Real time situation reporting in battlefield using YOLOv4

Atif Rizwan

05 Oct 2021

Introduction

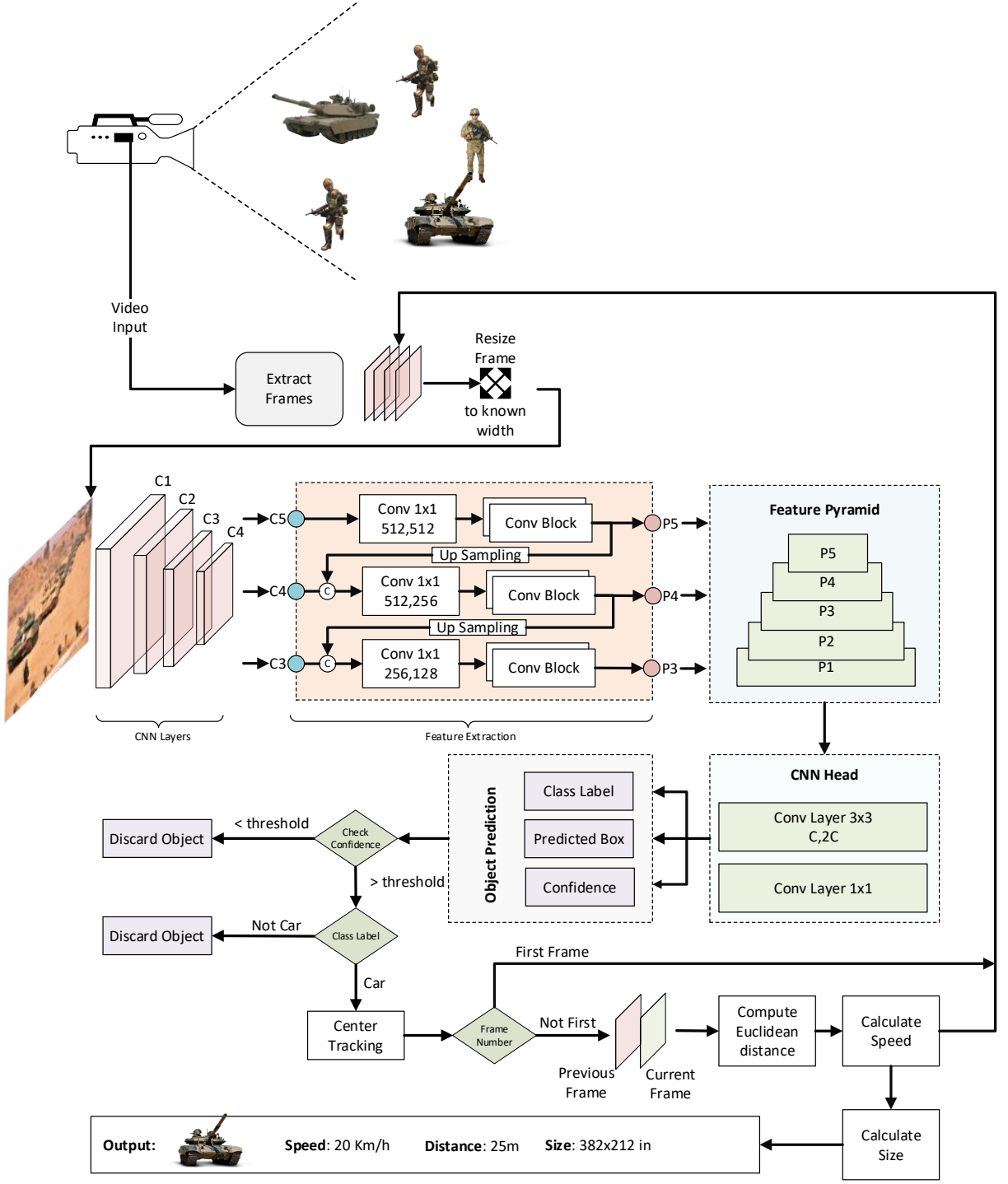
- Computer Vision (CV) is a technology that allows a computer to see. Multiple techniques have been used to process the images and extract patterns from the input frames
- Decision based systems are developed by using various CV techniques to report the situation in the real environment
- The study presents object understanding based on an object's size, distance, and speed
- Firstly the live video stream is passed to the Convolutional Neural Network (CNN) model based YOLOv4 framework
- The model is based on two steps
  - Firstly, Live video stream is passed to trained CNN model (YOLOv4) to detect the type of the object
  - Secondly, the size, distance and speed of the object is computed by using reference object
- After computing the required information, the situation in the video is reported like "The Tank of size 572 x 220cm in the distance of 125m is moving at the speed of 45 km/h"

CNN to Detect Object type and Darknet

- Real time video from a camera (CCTV or any other) is passed to CNN model to detect the type of an image
- The video stream is passed to a process that convert the stream into frames. The frame size is then changed and a known width is assigned to calculate pixel per meter in next stages.
- After that frames are passed to Convolutional model of CNN which is based on 4 block conv neural network
- The main objective of the backbone is to extract the essential features, the selection of the backbone is a key step it will improve the performance of object detection. Often pre-trained neural networks are used to train the backbone.
- After the augmentation process in the backbone section features are extracted from the frames to detect the object type and classify the detected objects.
- YOLO is a network that uses Deep Learning (DL) algorithms for object detection.
- YOLO performs object detection by classifying certain objects within the image and determining where they are located on it.
- Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.
- Darknet is installed with only two optional dependencies: OpenCV if users want a wider variety of supported image types or CUDA if they want GPU computation.
- Neither is compulsory but users can start by just installing the base system which has only been tested on Linux and Mac computers.

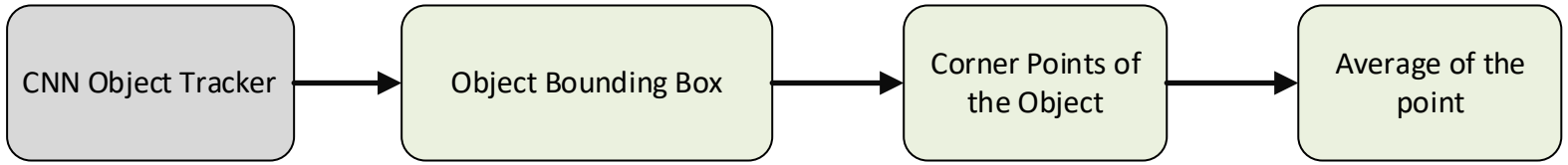
Architecture

- The detection of object starts from data acquisition from live camera
- After the video capturing, frames from video are extracted and passed to the object detection model which is based on convolutional layers
- These convolutional layers extract features from image for detection of objects
- The feature pyramid is the output of DNN and these features are used to predict the label of object, the bounding box and the confidence level of prediction
- The final output is based on 3 attributes label, prediction box and confidence level of prediction
- The output of feature extraction phase is a feature pyramid which is passed to head of CNN which classify the data
- The classification of the data is based on two convolutional layer, which down sample the image and reduce the size of the matrix.
- Finally the output of the CNN is the tuple which contain (class label, Confidence of prediction and the bounding box of the detected object)
- To perform the further processing on the predicted objects, some instances are discarded based on the confidence of the prediction
- To detect the speed of the specific object (car) all other objects are discarded by filtering the predicted class label.
- Next, the center, distance, speed and size of the object is calculated and discussed in details in next slides



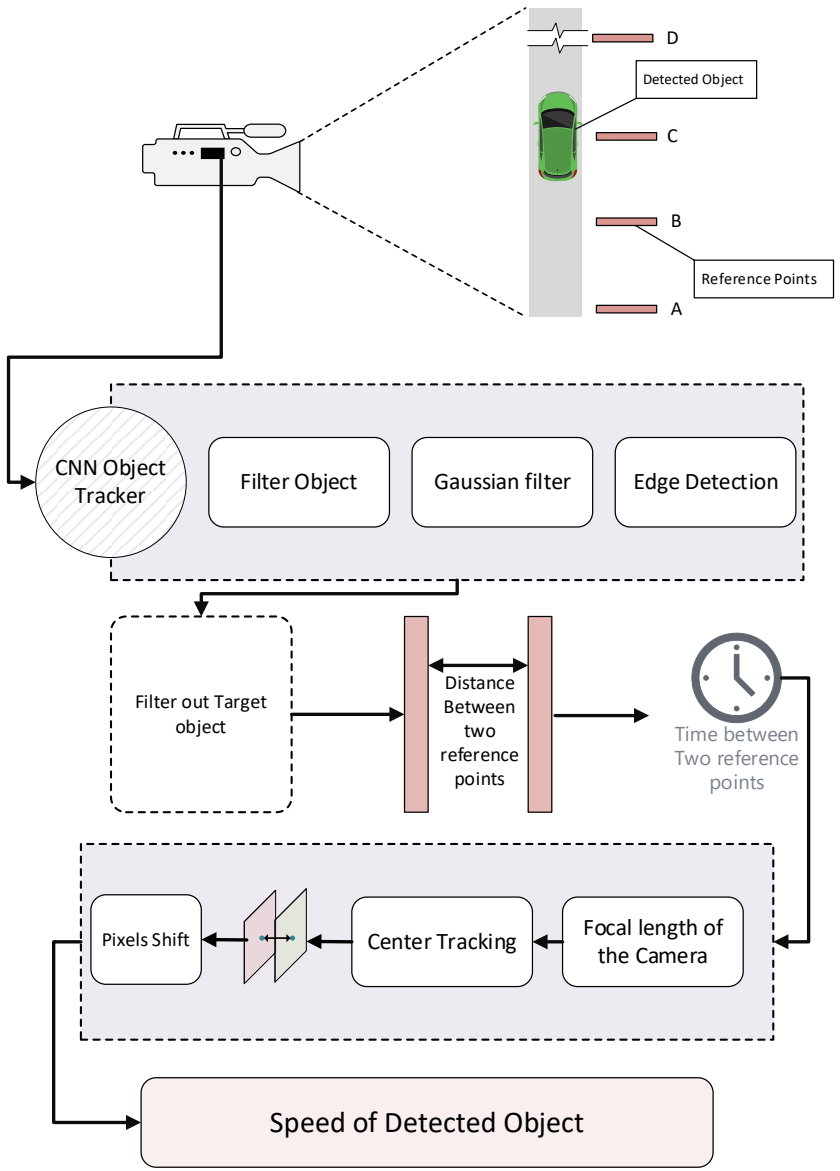
Object Center detection

- When the video stream passed from CNN model, in result we get bounding boxes of the detections.
- Furthermore, the detections are preprocessed based on the class label and the level of confidence.
- The output of CNN contain bounding box for each detected object.
- The average of the box give us the center point of the object
- The purpose of tracking center point is to compute the size and speed of the object.
- Similarly, the distance of two frames will be computed by using the center point of the image.



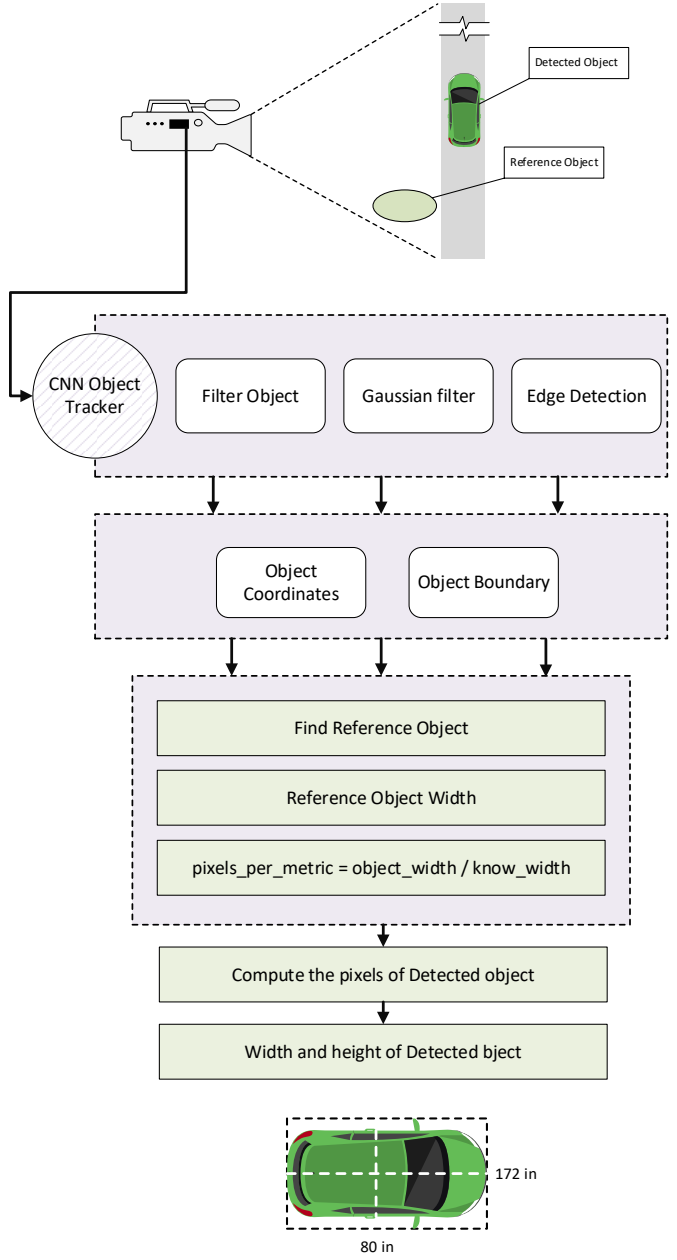
Object Speed Detection

- To detect the speed of an object in a live video stream we need to first separate the targeted object from all others.
- After that, we also need to place reference objects to detect the speed of an object
- First image smoothing is applied to detect the edges of an object
- Targeted object is filtered by using the class label assigned by CNN model
- We should know the distance between two reference points in advance and the time will be calculated during process
- Next, focal length of camera and the center of an image is calculated.
- The different between two consecutive frames gives us the total shifted pixels of an object
- By considering the meter per pixel (calculated initially) we can calculate the speed of the object



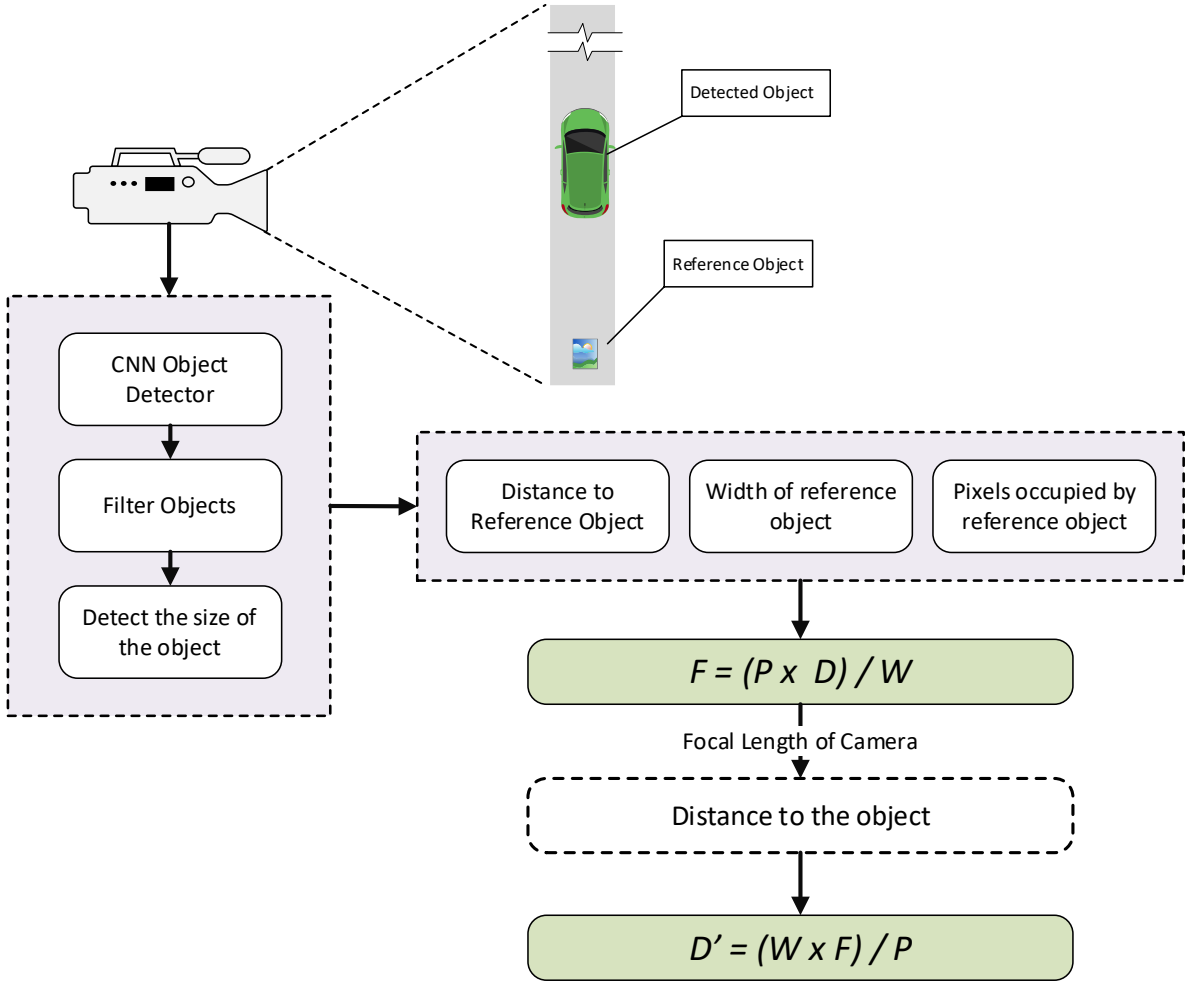
Object Size detection

- The size of the object is also an important feature that can be detected
- As we receive the data from the live video stream and all the objects are classified in different classes
- CNN model return the bounding box of all the detected objects, so the targeted object is extracted from all others.
- We need to place a reference object (known width and distance from camera)
- First the pixels occupied by the reference point are computed and based on known width of object, pixel per metric value is computed.
- Finally, the pixels of targeted object are computed and the width of height of the object are computed.



Distance from Object to camera

- To compute the distance of an unknown object from the camera, we need to utilize the triangle similarity
- Triangle similarity can be defined like this: For Instance If we have a object or marker with known width W, then we place a this marker to some distance D from the camera
- After that, we will take a picture of an object by using a camera.
- Objects in the picture will be detected by CNN and compute the focal length of the camera.
- The focal length of camera depends on P (pixels taken by reference object), D (Distance to reference object) and W (Known width of reference object.



Contd.. Distance from the object

- The focal length of the camera can be computed by using the equation below
  - $F = (P \times D) / W$
- Now, we can apply triangle similarity by moving the camera closer and farther from the object. This allows us to compute the distance of the object to the camera.
  - $D' = (W \times F) / P$

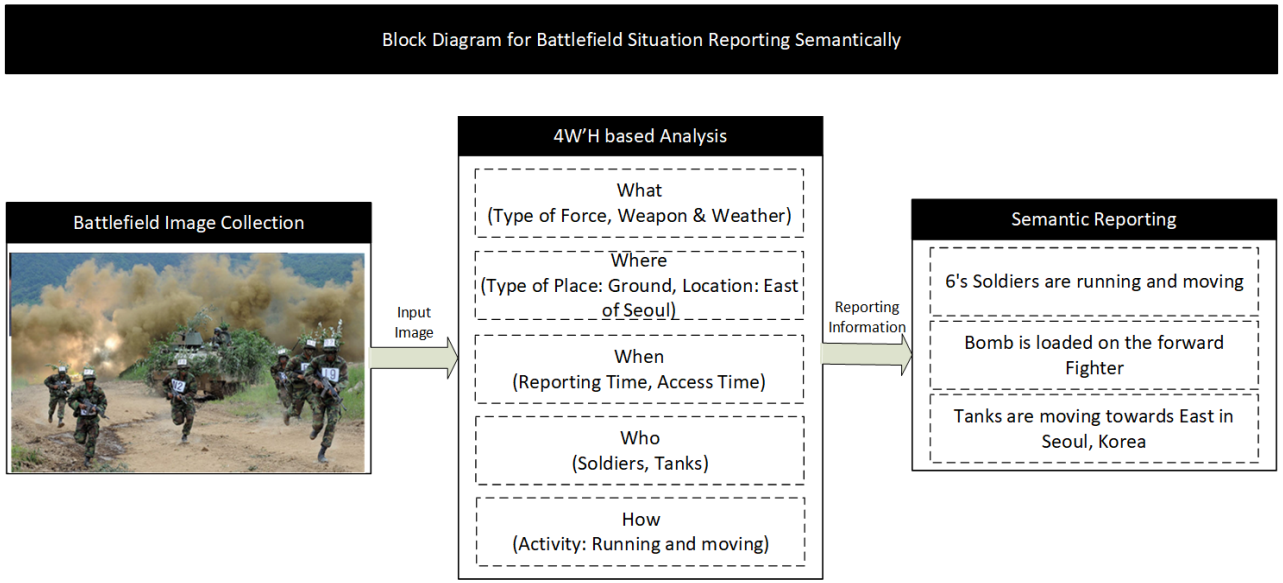
For instance, If we move camera 3 *ft* away from the reference object and take another photo or take frame from the video. Now, by using OpenCV we are able to determine the perceived width of reference object, which is 170 *pixels.* By putting the values in above equation we get

$$D' = (11in \times 543.45) / 170 = 35in$$

Or roughly 36 inches, which is 3 feet.

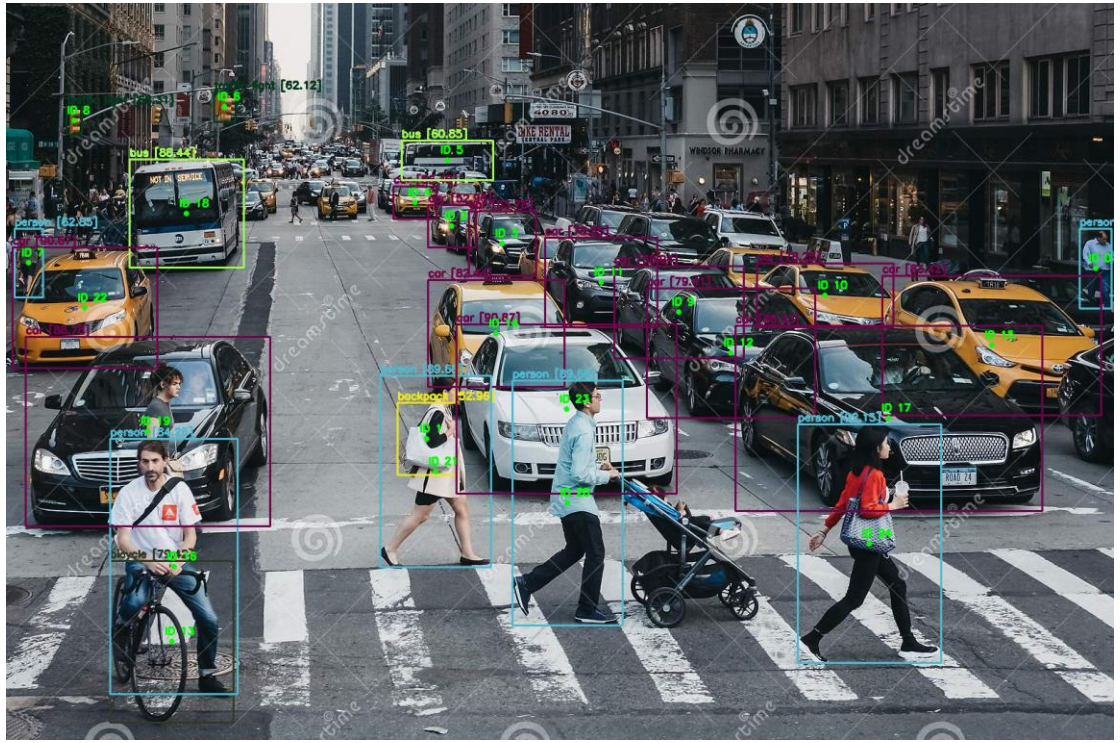
Situation Reporting

- The following figure presents proposed block diagram for battlefield situation reporting semantically.
- The input layer consists of battlefield image in order to extract context source.
- The 4W'H architecture consist of What (Type of Force, Type of Weapon), Where (Place and Location coordinates), When (Reporting Time), Who (Soldiers and Tank), and How (Activities monitoring).
- The following semantic situation information are reported:
  - 6's Soldiers are running and moving
  - Tanks are moving towards East of Seoul, Korea
  - Bomb is loaded on the forward Fighter



Object Detection results

- The framework of CNN is used to detect the class of the object and return the class label along with confidence value
- Latest darknet version is able to detect 90 classes of different objects
- Object with the bounding box is then forwarded to the object center detection module to detect the center point which will be used to calculate the speed of moving object
- The figure shows the output of CNN, which detect the type of an object

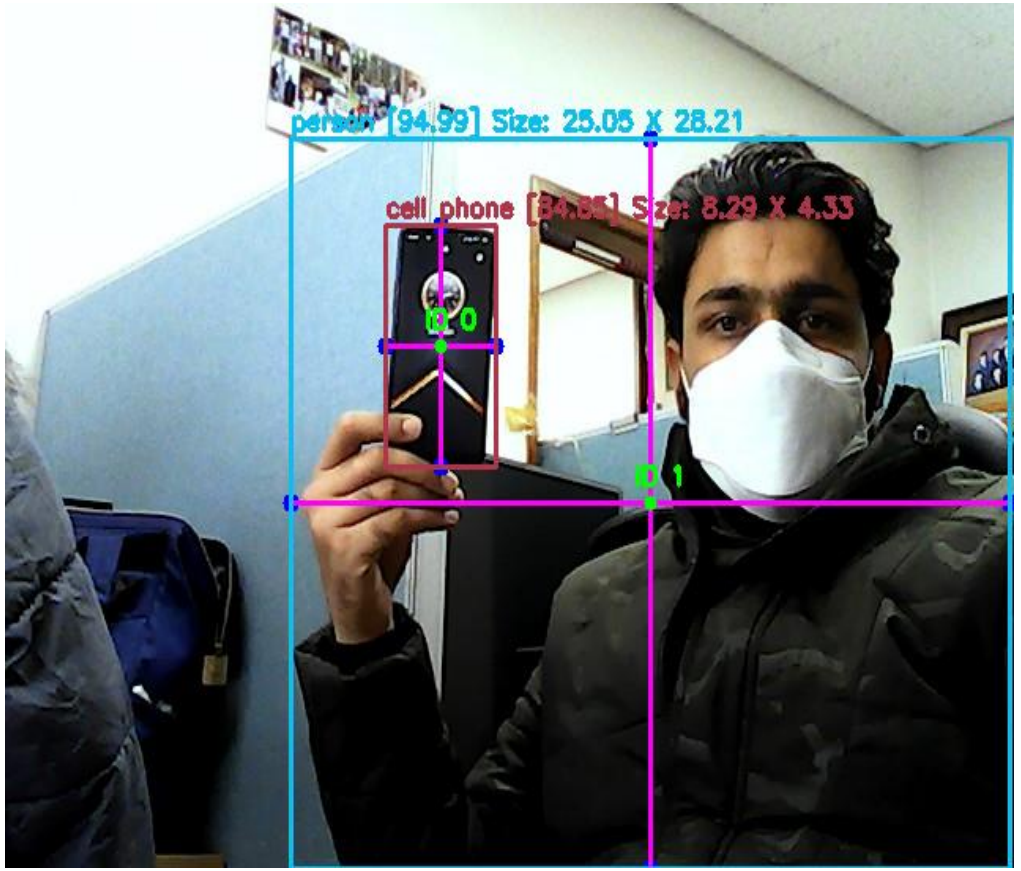


Center point

- The center point is detected by getting the average of bounding box points.
- The bounding box for each object is returned by CNN model
- The center point is the average of two diagonal points of the box
- Four points of the box (left, right, bottom, top) are combined as (left, top) and (right, bottom)
- Left top and bottom right are averaged and we get the center point of an image
- A unique ID to each center point is assigned as shown in figure

Results (Size detection)

- Size detection of an object is based on the pixel per metric value
- Pixel per metric is calculated by using the initial and actual width of reference object.
- Pixel per metric shows that how many pixels are covered in one metric (cm).
- Width and height of all the detected objects are calculated by using reference object
- The coordinates of the frame are averaged by using the same method which was used in center point detection.

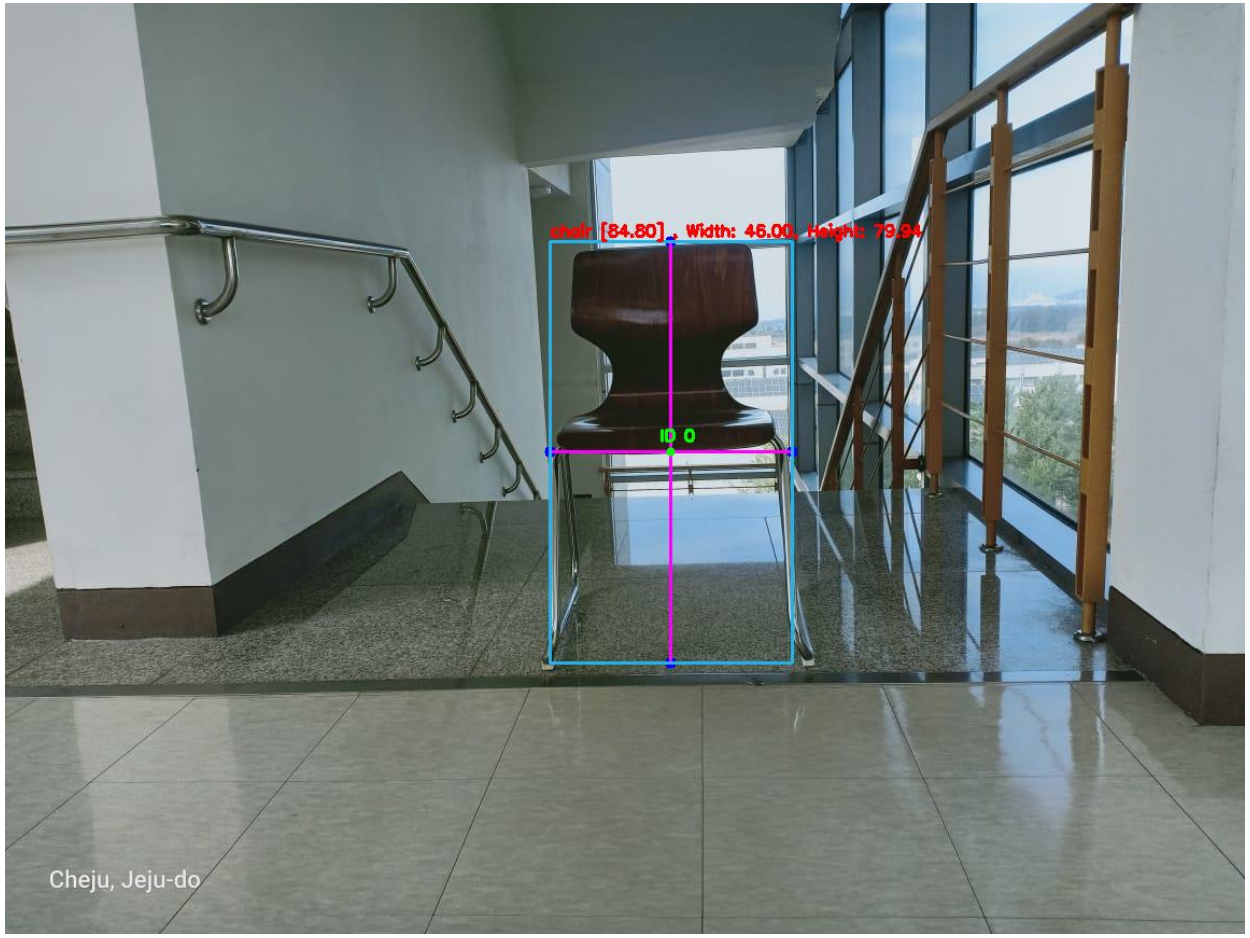


Demo Video

Distance and Size
of moving object

Size Computation

- To compute the size of object we need to know about the distance and the width of reference object
- The picture of chair is captured from the distance of 157cm
- Next the width and height of same object is computed manually
- The picture is passed to the model, first CNN predict the object's bounding box
- Next the width and heigh of object is computed as shown in figure



Speed of the moving Object

Demo Video

Speed of moving object

Thank You