able to remove distortions is essential to prepare them for efficient survivor detection, the next phase of the proposed framework, which in turn is done using OpenCV.

$$L_G = \lambda_{L_1} L_1 + \lambda_{adv} L_{adv} \tag{1}$$

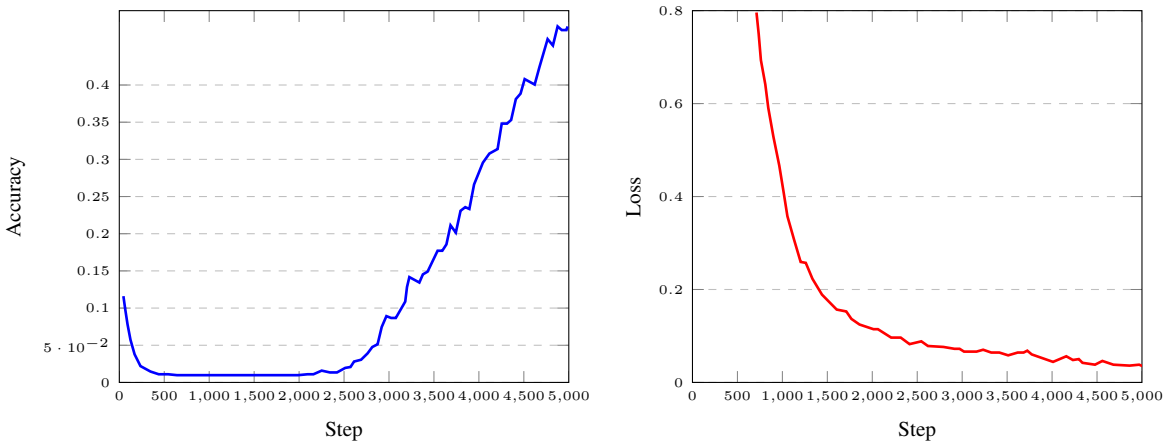
where L_1 is the L1 distance between the generated output image and the real output image, L_{adv} is the GAN loss function, λ_{L_1} and λ_{adv} are the hyperparameters that control the relative importance of the L1 loss and the GAN loss, respectively.

In order to facilitate the rendering of the image devoid of occlusions which are bound to occur in the case of panoramic disaster stills posing a hindrance to the detection of survivors, a Context-Encoder GAN is proposed. Context-Encoder GANs most commonly find their use in Inpainting problems wherein a portion of the image is either to be removed or generated which specifically finds its use here in the removal of obstacles that pose occlusions in the disaster images. The Context-Encoder GAN possesses an Auto Encoder-based structure for the generator with an Encoder and a Decoder both involving Convolutional Neural Networks consisting of Conv2D operations wherein there is a periodic increase of the number of operations followed by a bottleneck layer bridging the Encoder and Decoder thus resulting in the inception of the term Context Encoder as the encoded context of the image is stored.

$$\Gamma_{rec} = \left\| P - CE(X^{'}) \right\|_2^2 \tag{2}$$

where Γ_{rec} is the reconstruction loss, P is the original region before damage, CE is the model and $X^{'}$ is the entire image that needs to be inpainted.

$$\Gamma = \lambda_{adv} \Gamma_{adv} + \lambda_{rec} \Gamma_{rec} \tag{3}$$



(a) (b)

Fig. 3: Semantic Segmentation Performance Evaluation: (a) Step-Accuracy curve, (b) Step-Loss curve

where Γ is the overall loss, λ_{adv} and λ_{rec} are the coefficients for tuning the influence of each of the losses. The Decoder similarly is made up of Conv2D operations termed as deconvolutions as the number of channels is continuously reducing. Reconstruction Loss or L2 loss is the metric most commonly used to judge the performance of the generator and is also termed generator loss. Fig. 2 depicts the performance evaluation done on the trained GAN-based occlusion Removal model. Fig. 3 portrays a sample of the output obtained from the trained CCGAN model.

Algorithm 2 Hybrid Ensemble Network

Input: Enhanced Images containing entities(ϕ) of post-disaster scene obtained from UAV
Output: Images with Detected Survivors (λ)

procedure ENSEMBLE_NETWORK(Φ)
  for x_n in Φ **do**
    $YOLOv5()$ model deployed for each image
    $\lambda \leftarrow$ CCGAN-enhanced image
    $\phi \leftarrow YOLOv5(\lambda)$
    if $classDetected(\phi) ==$ human or car **then**
      Bounding box τ created
    else
      $\lambda' \leftarrow$ another CCGAN-enhanced image
      $\phi' \leftarrow YOLOv5(\lambda)$
    end if
    $FasterRCNN()$ model deployed for each image
    $\lambda \leftarrow \phi$
    $\phi \leftarrow FasterRCNN(\lambda)$
    if $classDetected(\phi) ==$ human or car **then**
      Bounding box ϵ created
      ϵ embedded with ϕ
    else
      $\lambda' \leftarrow$ another CCGAN-enhanced image
      $\phi' \leftarrow FasterRCNN(\lambda)$
    end if
  end for
  Apply simple averaging to ensemble results in ϕ
  $\eta \leftarrow 1/N\Sigma_i f_i(\phi)$
  return η
end procedure

IV. Semantic Segmentation

A post-disaster image taken from a UAV will comprise several objects or entities. For the task of survivor detection, human survivors and cars in which survivors may be present are the only entities of interest for the succeeding survivor detection model. Hence, all other entities present in the scene are unnecessary for survivor detection and may lead to inaccurate detection of survivors due to ambiguity caused by similar entities present in the 3D model. Hence, to mitigate the problem of ambiguity and improve survivor detection performance and accuracy, we propose a semantic segmentation mechanism atop the CCGAN-based image-enhancement mechanism to be able to differentiate between various entities present in the post-disaster UAV image. Semantic segmentation deploys pixel-level prediction of images and categorizes and classifies the various entities present in the image.

$$Y = f(X;\Theta) = W^T * g(V;\theta) + b \tag{4}$$

where Y is the output segmentation map, X is the input image, is the set of network parameters for the convolutional layers, V is the input to the decoder, θ is the set of network parameters for the decoder, W is the weight matrix, b is the bias term, and $g()$ represents the upsampling layers in the decoder. While implementing pixel-level entity classification, color coding is generated for each entity observed in the image. Unlike instance segmentation wherein each instance of the same entity is highlighted in a different color, all entities present in the image are given the same color coding in this scenario. This makes it easier to separate all entities present in the image, thereby highlighting human survivors alone in the survivor detection model.

For the detection of survivors, the SegFormer model has been deployed. Being a semantic segmentation model composed of several key components, SegFormer incorporates a Transformer encoder with self-attention layers to capture relationships between image patches, enabling spatial dependencies to be learned across the entire image. The input image is divided into non-overlapping patches, which are linearly embedded into a lower-dimensional feature space.

Additionally, position embeddings are added to the patch embeddings, providing spatial cues and preserving the image's spatial structure. The patch and position embeddings pass through transformer encoder layers and then into a segmentation head, which employs convolutional layers to predict semantic segmentation masks. Furthermore, a decoder and upsampling module refine the segmentation predictions and upsample the feature maps to the original input resolution, thereby collectively contributing to the model's effectiveness in semantic segmentation tasks. The loss function is taken as an important characteristic while training the SegFormer model for semantically classifying entities present in the post-disaster image.

$$L = -1/N\Sigma_i\Sigma_j[y_{ij} * log(\hat{y}_{ij}) + (1 - y_{ij}) * log(1 - \hat{y}_{ij})] \quad (5)$$

where L is the loss associated with the SegFormer model, N is the total number of pixels in the image, y_{ij} is the ground truth label for pixel (i,j), $\hat{y}_{ij}$ is the predicted label for pixel (i,j), and the summation is taken over all pixels in the image. The attention mechanism used in the model is of vital importance in the working of the SegFormer framework.

$$A = softmax(QK^T/d)V \quad (6)$$

where Q, K, and V are the query, key, and value matrices, respectively, and d is the dimension of the embedding space. The softmax function is applied to the dot product of Q and K transposed divided by the square root of d, and the result is multiplied by V to obtain the attention output A. The classes of importance, namely car and people, were classified efficiently through entity-wise color coding generated by the SegFormer module.

V. Hybrid Ensemble Network

The images received as output from the semantic segmentation module were fed as input to the hybrid ensemble network for detecting the presence of survivors present in the images.

For survivor detection, we propose a hybrid single-stage and multi-stage detector combination as an ensemble model for object detection. Both single-stage and multi-stage detectors have advantages and disadvantages when used as standalone object detection models. The main disadvantages include the high false-negative rate of multi-stage detectors and the high training and inference time of single-stage detectors. We propose that the deployment of an ensemble model for survivor detection will nullify the disadvantages of both systems, thereby decreasing the false-negative rate but maintaining the low training and inference time. Algorithm 2 describes the proposed hybrid ensemble network.

$$Y = 1/N\Sigma_i f_i(X; \Theta_i) \quad (7)$$

where Y is the set of predicted bounding boxes and class labels, N is the number of models in the ensemble, $f_i()$ represents the i-th model in the ensemble, X is the input image, and $_i$ is the set of network parameters for the i-th model.

In the proposed model, we use the YOLOv5 framework as the single-stage detector and the Faster RCNN mechanism for the multi-stage framework of choice. The training loss of the YOLOv5 model is to be reduced in order to obtain maximum throughput in terms of performance.

$$L = \lambda_r L_r + \lambda_c L_c + \lambda_a L_a \quad (8)$$

where λ_r, λ_c, and λ_a are hyperparameters that control the relative importance of the different components of the loss function, L_r is the regression loss that penalizes the difference between the predicted and ground truth bounding box coordinates, L_c is the classification loss that penalizes the difference between the predicted and ground truth class probabilities, and L_a is the anchor loss that encourages the model to predict anchors that match the ground truth objects. The regression loss L_r is taken into consideration with vital importance while trying to minimize the overall loss of the detection model.

$$L_r = \lambda_{xy}\Sigma_i^k i = 1\Sigma_j \in (x, y)(B_{i,j} - t_{i,j})^2 + \lambda_{wh}\Sigma_i^k i = 1\Sigma_j \in (w, h)(\sqrt{B_{i,j}} - \sqrt{t_{i,j}})^2$$

where $t_{i,j}$ is the corresponding ground truth value for the j-th component of the i-th bounding box, λ_{xy} and λwh are hyperparameters that control the relative importance of the x-y and w-h components of the regression loss, and the summations are taken over all predicted objects and all components of the bounding box coordinates. The YOLOv5 model uses anchor boxes to predict bounding box coordinates, which act as a set of k clusters of bounding boxes computed using k-means clustering on the ground truth bounding boxes. The anchor loss is used to encourage the model to predict anchor boxes that match the ground truth objects.

$$L_a = \lambda_a\Sigma_j(1 - IOU(B_j, A_{mj}))^2 \quad (9)$$

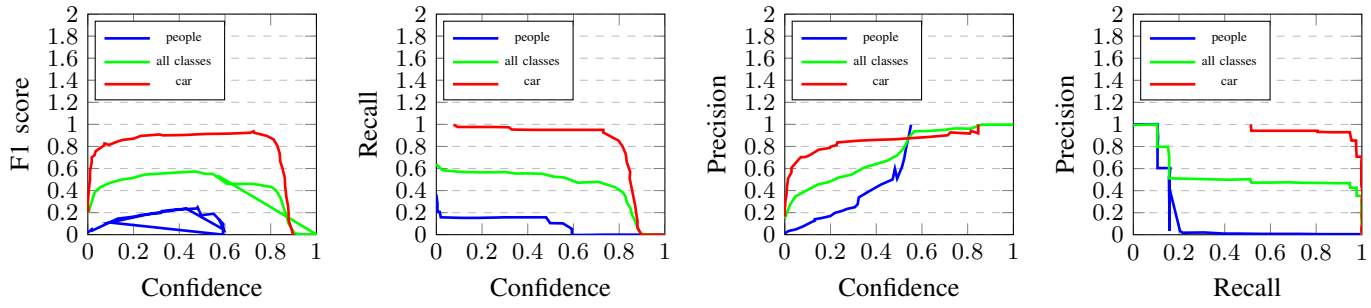
where A_{mj} is the m-th anchor box that best matches the j-th ground truth object, and $IOU(B_j, A_{mj})$ is the intersection over union (IOU) between the predicted and ground truth bounding boxes. $_a$ is a hyperparameter that controls the relative importance of the anchor loss.

Similar to the YOLOv5 model, the Faster CNN model is trained to reduce the training loss of the model to optimize it for efficient survivor detection.

$$L = L_{rpn} + L_{roi} \quad (10)$$

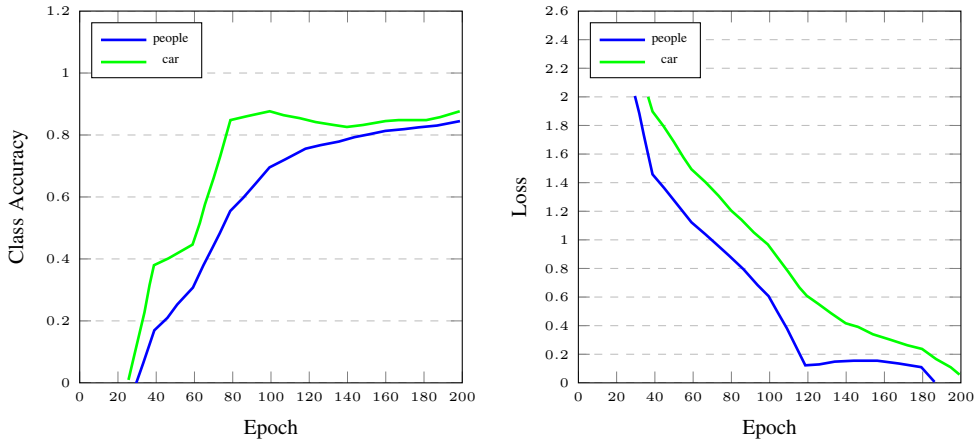
where L_{rpn} is the loss function for the RPN that encourages it to generate accurate proposals, and L_{roi} is the loss function for the second-stage network that classifies the proposals and refines their bounding box coordinates. The RPN generates a set of object proposals by sliding a small window (called an anchor) over the feature map output by the backbone network. Each anchor is associated with a set of scores that indicate the likelihood that it contains an object and the accuracy of its bounding box coordinates. The RPN loss function encourages the network to generate accurate scores and coordinates for the positive proposals, which are those that have high overlap with a ground truth object, and to suppress the scores for the negative proposals that have low overlap with any ground truth object.

$$L_{rpn} = L_{obj} + \lambda_{reg} L_{reg} \quad (11)$$



(a) (b) (c) (d)

Fig. 4: (a) YOLOv5 F1-Confidence curve, (b) YOLOv5 Recall-Confidence curve, (c) YOLOv5 Precision-Recall curve, (d) YOLOv5 Precision-Confidence curve



(a) (b)

Fig. 5: Faster RCNN Performance Evaluation: (a) Epoch-Class Accuracy curve, (b) Epoch-Loss curve

where L_{obj} is the binary cross-entropy loss for objectness classification, L_{reg} is the smooth L1 loss for bounding box regression, and λ_{reg} is a hyperparameter that controls the relative importance of the two components. The second-stage network takes the proposals generated by the RPN and classifies them into one of the object classes and refines their bounding box coordinates. The loss function for the second-stage network consists of two components: the classification loss and the bounding box regression loss.

$$L_{roi} = L_{cls} + \lambda_{reg} L_{reg} \tag{12}$$

where L_{cls} is the cross-entropy loss for classification, L_{reg} is the smooth L1 loss for bounding box regression, and λ_{reg} is a hyperparameter that controls the relative importance of the two components.

Both models are trained, implemented and tested as standalone mechanisms for survivor detection, and an ensemble network combining the two models were implemented as well, followed by the evaluation of the same. Extensive testing of the hybrid ensemble survivor detection mechanism revealed the accuracy of survivor detection to be 94.92% without the use of the GAN-aided 3D reconstruction mechanism. Fig. 3 showcases the output of the survivor detection model based on the hybrid ensemble mechanism.

VI. RESULTS AND DISCUSSIONS

A. Experimental Analysis

For the training of the survivor detection models, the RescueNet dataset was used. It is a newly introduced dataset for the task of search and rescue in disaster scenarios. This dataset comprises 10,000 synthetic images and 100 real-world images, designed to provide a comprehensive training and evaluation platform for researchers working on computer vision and robotics applications in the domain of search and rescue. The main classes of interest were cars and people, which if found in an image, were annotated using the Roboflow software. Roboflow is a multi-model annotation and data augmentation tool. Once the images were annotated, the annotations were extracted in the COCO JSON and YOLOv5 PyTorch versions for use in respective training procedures, The models were trained using Google Colab as the platform, wherein a Tesla P100 GPU was deployed to improve the efficiency of training, thereby improving performance and reducing training time. Standard libraries like TensorFlow and PyTorch were made use of to implement the survivor detection models.

B. CCGAN Occlusion Remover

The Context Conditional GAN framework was used to remove occlusions present in images, especially that pertaining

to survivors present in the images, thereby regenerating them to improve detection accuracy. All necessary libraries were imported to implement the CCGAN architecture, followed by which the configuration for setting up and training the generator and the discriminator were defined. The Image-Dataset class was defined, enabling the loading and usage of a custom dataset to be used for training purposes. Followed by this, the testing and training data loaders were defined for loading the dataset using the ImageDataset class. The generator and the discriminator classes were defined, enabling the creation of the CCGAN model for occlusion removal. The configurations defined earlier were made use of during this stage. The CCGAN model was then trained by first instantiating a model using the classes declared previously and training the same on the RescueNet dataset for occlusion removal. The training results and performance evaluation were then printed as output to visualize the efficiency of the trained CCGAN model for occlusion removal. Fig. 2 portrays the performance statistics of the trained CCGAN model, thereby showcasing its low error rate in removing occlusions. Fig. 2(a) depicts the generator loss plotted against the number of iterations through which the CCGAN model was trained for. Fig. 2(b) plots the discriminator loss alongside the number of iterations, and Fig. 2(c) plots the generator adversarial loss with respect to the number of iterations.

C. SegFormer Semantic Segmentation Model

Semantic segmentation of all entities present in the CCGAN-enhanced images is executed by the deployment of the SegFormer-based semantic segmentation mechanism. Deploying the SegFormer framework on the GAN-enhanced post-disaster images resulted in the pixel-wise prediction of various entities present in an image. This mechanism generated a color coding, wherein each entity present in the image was associated with a unique color, thereby making it easy for the detection models to classify between various entities present in the image. Classes of importance, namely car and people, were annotated for semantic segmentation and used to train the SegFormer model. The necessary libraries were imported to be able to implement the SegFormer framework. The images obtained as output from the GAN module were transferred to the GAN folder to be able to make use of the occlusion-removed images for semantic segmentation. The SemanticSegmentationDataset class was defined to load and process the dataset for the semantic segmentation task. The SegFormer model was then defined and implemented to carry out the task of semantic segmentation on the images obtained from the previous module. The SegformerFinetuner class was defined to prune and finetune the parameters used for implementing the model. The model was then trained for 300 epochs. The Early Stopping mechanism was incorporated to reduce the chances of overfitting of the model. The model was trained by utilizing the GPU provided by Google Colab using the CUDA library. Tensorboard was instantiated to print the performance evaluation metrics that were witnessed during the training period. The model had minimal training loss, recorded at 0.0233. Fig. 3 depicts the performance evaluation of the SegFormer model trained for semantic segmentation. Fig. 3(a) displays the step-accuracy curve, and Fig. 3(b) portrays the Step-Loss curve.

D. Hybrid Ensemble Network

The images obtained as output was saved to the respective folders of YOLOv5 and Faster RCNN models, which were trained on enhanced images with entity separation for better performance. The YOLOv5 model was instantiated by downloading all dependencies from the official 'ultralytics' GitHub repository. Our dataset of choice, namely RescueNet, was annotated and augmented using the Roboflow software. Additional pre-processing techniques were incorporated within the Roboflow workspace, and a unique API key was generated for our dataset. This API key was then used to download the dataset and the respective annotations in PyTorch YOLOv5 format. The YOLOv5 model was then trained with a batch size of 16 for 492 epochs. The configurations previously defined in custom_yolov5s.yaml were also given as input parameters. In our case, early stopping was observed as the model's performance did not vary over a period of epochs, thereby stopping the training process in the 492nd epoch.

TABLE I: Performance Comparison of Standalone Models and Hybrid Ensemble

Model	Accuracy	Precision	Recall	F1 Score
YOLOv5	55.5	47.3	52.8	67.5
Faster RCNN	94.95	89.72	91.25	96.23
Proposed Hybrid Ensemble	**96.4**	92.5	93.79	97.95

TABLE II: Comparison of various Survivor-Detection Models

Detection Model	Accuracy
YOLOv3-MobileNetV1 pruned, fine-tuned [1]	61.98%
Global Local Feature Enhanced Network [5]	86.52%
Cascade RCNN + CenterNet (SyNet) [7]	52.10%
Soft-Weighted-Average Ensemble [8]	94.75
UGEN System	**96.40**

The Mean Average Precision (mAP) was calculated for every epoch/iteration and displayed in the output. Finally, the total number of instances that the model visualized under each class was tabulated and presented, and the weights were stored in the yolov5s_results folder as 'best.pt'. Tensorboard was then used to visualize the performance of the model. Tensorboard plotted all performance metrics observed while training the model over the range of epochs for which the model was trained. The overall training loss of the model was found to be 0.002 for identifying various classes. The accuracy of the model after 492 epochs was found to be 55.5%. The trained model was then tested by visualizing results obtained when test images were passed through the model. The bounding boxes constructed over detected classes were displayed.

The Faster RCNN multi-stage CNN model was implemented using Detectron2, a computer vision library that allowed the implementation of various CNN models through

the usage of PyTorch. Initially, the GPU instantiated in the Google Colab environment was displayed using the nvidia-smi command. The Faster RCNN model was then instantiated and trained using the faster_rcnn_X_101_32x8d_FPN_3x architecture present in Detectron2. Other parameters required for training the model were defined as well. The dataset previously annotated and augmented on Roboflow was downloaded along with annotations in the COCO JSON format. Roboflow, being a versatile software, allowed exporting various annotation formats for the same dataset. The COCO JSON format was used while training a Faster RCNN model. The dataset was then registered to Detectron2 to be able to use the dataset for training the Faster RCNN model. Once the model was trained, Tensorboard was initialized, thereby providing various performance metrics observed during the training process of the Faster RCNN model. The total loss of the model was observed to be 0.89. The accuracy of detecting various classes was found to be 94.92%. The trained model was then tested using the test set, and the results were visualized. The model and its corresponding weights were then saved and stored in the 'Saved_Models' folder.

The final ensemble model was created using the two trained models obtained in the previous steps. Both models were executed for the same image, and their respective outputs were visualized. The models were then combined using the ensembling technique named 'simple averaging', wherein multiple models were trained independently, and their predictions were combined to improve overall performance. The ensemble model was then evaluated using the same image used for the previous models, and the outputs were compared. Survivors were detected with an accuracy of 96.4%. Fig. 4 and Fig. 5 depict the performance evaluation of the standalone YOLOv5 and Faster RCNN models for survivor detection. The confidence-F1 score curve, confidence-recall curve, confidence-precision curve, and recall-precision curves for YOLOv5 have been plotted and displayed in Fig. 4(a), Fig. 4(b), Fig. 4(c), and Fig. 4(d) respectively. Fig. 5(a) depicts the Epoch-accuracy curve for each class identified by the faster RCNN model, and the Epoch-Loss curve is plotted in Fig. 5(b).

Table I depicts the comparison of the performance of standalone object detection mechanisms for survivor detection with the proposed hybrid ensemble network enhanced by a GAN-aided semantic segmentation module. It is evident that the performance of the hybrid ensemble network surpassed that of the standalone models. Table II portrays the accuracy of several existing survivor detection models and compares the same with the proposed UGEN system, from which it is evident the UGEN system's performance exceeds that of the existing detection models.

VII. CONCLUSION

Natural calamities lead to immense building damage and cause havoc among people who get trapped in the disaster. Since survivors may be present in a disaster-struck area, post-disaster survivor detection is essential to carry out effective Search and Rescue operations. Though UAVs are being widely used to scan the post-disaster area for survivors, inaccurate detection mechanisms lead to many survivors not being detected. Hence to mitigate the issues faced by current survivor detection frameworks, we propose a UAV-based post-disaster survivor detection mechanism that employs a GAN-aided ensemble network. The proposed mechanism aims to improve the accuracy of survivor detection in disaster-stricken areas. The GAN-aided ensemble network is trained using UAV imagery, enabling it to detect survivors in real-time. Furthermore, a novel GAN-aided semantic segmentation pre-processing module has been implemented to enhance the images fed as input to the detection model. The hybrid ensemble model comprising a single-stage YOLOv5 model and a multi-stage Faster RCNN model improves the accuracy of survivor detection, and reduces the inference time, thereby making the framework more suitable for real-time use. The proposed mechanism has shown promising results in terms of accuracy and speed compared to existing methods.In the future, the integration of data obtained from other sensors such as thermal imaging cameras, ground-based sensors, and acoustic sensors can be implemented to improve detection accuracy.

REFERENCES

[1] J. Dong, K. Ota and M. Dong, "UAV-Based Real-Time Survivor Detection System in Post-Disaster Search and Rescue Operations" in IEEE Journal on Miniaturization for Air and Space Systems, vol. 2, no. 4, pp. 209-219, 2021

[2] M. Rahnemoonfar, T. Chowdhury, A. Sarkar, D. Varshney, M. Yari and R. R. Murphy, "FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding," in IEEE Access, vol. 9, pp. 89644-89654, 2021

[3] M. Żarski, B. Wójcik, J. A. Miszczak, B. Blachowski and M. Ostrowski, "Computer Vision Based Inspection on Post-Earthquake With UAV Synthetic Dataset," in IEEE Access, vol. 10, pp. 108134-108144, 2022

[4] Isaac-Medina, Brian KS, Matt Poyser, Daniel Organisciak, Chris G. Willcocks, Toby P. Breckon, and Hubert PH Shum. "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark" In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1223-1232, 2021.

[5] T. Ye, W. Qin, Y. Li, S. Wang, J. Zhang and Z. Zhao, "Dense and Small Object Detection in UAV-Vision Based on a Global-Local Feature Enhanced Network," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-13, 2022

[6] A. Abdollahi, B. Pradhan, S. Gite and A. Alamri, "Building Footprint Extraction from High Resolution Aerial Images Using Generative Adversarial Network (GAN) Architecture," in IEEE Access, vol. 8, pp. 209517-209527, 2020

[7] Albaba, Berat Mert, and Sedat Ozer, "SyNet: An ensemble network for object detection in UAV images" in 25th IEEE International Conference on Pattern Recognition (ICPR), pp. 10227-10234, 2021

[8] H. Wang, Y. Yu, Y. Cai, X. Chen, L. Chen and Y. Li, "Soft-Weighted-Average Ensemble Vehicle Detection Method Based on Single-Stage and Two-Stage Deep Learning Models," in IEEE Transactions on Intelligent Vehicles, vol. 6, no. 1, pp. 100-109, 2021

[9] A. Bouguettaya, H. Zarzour, A. Kechida and A. M. Taberkit, "Vehicle Detection From UAV Imagery With Deep Learning: A Review" in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 11, pp. 6047-6067, Nov. 2022

[10] Rui, Xue, Yang Cao, Xin Yuan, Yu Kang, and Weiguo Song., "DisasterGAN: Generative Adversarial Networks for Remote Sensing Disaster Image Generation," in MDPI Remote Sensing 13, no. 21: 4284, 2021

[11] J. Dong, L. Zhang, H. Zhang and W. Liu, "Occlusion-Aware GAN for Face De-Occlusion in the Wild," in IEEE International Conference on Multimedia and Expo (ICME), London, UK, pp. 1-6, 2020

[12] Rahnemoonfar, Maryam, Tashnim Chowdhury, and Robin Murphy. "RescueNet: A High-Resolution Post Disaster UAV Dataset for Semantic Segmentation." UMBC Student Collection, 2021

[13] Li, Tianjiao, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. "Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles" In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16266-16275, 2021

[14] T. Chowdhury and M. Rahnemoonfar, "Attention Based Semantic Segmentation on UAV Dataset for Natural Disaster Damage Assessment" 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 2325-2328, 2021

[15] Lyu, Ye, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang., "UAVid: A semantic segmentation dataset for UAV imagery.," in ISPRS journal of photogrammetry and remote sensing 165, pp. 108-119, 2020

[16] T. Chowdhury, M. Rahnemoonfar, R. Murphy and O. Fernandes, "Comprehensive Semantic Segmentation on High Resolution UAV Imagery for Natural Disaster Damage Assessment," in IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 3904-3913, 2020

[17] M. Axelsson, M. Holmberg, S. Serra, H. Ovrén and M. Tulldahl, "Semantic labeling of lidar point clouds for UAV applications," in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, pp. 4309-4316, 2021

[18] H. Ren et al., "Swarm UAV SAR for 3-D Imaging: System Analysis and Sensing Matrix Design" in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-16, 2022

[19] T. C. Bybee and S. E. Budge, "Method for 3-D Scene Reconstruction Using Fused LiDAR and Imagery From a Texel Camera" in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 11, pp. 8879-8889, Nov. 2019

[20] M. Dai, T. H. Luan, Z. Su, N. Zhang, Q. Xu and R. Li, "Joint Channel Allocation and Data Delivery for UAV-Assisted Cooperative Transportation Communications in Post-Disaster Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 16676-16689, 2022