Object Detection in Artisanal Small-Scale Gold Mining Environments using a Modified YOLOv4 Algorithm

Akpah Sylvester
Computer Science and Engineering Department
University of Mines and Technology

Alese Boniface Kayode
Department of Cyber Security
The Federal University of Technology

Yao Yevenyo Ziggah
Geomatics Engineering Department
University of Mines and Technology

ABSTRACT

Detecting objects in high-resolution images could be a very challenging task, particularly, when analysing remote sensing imagery captured with an Unmanned Aerial Vehicle (UAV) at Artisanal Small-Scale Gold Mining (ASGM) environments. Due to the heterogeneous nature of ASGM environments in Ghana, object detection algorithms are prone to misclassification errors in identifying irrelevant ground objects for target objects. In recent times, research into Convolutional Neural Networks (CNNs) for object detection has gained immense popularity which can be attributed to its proven dominance to efficiently learn and extract image features. This study proposes a modified You Only Look Once (YOLO) algorithm known as ASGM-YOLO which is based on the YOLOv4 framework to detect objects of interest such as excavators, sluice boards, tailings dump, crushers, persons, and trucks from UAV captured images at ASGM sites. The goal is to monitor illegal ASGM activities by detecting these objects quickly so that further damage to the environment can be stopped. The ASGM-YOLO algorithm is a single-stage object detector that adopts an end-to-end detection approach to predict class probabilities and bounding boxes around objects faster with optimal accuracy. The detection accuracy of the proposed ASGM-YOLO algorithm was compared to other algorithms and the results showed that the ASGM-YOLO performed better by achieving a detection accuracy of 96.50%.

General Terms

Object Detection, Convolutional Neural Networks,Contrast-Limited Adaptive Histogram Equalization, LabelImg

Keywords

Artisanal Small-Scale Gold Mining, YOLOv4 Algorithm, Unmanned Aerial Vehicle, Object Detection, Galamsey

1. INTRODUCTION

Monitoring the activities of illegal ASGM popularly known as “Galamsey” in Ghana is a very difficult task. This is largely due to the remote and inaccessible areas in which these illegal activities are carried out. The first logical step to overcome the problem of inaccessibility and thus be able to control and prevent illegal ASGM is identifying the locations of these illegal activities [1]. So far, the use of 4WD vehicles and/or trekking are the only means of assessing the sites of illegal ASGM. Unfortunately, this way of assessing the sites have yielded no meaningful results because these illegal activities are carried out in remote and inaccessible areas. A promising area to explore to address the problem of inaccessibility is to employ the use of Unmanned Aerial Vehicles (UAVs) for monitoring purposes [2], specifically in the monitoring of ASGM. In recent times, UAVs also known as drones have become popular due to their widespread application in several areas notably, surveillance [3], urban mapping and analysis [4], disaster management [5], precision agriculture [6], inventory management [7], infrastructural inspection [8]and object detection [9]. Remotely controlled, UAVs are inexpensive aircraft that are equipped with point-and-shoot cameras which allows them to reach inaccessible areas from a safe distance to collect aerial images [10]. Even though the use of UAVs is replete with several advantages, there are some challenges which mainly can be attributed to the manual processing of captured aerial images, which usually require a lot of man-hours and, in some instances, become excessively expensive when processing large data files. In the works of [11] and [12], the authors employed human experts to manually process and detect objects of interest in aerial images. The results showed that their approach worked very well on small datasets; however, a lot of time was wasted when analysing large datasets. The difficulty in analysing large datasets can be alleviated by analysing only the images that are likely to contain any object of interest. Regrettably, analysing just the images that contain some objects of interest is likely to introduce many false-negative predictions into the dataset. To minimise the challenges posed by human intervention, this paper sought to leverage the successes of deep learning techniques to efficiently detect objects of interest in UAV-captured images at ASGM sites.

2. RELATED WORK

Deep learning is a subset of machine learning that focuses primarily on teaching computers to learn deep feature representation of objects hierarchically [13] [14]. Learning the object features are carried out through representation learning. In the work of [15], the authors opined that deep learning techniques have significantly enhanced numerous state-of-the-art tasks in computer vision notably; object detection and classification [16] and image processing [17]. In [18] [19], the researchers concurred that Convolutional Neural Networks (CNNs) have become the leading deep learning architecture [20], particularly, for general object detection and classification [21] [22]. Several research works have employed different techniques to map out the extent of environmental degradation caused by illegal ASGM. For example,[23] employed UAVs and Google Earth scenes to assess the extent of environmental degradation in the Apamprama Forest Reserve of Ghana due to illegal mining activities. [24] [25] [26] proposed the use of a multi-temporal optical remote sensing dataset from LANDSAT to map the expansion of illegal mining sites in Wa in the Upper West region of Ghana. [27] adopted field surveys, open data kit (ODK), ArcGIS, and Google Earth images to map and visualise the spatial distribution patterns of ASGM in the Western region of Ghana. [28] investigated the use of annual time-series Sentinel-1 data to map and monitor illegal mining activities along major rivers in South-Western Ghana between 2015 and 2019. [29] assessed the potential of Sentinel-2 data to identify mining areas to detect, map, and understand the dynamics in the land cover change in the Municipalities of El Bagre and Zaragoza in Bajo Cauca, Colombia. [30] adopted three CNN architectures namely, SharpMask, U-Net, and ResUnet to classify Landsat data to monitor change detection of deforestation in the Brazilian Amazon between 2018 and 2019. [31] explored a combination of free cloud computing, free open-source software, satellite images, and CNN to analyse a real, large-scale problem relating to the automatic country-wide identification and classification of landforms and mine tailings dams in Brazil. [32] explored the use of high-resolution satellite imagery and a single shot multibox detector for the detection of tailings dams in the Jing–Jin–Ji Region in China. [33] explored the use of deep learning algorithms to map changes in the ASGM landscape based on the 2017 ban on ASGM in Ghana.

Even though the work of [31] [32] [33] used CNN techniques, their approaches presented weaknesses in detecting small and densely occluded objects from aerial images at ASGM sites. This study seeks to explore the capability of using deep learning for object detection at ASGM sites. The contributions made in this paper are summarised as follows:

(i) The study leveraged the performance and robustness of the YOLOv4 (Bochkovskiy*et al.*, 2020) deep learning technique based on the CSPDackNet 53 backbone architecture to propose an ASGM-YOLO algorithm to detect objects of interest from UAV images captured at ASGM sites;

(ii) With the introduction of the Spatial Pyramid Pooling (SPP-1) and Spatial Pyramid Pooling (SPP-2) in front of the CSPDackNet 53 backbone and the tail of the ASGM-YOLO algorithm, more feature-rich image information was extracted and great enhancements were made in the performance of the proposed ASGM-YOLO algorithm compared to other-state-of-the-art techniques; and

(iii) With improvements made to the localisation and classification loss functions, the ASGM-YOLO algorithm achieved excellent performance on making bounding box predictions for objects in UAV-captured images at ASGM sites.

The rest of the paper is structured as follows: Section 2 presents the resources and methods employed to achieve the objectives of the study. The section further presents the frameworks of the proposed deep learning algorithms. Section 3 shows the experimental results with the discussion and interpretation, and Section 4 concludes the study and presents recommendations for future works.

3. METHODOLOGY

This section focused explicitly on the research methodology employed to achieve the objectives of this paper.

3.1 Resources Used

A Garmin hand-held Global Positioning System (GPS) was used to collect ground coordinates of all ASGM sites. A DJI Phantom 4 UAV equipped with a high-resolution digital camera of pixel resolution of 4000 × 2250 and a Field of View (FOV) of 70^{o} was used to capture the aerial images at the ASGM sites. A Dell G5 15 laptop with Intel Core i7 (3.1 GHz base frequency, 6 cores) was used to train the ASGM-YOLO algorithm. The system is a CUDA-capable device and runs on Windows 10 Professional operating system. It features an NVIDIA GeForce RTX 1060 (10GB GDDR5 Dedicated) and 64GB RAM with a 128-bit interface. Python programming language was used and anaconda jupyter notebook was the model development environment. Keras deep learning library running on TensorFlow v1.12.0 backend was used for experimentations.

3.1.1 Study Area

Tarkwa-Nsuaem Municipality, located in the Western Region of Ghana was chosen as the study area for this research. The Municipality is located between latitudes 4° 54' 30" N, and 5° 22' 20" N and longitudes 2° 10' 50" W and 1° 45' 30" W, respectively. Tarkwa-Nsuaem Municipality was selected for this study because it represented one of the most active gold mining districts in Ghana, thereby presenting a vast concentration of ASGM activities. Apart from Large-Scale Gold Mining (LSGM), registered and controlled ASGM also abounds in the Municipality. Illegal ASGM is ubiquitous in areas that are replete with registered LSGM and ASGM operations. All these influenced the choice of the study area which is shown in Fig. 1.

3.2 Methods

The workflow chart for the proposed ASGM-YOLO algorithm is shown in Fig. 2. The research was implemented in four phases namely; data acquisition, dataset construction, model training, and object detection. The dataset construction phase included the use of UAV for data acquisition, pre-processing techniques applied, data annotation and data augmentation. The model training phase included setting model hyperparameters, model initialization, and setting anchor box parameters. ASGM object detection was the final phase. It included loading the testing dataset, loading the model, and predicting the presence of objects in the testing images.

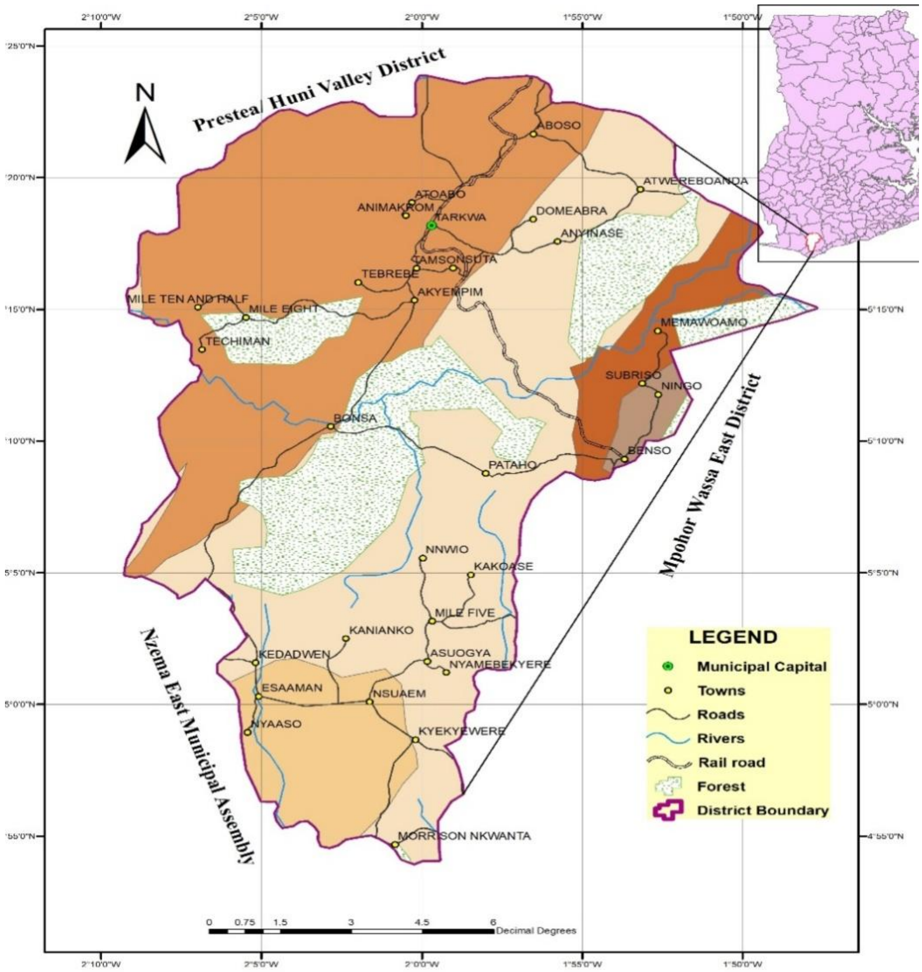


Fig. 1 Study Area

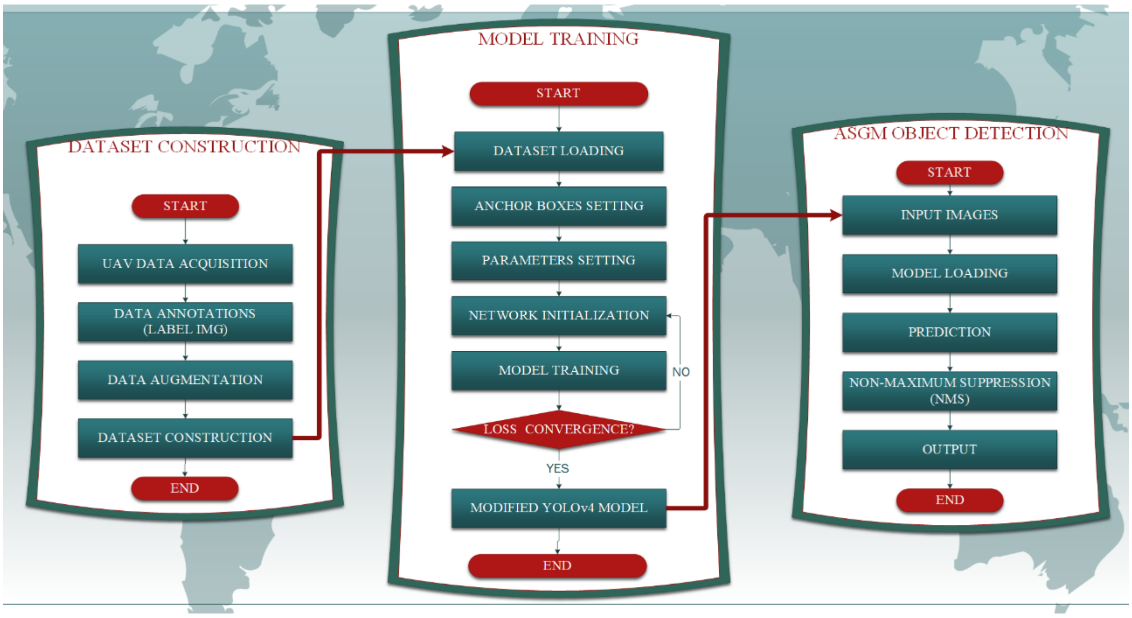


Fig. 2 Workflow Chart of the ASGM-YOLO Algorithm

3.2.1 Data Acquisition

A DJI Phantom 4 UAV as shown in Fig. 3 and equipped with a high-resolution digital camera of resolution 4000 × 2250 pixels and a Field of View (FOV) of 94^{o} was used to collect aerial photos of the various ASGM sites. Different ASGM sites were visited between 12th and 18th October 2020 between the hours of 11:00 am to 2:00 pm each day when the weather was fine. Six separate flight plans were performed at three different altitudes (i.e., 100 m, 150 m, and 200 m). Table 1 shows the various flight plans carried out to capture the aerial images and Fig. 4 shows some of the sample images captured with the UAV.

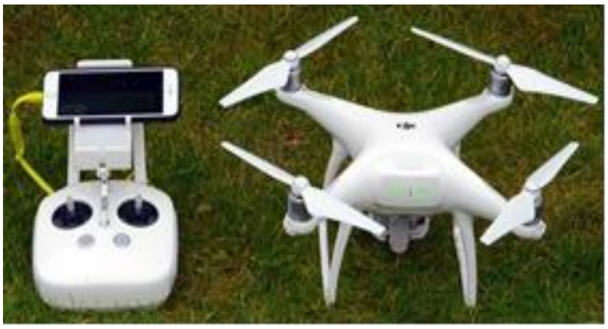
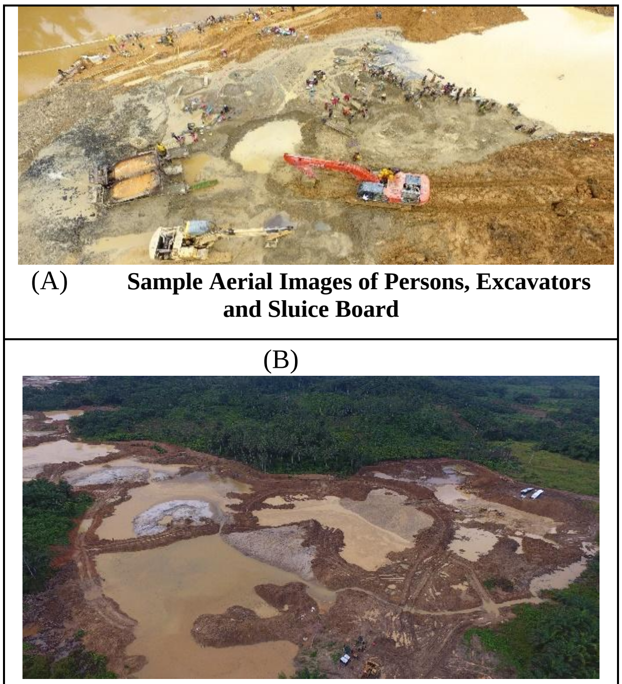


Fig. 3 DJI Phantom 4 UAV used for Data Collection

Table 1 Flight Details for Multiple Flight Altitudes

SN	Parameters	100 m	150 m	200 m
1	Speed (m/s)	2	2	2
2	Shooting Angle	67^{o}	67^{o}	67^{o}
3	Front Lap	75%	75%	75%
4	Side Lap	75%	75%	75%
5	Resolution (cm/px)	2.03	2.03	2.03
6	No. of Flights	6	6	6
7	Start Time of Flight	11:05 am	13:00 pm	10:30 am
8	Total Flight Duration	2hrs 10 mins	2hrs 05 mins	1hr 45 mins
9	Number of Images Collected	1 713	1 604	1 543



(A) **Sample Aerial Images of Persons, Excavators and Sluice Board**

(B)

((B) Sample Aerial Image of Tailings Dump and Physical Structures

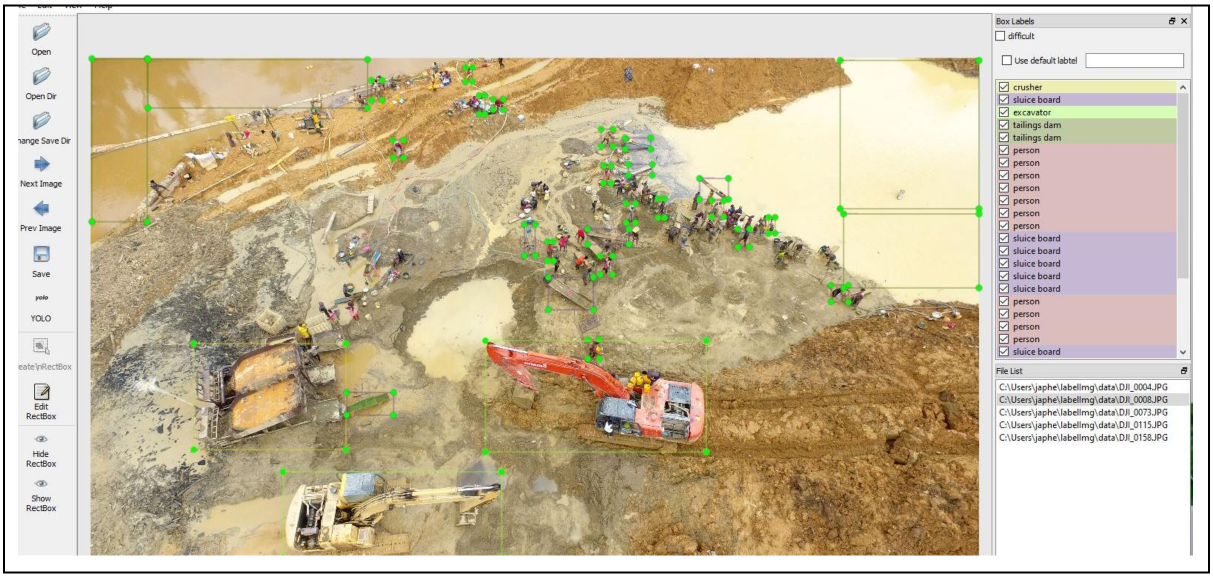
Fig. 4 Sample Aerial Images Captured with the UAV

3.2.2 Image Pre-processing

Image preprocessing was undertaken as the first step in the object detection process. It enhanced the contrast of images. In this paper, Contrast-Limited Adaptive Histogram Equalization (CLAHE) [34] was employed as the image pre-processing technique and was used to improve the contrast of the images. This was carried out by computing several histograms that make up the various sections of the image and reallocating the luminance values across the entire image. The original histogram was clipped and reallocated at each gray-scale level on each image. At this stage, the assigned histogram was different from the original histogram, because each image pixel intensity was limited to a defined maximum value. However, the optimised image, as well as the original image, had the same minimum and maximum gray values.CLAHE was adopted because it presents two main sub-sections namely: Block Size (BS) and Clip Limit (CL). In this case, the BS and CL were used to control the quality of the images produced during the optimization process. When the illumination rate of an image is too high, it meant that the CL values had increased because input images generally have low-intensity values and a larger CL flattened the histogram values. During the pre-processing phase, each image frame was resized to a defined pixel size of 416 x 416 before being fed into the network.

3.2.3 Data Annotation

The aerial images were annotated using the LabelImg software [35]. LabelImg is a graphical image annotation tool written in Python and easy to use because of its Graphical User Interface (GUI). The annotations are saved as XML files in the Pascal VOC and ImageNet formats. Each image frame was manually labelled by drawing rectangular bounding boxes perfectly around the desired objects of interest in the images and their respective object classes were also selected, as shown in Fig. 5.



Box Labels

☐ difficult

☐ Use default labtel

☑ crusher
☑ sluice board
☑ excavator
☑ tailings dam
☑ tailings dam
☑ person
☑ person
☑ person
☑ person
☑ person
☑ person
☑ person
☑ sluice board
☑ sluice board
☑ sluice board
☑ sluice board
☑ sluice board
☑ person
☑ person
☑ person
☑ person
☑ sluice board

File List

```
C:\Users\japhe\labelImg\data\DJI_0004.JPG
C:\Users\japhe\labelImg\data\DJI_0008.JPG
C:\Users\japhe\labelImg\data\DJI_0073.JPG
C:\Users\japhe\labelImg\data\DJI_0115.JPG
C:\Users\japhe\labelImg\data\DJI_0158.JPG
```

Fig. 5 Sample Annotated Aerial Images of ASGM Sites

3.2.4 Data Augmentation

The ultimate goal of data augmentation is to increase the number of images in the dataset as well as to introduce more variability into the dataset thus helping the model to generalise very well. Finally, issues about dataset class imbalance and problems associated with model overfitting were handled with the introduction of several data augmentation techniques. To achieve this, a multiple-way data augmentation (MDA) technique proposed by [36] was adopted and optimised. [36] in their approach proposed a 14 MDA which employed seven different data augmentation techniques on two horizontal plains namely; noise injection, horizontal shear, vertical shear, rotation, Gamma correction, scaling, and translation on raw data images. To optimise the technique proposed in [36], this paper proposed a 16 MDA. Compared to the 14 MDA, CutOut and CutMix augmentation techniques were added to the proposed 14 MDA to make up a 16 MDA technique. It was noticed that the introduction of these two techniques helped to increase the performance of the model. The proposed data augmentation technique is expressed as:

$$n(i) \to C(i) = D\left\{x = 1^D \dots 8\frac{f_{x^{DA}}[n(i)]}{w}x = 1^D \dots 8\frac{f_x^{DA}\{M[n(i)]\}}{w}\right\}(1)$$

where W = 40. Hence, $D_{(i)} = 16W + 1215 = 19440$ images was generated from the original image *n(i).*

The various phases of the augmentation process are described as follows.

Phase 1 - Eight photometric data augmentation techniques were applied on each raw image data *n(i).*$f_x^{DA}, x = 1, \dots, 8$ was used to represent each data augmentation process. It is important to note that for each

f_x^{DA} the data augmentation process produced several W new images. Therefore, for each data image *n(i)*, an enhanced dataset $x = 1^N \dots . 8^{f_x^{DA}[n(i)]}$ was produced, where N represented a concatenation function.

Phase 2 – Avertical reflected data image was produced $M[n(i)]$, where M denotes the vertical mirror function.

Phase 3 – A combination of the raw image data *n(i)*, the mirrored data image $M[n(i)]$, results from the 8-way data augmentation process $x = 1^N \dots . 8^{f_x^{DA}[n(i)]}$, and the results from the mirrored 8-way data augmentation process $x = 1^N \dots . 8^{f_x^{DA}[n(i)]}$are concatenated. Representing these results mathematically proved that, one training image *n(i)* produced a dataset D*(i)* which contained 16W + 1 new data image. A total of 19 440 images was obtained after going through the augmentation process. Algorithm 1 shows the improved data augmentation technique. A total of 19 440 images were obtained after the augmentation process. The dataset was divided into 15 552 (80%) for the training set and 3 888 (20%) for the testing set. technique.

Algorithm 1 16-Way MDA Algorithm

Input: Raw image data *n(i)*

Process:

Phase 1 - Eight different photometric data augmentation techniques were applied to the raw data image *n(i)*

Phase 2 - A vertical mirror image data of the data augmentation techniques was generated.

Phase 3 - The raw image n(i), the mirrored image, the 8-way data augmentation techniques result of the raw image, and the 8-way data augmentation results of the vertical mirrored data images were combined to form a new dataset *D(i).*

Output: The new dataset of the original data image *D(i)* with the augmented data images was created.

3.2.5 Dataset Construction

After augmenting the data, the constructed dataset contained static aerial images captured at the different ASGM sites at different orientations with the DJI Phantom 4 UAV. The ground truth data helped in retrieving the point coordinates of the objects of interest captured in the images. The dataset contained a total of 19 440 static images.

3.2.6 Dataset Class Distribution

For the ASGM-YOLO algorithm to efficiently learn high feature representations of the objects of interest, the images in the dataset were categorized into six main classes; person, truck, excavator, tailings dump, sluice board, and crusher. The class distribution graph of the dataset is shown in Fig. 6.

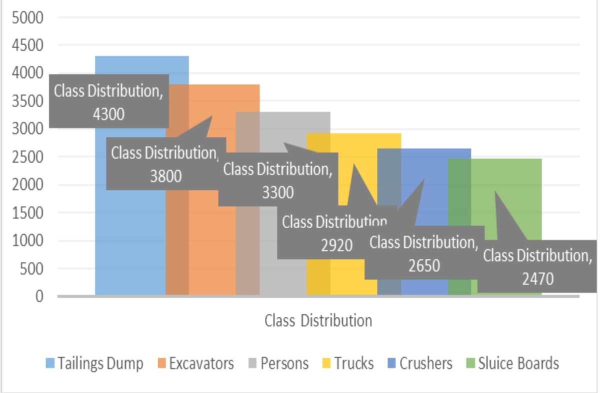


Fig. 6 Dataset Class Distribution

3.3 Object Detection

Object detection is a computer vision technique that involves identifying the presence of an object, the class the object belongs to, and the location-specific coordinates of the objects in a given image file. Object detection has shown to be a very difficult task that encompasses the development of techniques for object recognition (i.e. where is the object), object localisation (i.e. To what extent is the objects), and object classification (e.g. what are they). In the last decade, deep learning techniques continue to achieve excellent results for object detection on standard benchmark datasets. In recent times, "You Only Look Once (YOLO)" [37] family of Convolutional Neural Networks has proved dominant with its near state-of-the-art results with a single end-to-end approach that can perform object detection in real-time. The YOLO algorithm has evolved over the years from YOLOv1 [37] predicted bounding box coordinates and the presence of objects, and class scores where necessary. YOLOv2 [38] employed the k-means clustering technique to cluster the bounding boxes in the training set before predictions are made. YOLOv3 [39] introduced a residual skip connection to manage the vanishing gradient problems in deep neural networks. YOLOv4 [40] combined Weighted Residual Connections (WRC), Cross-Stage Partial Connections (CSP), and cross small-batch connections (Cross mini-Batch Normalization (CmBN)), self-adversarial training (SAT), and Mish activation function which makes use of the head part of the YOLOv3 algorithm. YOLOv4 changes the backbone network to CSPDarknet53, and employs Spatial Pyramid Pooling (SPP) [41] to enlarge the receptive field, with PANet [42] as part of the neck structure. YOLOv4 algorithm was adopted and optimised in this paper because, at the time of writing, the authors of YOLOv4 had published a journal paper establishing the results of their findings.

3.3.1 ASGM-YOLO Algorithm

Building a robust and efficient ASGM-YOLO algorithm which is based on the original YOLOv4 algorithm required a thorough investigation into the right YOLO family of CNNs to make a choice. As shown in Fig. 7, the proposed ASGM-YOLO algorithm is a 106 fully connected CNN that is divided into five sub-sections: input layer, CSPDarknet-53 backbone, Spatial Pyramid Pooling (SPP-1) and (SPP-2) layers, Path Aggregation Network (PANet) which forms part of the Neck, and the head network which doubles as the output. The ASGM-YOLO algorithm received as input; images resized to a defined pixel size of 416 × 416 × 4. The resizing was important as it helped to maintain the aspect ratio of the original input images.To accurately predict and detect target objects of different sizes, four different scales of Bounding Boxes (BBs) would be predicted, expressed as $Y1$, $Y2$, $Y3$, and $Y4$ (Fig. 7). The four BBs would be predicted at each scale, which means a tensor of $M \times M \times [4(4+1+1)]$ for the 4 BB coordinates, the first 1 represents the confidence score, whilst the second 1 represent the probable class prediction. M denotes the feature map of sizes $Y1$, $Y2$, $Y3$, and $Y4$; which are represented by 19, 38, 76, and 95 respectively.

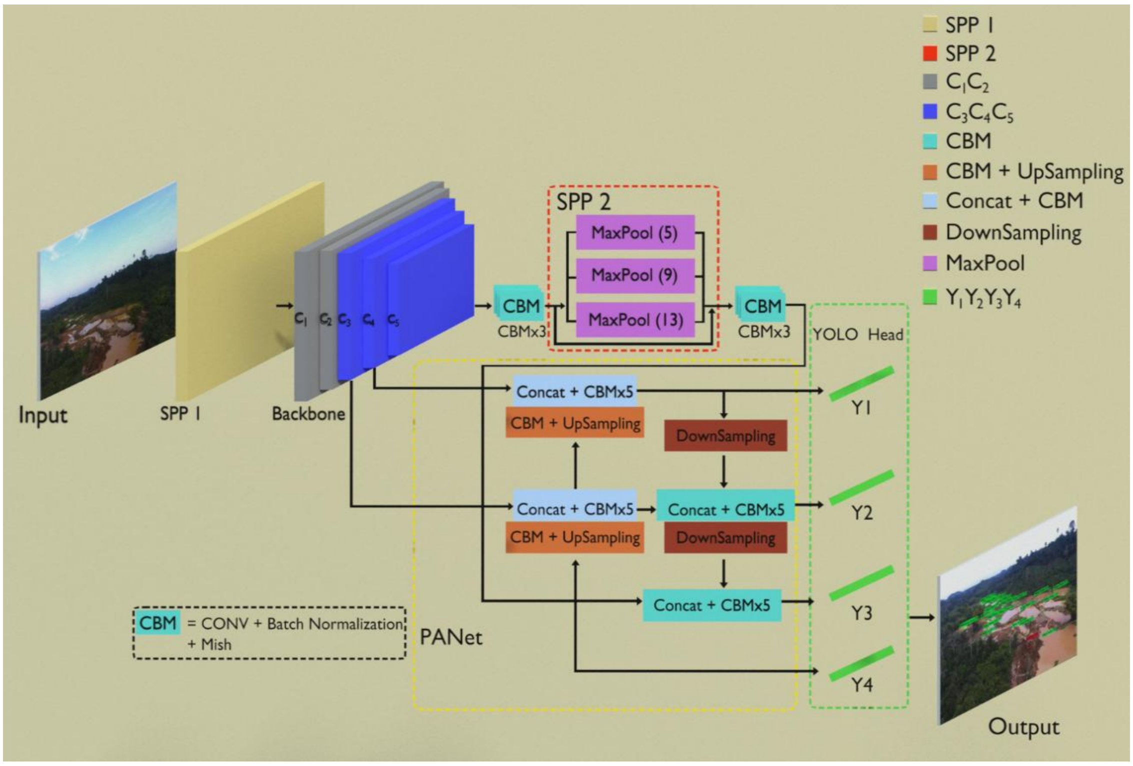


Fig. 7 Network Architecture of the ASGM-YOLO Algorithm

The input images were passed through the SPP-1 layer attached to the front of the CSPDarknet-53 backbone network. The introduction of the SPP-1 layer before the CSPDarknet-53 backbone helped to solve the problem of varying input image sizes through its multi-scale pooling approach. The SPP-1 also helped to pool important features and generated outputs with fixed- length, which were then fed into the CSPDarknet-53 [43]. By using the SPP layer, more feature-rich image information is obtained, and also great improvements in the network's time efficiency are observed. Hence, this technique shows remarkable detection accuracy. The CSPDarknet-53 acted as the backbone of the ASGM-YOLO algorithm and was responsible for extracting deep feature representations of the objects of interest from the input images through the five Resblock layers (C1-C5) which formed part of the backbone. The ASGM-YOLO algorithm contained 53 convolution layers of sizes 1×1 and 3×3, with each convolution layer connected to a batch normalization (BN) layer and a Mish activation layer. In the Resblock (Fig. 8), *x* represented the input to the network, *R(x)* represented the expected output, while the Residual Network represented the variance *R(x)* − *x* between the input and the output. The Residual units of the Resblock are expressed as:

$$y_k = f(x_k + F(x_k, w_k))$$

$$f = \max(0, x) \qquad (2)$$

where x_k and y_k denote the input and output of the k_{th} residual unit, f represented the activation function $Mish$ activation. The Mish activation function was employed in the network structure of the ASGM-YOLO algorithm as it introduced non-linearity into the network. The Mish activation function was expressed as:

$$Mish = x \ \times \tan h\,(1n\,(1 + e^{x})) \qquad (3)$$

For the proposed ASGM-YOLO algorithm to extract highly descriptive feature information of small target objects from the UAV images as well as improve the detection rate of the small target objects, a 152 × 152 feature map which was obtained by the second residual module was used to detect the target objects because it contained smaller target location information. Upsampling was performed twice on the 8 times downsampling feature map output by the proposed ASGM-YOLO algorithm to obtain the 152 × 152 feature map, and the 2 times upsampling feature map is connected to the feature map obtained from the second residual module in the CSPDarknet53 network structure. A feature fusion target detection layer with an output of 4 times downsampling is established to detect small targets. In addition, the 2 residual units of the second residual module in the CSPDarknet53 network structure are increased to 4 residual units, and the 4 residual units of the fifth residual module in the original CSPDarknet53 network structure are reduced to 2. The Neck network is mainly composed of the SPPNet and improved PANet. In this paper, the SPPNet module was used to enlarge the acceptance range of backbone features effectively, and this significantly separated the most important contextual features. The Prediction module made use of the features extracted from the model to make predictions. In this paper, the prediction network is divided into four effective feature layers: 13 × 13 × 24, 26 × 26 × 24, 52 × 52 × 24 and 104 × 104 × 24, which correspond to big object, medium object, small object and very small dense objects respectively. Here, 24 can be understood as the product of 3 and 8, and 8 can be divided into the sum of 4, 1 and 3, where 4 represents the four position parameters of the prediction box, 1 is used to judge whether the prior box contains objects, and 3 represents that there are three categories of mask detection tasks. The PANet employed both top-down and bottom-up approaches to extract the features of interest. Another key contribution was the introduction of the SPP-2 layer into the tail of the modified YOLOv4 to separate the features of interest and facilitate the neck network fusing the global feature information. The SPP-2 further merged the output features of the pooling layers and sends them to the next convolutional module to perform additional feature learning to obtain the local features of interest. By introducing the SPP-2-layer, deep feature representations were obtained from the images as well as great enhancements in the performance of the network. Four YOLO heads of sizes 19×19, 38×38, 76×76, and 152×152 are then employed to interact with feature maps at different scales to detect objects of different sizes. YOLOv4 adopts three detection layer heads of the original YOLOv3 model. The fourth detection layer was added to enable the ASGM-YOLO algorithm to extract more geometric features that are concatenated with deeper-level features. The addition of the fourth detection layer head also helped to obtain more comprehensive features that enhanced the performance of the proposed model to detect small objects compared with the original YOLOv4 model. To further improve the performance of the ASGM-YOLO algorithm, 12 new anchor box sizes were generated and evenly distributed to the four detection layer heads based on their size. The ASGM-YOLO showed improved signs of efficient detection in the experimental results.

3.3.2 ASGM-YOLO Loss Function

The loss function of the original YOLOv4 algorithm presented the same errors for detecting large and small or densely occluded objects, which hindered the detection accuracy for predicting the presence of neighbouring objects. To this end, when two objects appeared in the same grid cell, the more imposing target object was detected thus presenting a challenge to detect very small objects. Compared to the original YOLOv4 loss function, the loss function for the ASGM-YOLO was optimised to use a single loss function for both bounding box prediction and object classification. The loss function for the ASGM-YOLO algorithm was expressed in five main parts: the first and second parts focused specifically on the loss of bounding box coordinates, the third and fourth focused on the difference in the confidence of the presence or absence of an object in a particular grid cell and the fifth part responsible for the difference in class probability. Mathematically, the loss function of the ASGM-YOLO algorithm was computed as the sum of object classification loss, confidence loss, and bounding box regression loss is expressed as:

$$L_{text} = L_{box} + L_{confidence} + L_{class} \tag{4}$$

The confidence and classification losses are expressed as:

$$\begin{aligned} L_{confidence} &= -\sum_{i=0}^{S\times S}\sum_{J-0}^{B} I_{ij}^{Obj}\left[\hat{c}_i \log(C_i) + (1+\hat{c}_i)\log(1-C_i)\right)] \\ &= \lambda noobj \sum_{i=0}^{S\times S}\sum_{j-1}^{B} I_{ij}^{Obj}\left[\hat{c}_i \log(C_i) + (1 + \hat{c}_i)\log(1-C_i)\right)] \end{aligned} \tag{5}$$

$$L_{class} = -\sum_{i=0}^{S\times S} I_i^{obj} \sum_{c\in class} [p_i(c)\log\big(p_i(c)\big) + (1 + p_i(c))\log(1-p_i(c))] \tag{6}$$

Expressions (5) and (6) are described in Table 3.

Table 3 Mathematical Representation of the Classification and Confidence Loss Functions

Notation	Description
S × S	Total number of grid cells for each feature map
B	Number of possible predictors contained in each grid cell
I_{ij}^{obj}	Denote the presence of a target object contained in the j^{th} bounding box of each i^{th} grid cell
I_{ij}^{noobj}	Denote the presence or absence of a target object in the j^{th} bounding box of each i^{th} grid cell
$\lambda noonbj$	Refer to balancing constraints that predict the presence or absence of a target object.
$\hat{C}_i$ and C_i	Represent the true and predicted confidence of the presence of a target object.
I_i^{obj}	Represent whether the target object is present in the cell i
$\hat{p}_i(c)$	Denote the true probability of the target object in a grid cell
$p_i(C)$	Denote the correctly predicted value of the presence of a target object

To compute the bounding box regression loss, a Complete Intersection over Union (CIoU) loss and Mean Square Error (MSE) is presented. To achieve this, the inclined boundary box regression based on MSE loss was implemented. The MSE loss is expressed as:

$$\begin{aligned} L_{box} = \lambda_{coord} \sum_{i=0}^{S\times S}\sum_{j-0}^{B} I_{ij}^{obj}\left(2-\widehat{w}\times\hat{h}_i\right)[(x_i-\hat{x}_i)^2 \\ + (y_i-\hat{y}_i)^2] \\ + \lambda_{coord}\sum_{i=0}^{S\times S}\sum_{j-0}^{B} I_{ij}^{obj}\big(2 \\ - \widehat{w}_i\times\hat{h}_i\big)\left[(w_i-\widehat{w}_i)^2 \right. \\ \left. + \left(h_i-\hat{h}_i\right)^2\right] \end{aligned} \tag{7}$$

where

λcoord refers to a balancing parameter with its parameter value set to 1. I_{ij}^{noobj} shows whether the absence of a target object is in the *j*th bounding box of the *i*th grid cell. (x_i, y_i, w_i, h_i) *and* $(\hat{x}i, \hat{y}i, \widehat{w}i, \hat{h}i)$ denote the height, and width center coordinates of the predicted bounding box and that of the ground truth.

The CIoU loss was expressed as follows:

$$L_{box} = 1 - IoU + \frac{P^2(b \cdot b^{gt})}{C^2} + \propto v \quad (8)$$

$$\propto = \frac{v}{(1 - IoU) + v} \quad (9)$$

$$\emptyset = \frac{4}{\pi^2} \times \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (10)$$

where

IoU denotes the intersection over Union between the predicted bounding box and the ground truth bounding box. $P^2(b, b^{gt})$ was the Euclidean distance, and c is the diagonal length of the smallest enclosing box covering the bounding boxes, (w, h) and w^{gt}, h^{gt} denote the height and width of the predicted bounding box and the ground truth respectively.

3.4 Model Evaluation Criteria

To assess the superiority of the proposed ASGM-YOLO, experimental analysis was carried out between the ASGM-YOLO, R-CNN, Fast R-CNN, Faster R-CNN and YOLOV4 respectively. This paper introduced Accuracy (A), Precision (P), Recall rate (R), F1 Score (F1), IoU, Average Precision (AP) and Frames per Second (FPS) were quantitively used to evaluate the performance of the algorithms. A detailed description of the performance metrics can be found in the literature [44].

The expressions of A, P, F1 Score and R are as follows:

Accuracy is the most intuitive multiclass performance measure used for validating the performance of image classifiers [45]. It is defined as the ratio of correctly predicted observations to the total observations. Accuracy was expressed as:

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Sample} \quad (11)$$

where

TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative.

Precision measures the accuracy of a classifier to correctly predict positive observations against the total predicted positive observations. It defines the percentage of True Positive (TP) predictions among all other detections made by the system. Precision is computed as:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive + False\ Positive}}$$ (**Error! No text of specified style in document.**12)

F1 score is interpreted as the weighted average of precision and recall. F1 Score is expressed as:

$$\boldsymbol{F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}} \quad (13)$$

Recall measures the ratio of correctly predicted positive observations to all observations in the actual class (Alsalem*et al.*, 2018). The recall is expressed as a fraction of positive instances that are correctly classified as computed as:

$$\boldsymbol{Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}} \quad (14)$$

In the context of object detection, IoU metric captures the similarity between the predicted region and the ground truth region for an object present in the image and is calculated as the size of the intersection of predicted and ground-truth regions divided by their union. IOU is expressed as:

$$\boldsymbol{IoU = \frac{Bgroundtruth \cap Bpredicted}{Bgroundtruth \cup Bpredicted}}$$ **Error! No text of specified style in document.**(15)

FPS represents the total number of images that can be successfully detected in a second.

AP is computed as the average accuracy rate, which is the integral of the P index to the R index. mAP is the average accuracy of the mean, which means that the AP value of each category is added, and then divided by all categories. AP and mAP are expressed as follows:

$$\boldsymbol{AP = \int_0^1 P(R)dR} \quad (16)$$

$$\boldsymbol{mAP = \frac{1}{Q_R} \sum_{q=Q_R} AP_{(q)}} \quad (17)$$

where Q_R is the total number of categories.

4. RESULTS AND DISCUSSION

4.1 Training the ASGM-YOLO Algorithm

The ultimate aim of training the ASGM-YOLO algorithm is to minimize the loss function to reduce the training loss. To achieve that, multi-scale training was enabled by rescaling the input image to a defined pixel size of 416 × 416. Following the default configurations made in the CSPDarknet-53 framework, the ASGM-YOLO algorithm was trained and optimised using the Adaptive Moment Estimation (Adam) optimizer [46] with a momentum of 0.9, at a learning rate of 0.001. The highest number of training epochs equaled 31,000 on the six-class dataset made up of 19 440 static aerial images. To reduce the memory consumption rate on the computer system, the batch size and subdivision were respectively set to 64 and 32. Weight decay of 0.0005 was used to test the model. Also, to enable the ASGM-YOLO algorithm to efficiently learn the disparities in the appearance of the objects of interest in the UAV captured images, backpropagation was carried out over the grid cells with high confidence scores. The grid cells that corresponded to the ground-truth bounding boxes were deemed to have high true positive predictions and those that did not correspond to the ground-truth bounding boxes were deemed to have false-positive predictions. During the final phase of training the ASGM-YOLO algorithm, Non-Maximum Suppression (NMS) operation was performed using a threshold of 0.5, with the predicted bounding boxes overlaid on the image to form the final output. This approach was adopted to stabilise the training of the ASGM-YOLO algorithm. Finally, to assess the effectiveness of the proposed ASGM-YOLO algorithm, the evaluation carried out was based on the loss function convergence curve. During the training phase, the loss function was intuitively fine-tuned to reflect the convergence stability of the ASGM-YOLO algorithm as the number of iterations increased. The loss function curve is shown in Fig. 8.

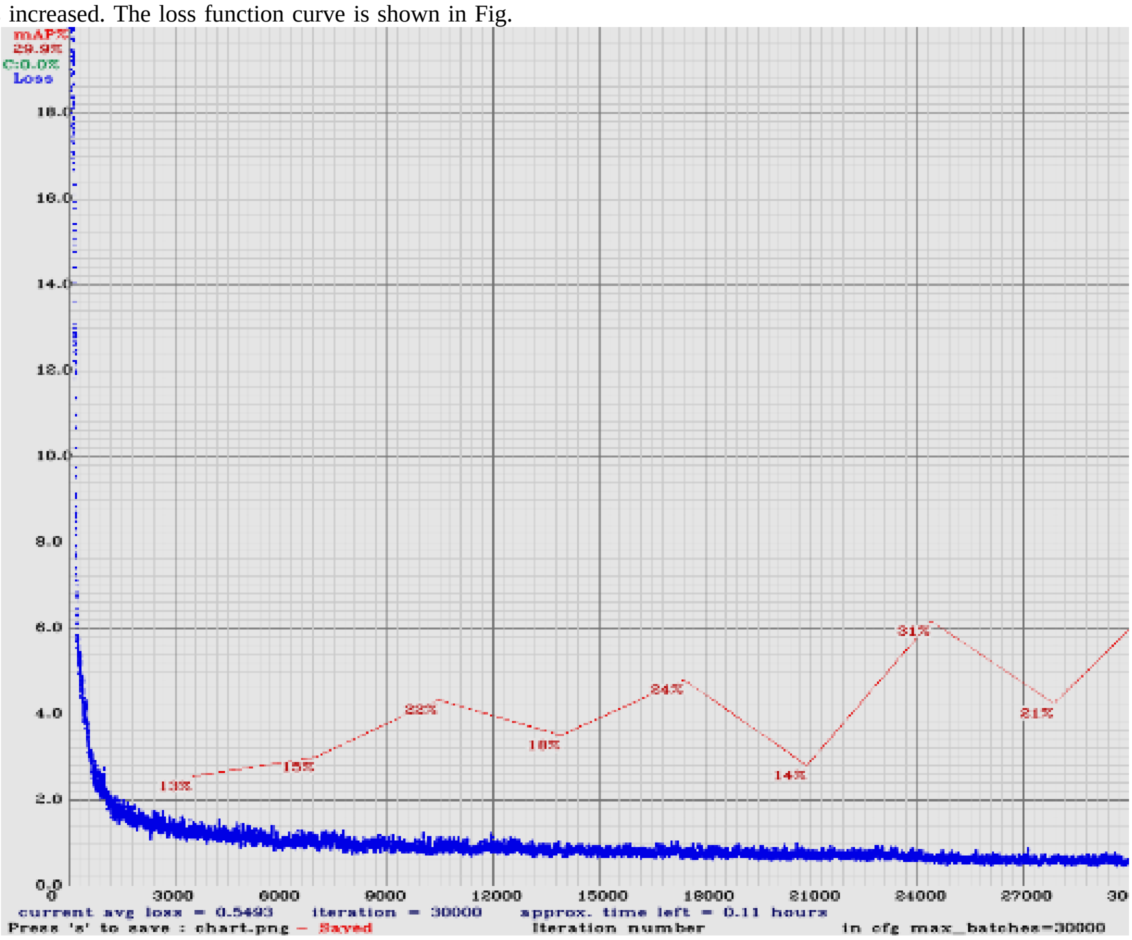


Fig. 8 Loss Function Convergence Curve of the ASGM-YOLO Algorithm

It can be deduced from Fig. 8 that, as the number of training iterations increased, the ASGM-YOLO algorithm loss function curve converged, with the loss value steadily decreasing. The value of the loss function dropped to a significantly low value at the 6000 iterations, however, it achieved a relatively stable convergence level when the number of iterations reached 21,000 with an average loss of 0.5493. This proved the robustness of the ASGM-YOLO algorithm. Table 4 shows the performance results of the ASGM-YOLO algorithm as against

Regional Convolutional Neural Network (R-CNN), Fast R-CNN, Faster R-CNN, and YOLOv4.

4.2 Comparison of Different Models

To assess the superiority of the ASGM-YOLO algorithm, four other object detector pipelines: R-CNN, Fast R-CNN, Faster R-CNN, and YOLOv4 were used in the experiment to establish the results of the proposed ASGM-YOLO algorithm. The algorithms were all trained and tested on the ASGM dataset which had a defined input image resolution of 416 × 416 pixels set at a batch size of 1. This was important as it maintained consistency with the training image resolution. The experimental results as shown in Table 4 proved that the ASGM-YOLO algorithm was able to efficiently detect the objects of interest: excavators, persons, crushers, tailings dump, trucks, and sluice boards in the test dataset thus achieving excellent detection results. From the results, it was noticed that the performance of all algorithms was very high, which was mainly attributed to the use of the small dataset. Meanwhile, it was no doubt that the data augmentation techniques applied to the dataset contributed immensely to the excellent performance of the algorithms, most especially, the proposed ASGM-YOLO. It can further be deduced from Table 4 that the ASGM-YOLO algorithm achieved the best detection accuracy of 96.50% which was closely followed by YOLOv4, Faster R-CNN, Fast R-CNN, and R-CNN which produced 94.88%, 89.34%, 87.68%, and 83.26% detection accuracies respectively. The understanding here is that the ASGM-YOLO algorithm had the least false detection rate of 3.50% while R-CNN, Fast R-CNN, Faster R-CNN, and YOLOv4 had 5.12%, 10.66%, 12.32%, and 16.74% respectively. The average Intersection of Union (IoU) results of 55.94%, 59.28% and 58.43% achieved by R-CNN, Fast R-CNN and Faster R-CNN respectively showed that these algorithms had poor recognition effects on distant objects on interest, thereby strengthening the detection performance of the proposed ASGM-YOLO algorithm as it had a good generalization ability.The detection time of the proposed ASGM-YOLO algorithm YOLO-Tomato-C is 26.4 ms per image on average, which is about 4.4 ms less than the detection time for YOLOv4 with Faster R-CNN, Fast R-CNN and R-CNN using detection times of 44.2 ms, 48.6 ms and 52.4 ms respectively. This gives an indication that the proposed ASGM-YOLO algorithm can perform real-time object detection in complex environments such as an ASGM site. Finally, the dominance of the ASGM-YOLO algorithm can be attributed to the following: (i) feature enhancement attribute provided by the CSPDarknet53 backbone; (ii) the introduction of the SPP-1 layer into the front of the backbone helped to use a definite input image size which enhanced the learning ability of the proposed ASGM-YOLO algorithm; (iii) the introduction of the SPP-2 layer into the tail of the ASGM-YOLO algorithm facilitated the separation of features of interest and merged the output features of the pooling layers and sent them to the next convolutional module to perform additional feature learning to obtain the local features of interest and (iv) the introduction of the PANet layer which improved the detection capability of the proposed algorithm to detect the objects of interest.

Table 4. Performance Evaluation of the Proposed ASGM-YOLO Algorithm

Deep Learning Techniques	**Performance Indicators**							
	Accuracy (%)	**Precision (%)**	**F1-Score (%)**	**mAP (%)**	**Avg. IoU (%)**	**Recall (%)**	**FPS**	**Time (ms)**
R-CNN	83.26	79.67	79.67	74	55.94	82.9	63	52.4
Fast R-CNN	87.67	61.54	65.38	78	59.28	65.3	78	48.6
Faster R-CNN	89.34	73.89	71.87	71.40	58.43	84.7	87	44.2
YOLOv4	94.88	69.25	76.49	94.45	68.85	91.2	95	32.8
ASGM-YOLO	**96.50**	**82.47**	**79.82**	**86.59**	**75.38**	**76.33**	**135**	**26.4**

From Table 4, the values for precision are interpreted as the exact measure of classification after the objects are predicted. The recall measured the ASGM-YOLO capability to detect positive instances by measuring the fraction of positive image instances that were correctly classified. The F1 Score represented the overall model performance which was categorised in the range of zero and one, with high values indicating high classification performance and vice versa. In the end, amAP of 86.59% was obtained by finding the average of the reported average precision results for the images. The detection time of the ASGM-YOLO was 26.4 ms per image on average. This indicates that the proposed ASGM-YOLO can perform better in near real-time object detection in ASGM environments.

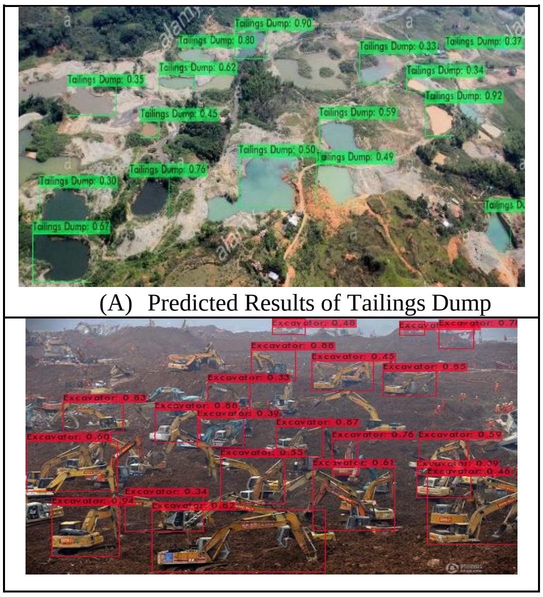
4.3 ASGM-YOLO Visualization

The visualization results of the ASGM-YOLO are shown in Fig. 9. The results show the detected objects and features of interest with their percentage scores. Six images were randomly selected from the ASGM dataset. The findings revealed the following: the predicted green BBs denote tailings dump class; the red BBs represent all detected excavator class, the yellow BBs represent person class, the light green BBs represent sluice board class, the pink BBs represent the crusher class and the blue BBs represent the truck class. From the six selected images, it can be seen the ASGM-YOLO algorithm successfully detected all the objects of interest. The difference in object detection is the confidence scores achieved in detecting some different objects. Table 5 displays the results of the confidence scores of the respective classes in the dataset.

Table 5. Results of Confidence Scores of the Six Classes

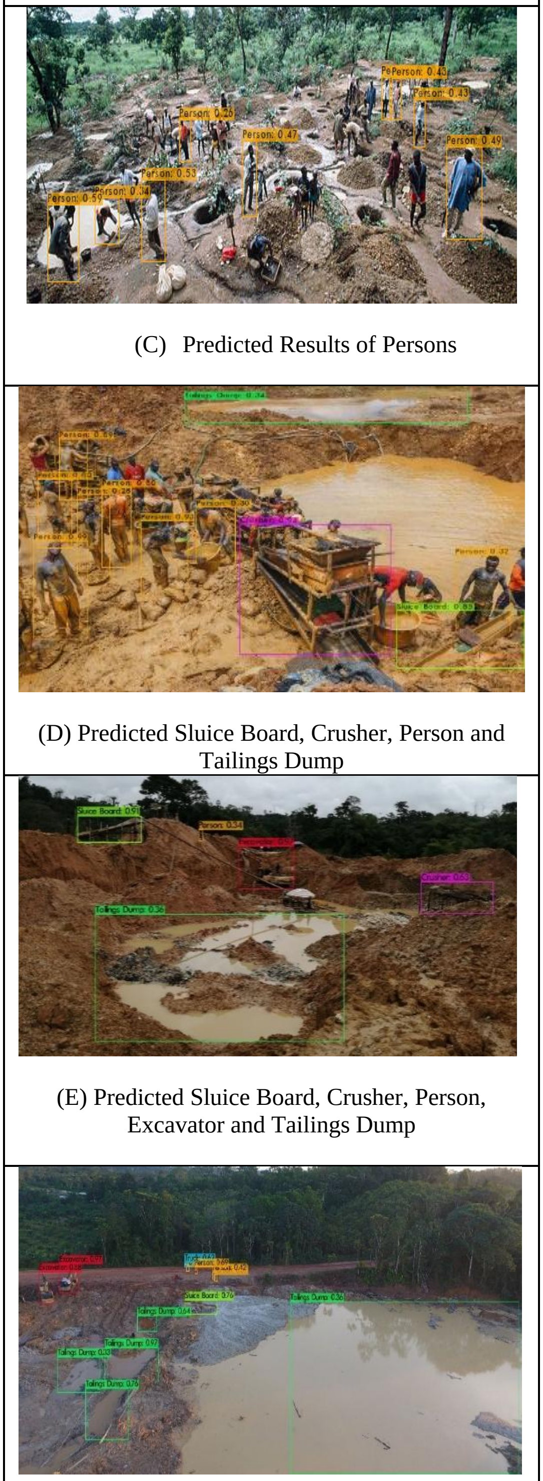
SN	Class Categorization	Confidence Score
A	Tailings Dump	0.90, 0.80, 0.62, 0.35, 0.33, 0.37, 0.34, 0.92, 0.59, 0.45, 0.59, 0.50, 0.49, 0.70, 0.34, 0.30, 0.36, 0.76, 0.33, 0.97, 0.64, 0.76, 0.36 and 0.67
B	Excavator	0.40, 0.78, 0.88, 0.45, 0.85, 0.33, 0.83, 0.88, 0.39, 0.60, 0.94, 0.87, 0.76, 0.59, 0.39, 0.46, 0.97, 0.88, 0.97, 0.34, 0.82, 0.55 and 0.61
C	Person	0.43, 0.43, 0.49, 0.26, 0.47, 0.53, 0.34, 0.99, 0.93, 0.30, 0.28, 0.80, 0.80, 0.34, 0.69, 0.89, 0.42 and 0.59
D	Crusher	0.94 and 0.63
F	Truck	0.63
G	Sluice Board	0.85, 0.91 and 0.76

From these values, it can be deduced that the proposed ASGM-YOLO performed better on detecting tailings dumps and excavators compared to the truck, sluice board and crusher classes. The reason for the poor performance of the model on the truck, sluice board and crusher classes are due to the small amount and low diversity of those classes in the ASGM-21 data-set. Also, taking into consideration the heterogeneous nature of the ASGM environments coupled with the irregular nature of some of the objects of interest particularly, tailings dump and excavators, it was clear that the ASGM-YOLO algorithm was able to accurately detect all the objects of interest in the test images. Even though in some cases, the percentage detections in the proposed algorithm were a bit low, the missed object detection in itself was also very low. This further shows the importance of the feature enhancement provided by SPP-1 and SPP-2 to the ASGM-YOLO algorithm.



(A) Predicted Results of Tailings Dump

(B) Predicted Results of Excavators



(C) Predicted Results of Persons

(D) Predicted Sluice Board, Crusher, Person and Tailings Dump

(E) Predicted Sluice Board, Crusher, Person, Excavator and Tailings Dump

(F) Predicted Truck, Excavator, Sluice Board, Person and Tailings Dump

Fig. 9 Visualization Results Predicted by the ASGM-YOLO Algorithm

5. CONCLUSION

The study proposed the ASGM-YOLO algorithm which is based on the YOLOv4 algorithm to detect objects of interest such as excavators, crushers, tailings dumps, persons, sluice boards, and trucks from UAV captured images at ASGM sites. LabelImg software was used to annotate the images in the dataset and a 16-way multiple data augmentation technique was adopted and applied on the ASGM dataset which introduced more variability into the dataset as well as helped the proposed ASGM-YOLO algorithm to generalise well. The SPP-1 layer was introduced into the CSP-DarkNet53 backbone of the ASGM-YOLO algorithm which restricted the input images to a defined pixel size of 416 × 416. SPP-2 layer introduced into the tail of the ASGM-YOLO algorithm facilitated the separation of features of interest and disregarded irrelevant information. Batch Normalization and mish activation functions were employed to accelerate the training of the ASGM-YOLO algorithm by using higher learning rates as well as minimise detection errors. From the observation, the proposed ASGM-YOLO algorithm proved superior by achieving the highest Accuracy – 96.50%, Precision – 82.47%, F1-Score – 79.82%, mAP – 86.59%, Avg. IoU – 75.38% and Recall – 76.33% compared to YOLOv4, Faster R-CNN, Fast R-CNN and R-CNN which indicated the dominance of the proposed model. Above all, the ASGM-YOLO algorithm generalised well and showed near real-time capabilities for detecting objects in ASGM environments. Even though the model proved efficient in detecting these objects, it still produced quite a high FP prediction. Therefore, future works will be focused on identifying ways to reduce or completely do away with the false positives. Due to the heterogeneous nature of ASGM environments, another future direction will be to introduce class borders around the detected objects to eliminate confusion caused by nearby objects in the background.

6. ACKNOWLEDGEMENT

Our thanks go to Joseph Adiba (Geomatics Department of the University of Mines and Technology) for assisting us to fly the Phantom drone to capture aerial images at the respective ASGM sites during the data collection phase. Our appreciation goes to JaphetKuntu-Blankson who supported us to annotate all the images used to train the models.

7. REFERENCES

[1] E. Stemn, B. Kumi-Boateng, “Spatial Analysis of Artisanal and Small-Scale Mining in the Tarkwa-Nsuaem Municipality of Ghana”, Ghana Mining Journal, 20(1): 66–74, 2020.

[2] J. C. Hodgson, R. Mott, S. M. Baylis, T. T. Pham, S. Wotherspoon, A. D. Kilpatrick, R. R. Segaran, I. Reid, A. Terauds, L. P. Koh, N. Yoccoz, “Drones Count Wildlife More Accurately and Precisely than Humans”, Methods in Ecology and Evolution, 9(5): 1160–1167, 2018.

[3] Bhaskaranand, M. and Gibson, J. D. Low-Complexity Video Encoding for UAV Reconnaissance and Surveillance, In Proceedings of the IEEE Military Communications Conference (MILCOM), pages 1633–1638, IEEE, 2011.

[4] H. Püschel, M. Sauerbier, H.Eisenbeiss, A 3D Model of Castle Landenberg (CH) from Combined Photogrammetric Processing of Terrestrial and UAV-based Images.The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37: 93-98, 2008.

[5] A.Y.-M. Lin, A. Novo, S. Har-Noy, N. D. Ricklin, K. Stamatiou, “Combining GeoEye-1 Satellite Remote Sensing, UAV Aerial Imaging, and Geophysical Surveys in Anomaly Detection Applied to Archaeology”, Journal of Remote Sensing, 4: 870-876, 2011.

[6] R. Chandra,“FarmBeats: Automating Data Aggregation”, Farm Policy Journal, 15: 7-16, 2018.

[7] Bae, S. M., Han, K. H.., Cha, C. N. and Lee, H. Y. Development of Inventory Checking System Based on UAV and RFID in Open Storage Yard.In Proceedings of the International Conference on Information Science and Security (ICISS), pages 1-2, 2016.

[8] I. Sa, S. Hrabar, “Outdoor Flight Testing of a Pole Inspection UAV Incorporating Highspeed Vision”, Journal in Advanced Robotics, 105: 107-121, 2015.

[9] Moranduzzo, T. and Melgani, F. A SIFT-SVM method for detecting cars in UAV images.In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, pages 6868-6871, IEEE, 2012.

[10] J. Linchant, J.Lisein, Semeki, J., P. Lejeune, C. Vermeulen, “Are Unmanned Aircraft Systems (UASS) the Future of Wildlife Monitoring? A Review of Accomplishments and Challenges”, Journal of Mammal Review,45: 239-252, 2015.

[11] R. D´ıaz-Delgado, M. Manez, A. Martınez, D. Canal, M. Ferrer, and D. Aragones, “Using UAVs to Map Aquatic Bird Colonies”The Roles of Remote Sensing in Nature Conservation, Springer,277-291, 2017.

[12] A. Hodgson, N. Kelly, D. Peel, “Unmanned Aerial Vehicles (UAVs) for Surveying Marine Fauna: A Dugong Case Study”, PloS One, 8(11): 1-15, 2013.

[13] I. Goodfellow, Y. Benjio, A. Courvile, “Deep Learning”,The MIT Press, pages 800, 2016.

[14] F. Chollet, “Deep Learning with Python”, Manning Publications, pages 384, 2017.

[15] Y. LeCun, Y. Bengio, G. Hinton, “Deep Learning”, Nature, 521: 436-444, 2015.

[16] Wu, J., Peng, B., Huang, Z. and Xie, J. Research on Computer Vision-Based Object Detection and Classification.In Proceedings of the International Conference on Computer and Computing Technologies in Agriculture, 2: pages 183-188, 2012.

[17] V. Wiley, T. Lucas, “Computer Vision and Image Processing: A Paper Review”, International Journal of Artificial Intelligence Research,2: 28-36, 2018.

[18] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview”,Journal of Neural Networks, 61: 85-117, 2015.

[19] S. Ren, K. He, R. Girshick, J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39: 1137-1149, 2017.

[20] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, 1: pages 1097-1105, 2012.

[21] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770-778, 2016.

[22] Redmon, J., and Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, pages 6517-6525, 2017.

[23] Mantey, S. and Otoo, E. K. Comparison of Analytical and Numerical Water Influx Models in Bottom Water Reservoir. In Proceedings of 6thUMaT Biennial International Mining and Mineral Conference, pages 1-6, 2020.

[24] J. M. Kusimi, “Assessing Land Use and Land Cover Change in the Wassa West District of Ghana Using Remote Sensing”, Journal of Geology, 71: 249-259, 2008.

[25] P. L. Basommi, Q. Guan, D. Cheng, “Exploring Land Use and Land Cover Change in the Mining Areas of Wa East District, Ghana Using Satellite Imagery”, Journal of Open Geosciences, 7: 618-626, 2015.

[26] B. Snapir, D. M. Simms, T. W. Waine, “Mapping the Expansion of Galamsey Gold Mines in the Cocoa Growing Area of Ghana using Optical Remote Sensing”, International Journal of Applied Earth Observation and Geoinformation, 58: 225-233, 2017.

[27] F. Owusu-Nimo, J. Mantey, K. B.Nyarko, E. Appiah-Effah, A. Aubynn, "Spatial Distribution Patterns of Illegal Artisanal Small-Scale Gold Mining (Galamsey) operations in Ghana: A Focus on the Western Region", Journal of Heliyon,4(2): 1-36, 2018.

[28] G. Forkuor, T. Ullmann, M. Griesbeck, “Mapping and Monitoring Small-Scale Mining Activities in Ghana using Sentinel-1 Time Series (2015–2019)”, Journal of Remote Sensing, 12: 1-26, 2020.

[29] E. Ibrahim, L. Lema, P. Barnabé, P. Lacroix, E. Pirard,“Small-Scale Surface Mining of Gold Placers: Detection, Mapping, and Temporal Analysis through the use of Free Satellite Imagery”, International Journal of Applied Earth Observation and Geoinformation, 93: 1-11, 2020.

[30] P. de B. Pozzobon, D. de C. J. Osmar, F. G. Renato, A. T. G. Roberto, “Change Detection of Deforestation in the BrazilianAmazon using Landsat Data and Convolutional Neural Networks”, Journal of Remote Sensing,12(6): 1-9, 2020.

[31] R. Balaniuk, O. Isupova, S. Reece, “Mining and Tailings Dam Detection in Satellite Imagery Using Deep Learning”, Journal of Sensors, 20(23): 1-27, 2020.

[32] Q. Li, Z. Chen, B. Zhang, B. Li, K. Lu, L. Lu, H. Guo,“Detection of Tailings Dams Using High-ResolutionSatellite Imagery and a Single Shot MultiboxDetectorin the Jing–Jin–Ji Region, China”, Journal of Remote Sensing, 12(16): 1-18, 2020.

[33] C. Nyamekye, B. Ghansah, E. Agyapong, S. Kwofie, “Mapping Changes in Artisanal and Small-Scale Mining (ASM) Landscape using Machine and Deep Learning Algorithms - A Proxy Evaluation of the 2017 Ban on ASM in Ghana”, Elsevier,1-15, 2021.

[34] Zuiderveld, K. 1994) Contrast Limited Adaptive Histogram Equalization, In Graphics Gems; Heckbert, P.S., Ed.; Academic Press: Cambridge, MA, USA, pp. 474 – 485.

[35] Tzutalin. 2015. LabelImg (version 1.8.3). Gitcode.

[36] S.-H. Wang, V. V. Govindaraj, J. M. Górriz, X. Zhang,

Y. D. Zhang, “Covid-19 classification by FGCNet with Deep Feature Fusion from Graph Convolutional Network and Convolutional Neural Network”, Journal of Information Fusion, 67: 208–229, 2021.

[37] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection.InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779 – 788.

[38] Redmon, J., and Farhadi, A. 2017. YOLO9000: Better, Faster, Stronger, In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, pages 6517–6525.

[39] Redmon, J. and Farhadi, A. 2018. YOLOv3: An incremental improvement, https://arxiv.org/pdf/1804.02767.pdf, (January, 2021),1 – 6.

[40] Bochkovskiy, C.-Y., Wang, C.-Y. and Liao, H-Y. M. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection, https://arxiv.org/pdf/2004.10934.pdf, (January, 2021), 1-18.

[41] He, K., Zhang, X., Ren, S. and Sun, J. 2014.Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.In Proceedings of the 13th European Conference in Computer Vision, pages 6-12.

[42] Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. 2018. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVFConference on Computer Vision and Pattern Recognition (CVPR), pages 8759 – 8768.

[43] Wang, C Y., Liao, H Y., Wu, Y H., Chen, P-Y., Hsieh, J-W. and Yeh, I-H. 2020.CSPNet : A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 390-391.

[44] O. Janssens, V.Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van De Walle, S. Van Hoecke, “Convolutional Neural Network Based Fault Detection for Rotating Machinery”, Journal of Sound and Vibration, 377: 331-345, 2016.

[45] M. A. Alsalem,A. A. Zaidan,M. Hashim,O. S. Albahri, A. S. Albahri, A. K. Hadi, I.Mohammed, “Systematic Review of an Automated Multiclass Detection and Classification System for Acute Leukaemia in Terms of Evaluation and Benchmarking, Open Challenges, Issues and Methodological Aspects”, Journal of Medical Systems,42: pages 204, 2018.

[46] Kingma, D. P. and Ba, J. 2014Adam: A Method for Stochastic Optimization. In Proceedings of theInternational Conference on Learning Representations, pages 1-15.