Journal of
Imaging

Article

Object Recognition in Aerial Images Using Convolutional Neural Networks

Matija Radovic [1,*], Offei Adarkwa [2] and Qiaosong Wang [3]

[1] Civil and Environnemental Engineering Department, University of Delaware, Newark, DE 19716, USA
[2] Research Associate, Center for Transportation Research and Education, Iowa State University, Ames, IA 50010, USA; oadarkwa@iastate.edu
[3] Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA; qiaosong@udel.edu
[*] Correspondence: mradovic@udel.edu; Tel.: +1-302-831-0529

Received: 17 November 2016; Accepted: 7 June 2017; Published: 14 June 2017

Abstract: There are numerous applications of unmanned aerial vehicles (UAVs) in the management of civil infrastructure assets. A few examples include routine bridge inspections, disaster management, power line surveillance and traffic surveying. As UAV applications become widespread, increased levels of autonomy and independent decision-making are necessary to improve the safety, efficiency, and accuracy of the devices. This paper details the procedure and parameters used for the training of convolutional neural networks (CNNs) on a set of aerial images for efficient and automated object recognition. Potential application areas in the transportation field are also highlighted. The accuracy and reliability of CNNs depend on the network's training and the selection of operational parameters. This paper details the CNN training procedure and parameter selection. The object recognition results show that by selecting a proper set of parameters, a CNN can detect and classify objects with a high level of accuracy (97.5%) and computational efficiency. Furthermore, using a convolutional neural network implemented in the "YOLO" ("You Only Look Once") platform, objects can be tracked, detected ("seen"), and classified ("comprehended") from video feeds supplied by UAVs in real-time.

Keywords: convolutional neural networks; Unmanned Aerial Vehicle (UAV); object recognition and detection

1. Introduction

There are a wide range of applications for unmanned aerial vehicles (UAVs) in the civil engineering field. A few applications include but are not limited to coastline observation, fire detection, monitoring vegetation growth, glacial observations, river bank degradation surveys, three-dimensional mapping, forest surveillance, natural and man-made disaster management, power line surveillance, infrastructure inspection, and traffic monitoring [1–5]. As UAV applications become widespread, a higher level of autonomy is required to ensure safety and operational efficiency. Ideally, an autonomous UAV depends primarily on sensors, microprocessors, and on-board aircraft intelligence for safe navigation. Current civil and military drones have limited on-board intelligence to execute autonomous flying tasks. In most cases, they utilize a Global Positioning System (GPS) for flight operation and sensors for obstacle detection and avoidance. In order for UAVs to be fully autonomous in decision-making, an on-board intelligence module needs to be supplied with appropriate information about its immediate surroundings. Most UAVs rely on integrated systems consisting of velocity, altitude, and position control loops to achieve operational autonomy. Despite its demonstrated reliability, such a system is presently limited in executing highly complex tasks. Fully autonomous UAV decision-making is only possible when the system is able to perform the dual function of object sighting and comprehension, which are referred to as detection and classification, respectively, in computer vision. While these tasks come naturally to humans, they are abstract and complicated for machines to perform on their own. One of the problems currently facing autonomous UAV operation is conducting detection and classification operations in real-time. To solve this problem, authors adapted and tested a convolutional neutral network (CNN)-based software called YOLO ("You Only Look Once"). This detection and classification algorithm was adapted and successfully applied to video feed obtained from UAV in real-time.

This paper is divided into six main parts. The second section covers the motivation for this project, and is presented after the introduction. Previous approaches for object recognition and UAV flight are discussed briefly in the background after the second section. The fourth and fifth parts of the paper focus on the methodology and results, while the conclusion and applications are highlighted in the last section.

1.1. Motivation and Objectives

The primary motivation behind this research is to test CNN image recognition algorithms that can be used for autonomous UAV operations in civil engineering applications. The first objective of this paper is to present the CNN architecture and parameter selection for the detection and classification of objects in aerial images. The second objective of this paper is to demonstrate the successful application of this algorithm on real-time object detection and classification from the video feed during UAV operation.

1.2. Background

Object detection is a common task in computer vision, and refers to the determination of the presence or absence of specific features in image data. Once features are detected, an object can be further classified as belonging to one of a pre-defined set of classes. This latter operation is known as object classification. Object detection and classification are fundamental building blocks of artificial intelligence. Without the development and implementation of artificial intelligence within a UAV's on-board control unit, the concept of autonomous UAV flight comes down to the execution of a predefined flight plan. A major challenge with the integration of artificial intelligence and machine learning with autonomous UAV operations is that these tasks are not executable in real-time or near-real-time due to the complexities of these tasks and their computational costs. One of the proposed solutions is the implementation of a deep learning-based software which uses a convolutional neural network algorithm to track, detect, and classify objects from raw data in real time. In the last few years, deep convolutional neural networks have shown to be a reliable approach for image object detection and classification due to their relatively high accuracy and speed [6–9]. Furthermore, a CNN algorithm enables UAVs to convert object information from the immediate environment into abstract information that can be interpreted by machines without human interference. Based on the available information, machines can execute real-time decision making. CNN integration into a UAV's on-board guidance systems could significantly improve autonomous (intelligent) flying capabilities and the operational safety of the aircraft.

The intelligent flying process can be divided into three stages. First, raw data is captured by a UAV during flight. This is followed by real-time data processing by the on-board intelligence system. The final stage consists of autonomous and human-independent decision-making based on the processed data. All three stages are conducted in a matter of milliseconds, which results in instantaneous task execution. The crucial part of the process is the second stage, where the on-board system is supposed to detect and classify surrounding objects in real time.

The main advantage of CNN algorithms is that they can detect and classify objects in real time while being computationally less expensive and superior in performance when compared with other machine-learning methods [10]. The CNN algorithm used in this study is based on the combination of deep learning algorithms and advanced GPU technology. Deep learning implements a neural network approach to "teach" machines object detection and classification [11]. While neural network algorithms have been known for many decades, only recent advances in parallel computing hardware have made real-time parallel processing possible [12,13]. Essentially, the underlying mathematical structure of neural networks is inherently parallel, and perfectly fits the architecture of a graphical processing unit (GPU), which consists of thousands of cores designed to handle multiple tasks simultaneously. The software's architecture takes advantage of this parallelism to drastically reduce computation time while significantly increasing the accuracy of detection and classification.

Traditional machine learning methods utilize highly complex and computationally expensive feature-extraction algorithms to obtain low-dimensional feature vectors that can be further used for clustering, vector quantization, or outlier detection. As expected, these algorithms ignore structure and the compositional nature of the objects in question, rendering this process computationally inefficient and non-parallelizable. Due to the nature of the UAV operations, where an immediate response to a changing environment is needed, traditional machine learning algorithms are not suitable for implementation in on-board intelligent systems.

As mentioned earlier, the CNN algorithm used in this study is based on deep learning convolutional neural networks which solve the problem of instantaneous object detection and classification by implementing efficient and fast-performance algorithms. In general, these algorithms use the same filter on each pixel in the layer, which in turn reduces memory constraints while significantly improving performance. Due to recent advances in GPU hardware development, the size and price of the GPU unit needed to handle the proposed software has been reduced considerably. This allows the design of an integrated software–hardware module capable of processing real-time detection and classification, but which is light and inexpensive enough to be mounted on a commercial-type UAV without significantly increasing the UAV's unit cost. However, before CNNs are incorporated in a UAV's on-board decision-making unit, they need to be trained and tested. This paper shows that modifying CNN architecture and proper parameter selection yields exceptional results in object detection and classification in aerial images.

2. Methods

2.1. Network Architecture

The CNN algorithm presented in this paper was based on an open-source object detection and classification platform complied under the "YOLO" project, which stands for "You Only Look Once" [14]. The "YOLO" has many advantages over other traditionally employed convolutional neural network software. For example, many CNNs use regional proposal methods to suggest potential bounding boxes in images. This is followed by bounding box classification and refinement and the elimination of duplicates. Finally, all bounding boxes are re-scored based on other objects found in the scene. The issue with these methods is that they are applied at multiple locations and scales. High scoring regions of an image are considered to be detections. This procedure is repeated until a certain detection threshold is met. While these algorithms are precise and are currently employed in many applications, they are also computationally expensive and almost impossible to optimize or parallelize. This makes them unsuitable for autonomous UAV applications. On the other hand, "YOLO" uses a single neural network to divide an image into regions, while predicting bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. The main advantage of this approach is that the whole image is evaluated by the neural network, and predictions are made based on the concept of the image, not the proposed regions.

The "YOLO" approaches object-detection as a tensor-regression problem. The procedure starts by inputting an image into the network. The size of the image entering the network needs to be in fixed format (n × m × 3, where the number 3 denotes 3 color channels). Our preliminary results show that the best-preforming image size is 448 × 448; therefore, we used a 448 × 448 × 3 format in all tests. Following image formatting, an equally sized grid (S × S) is superimposed over the image, effectively dividing it into N number of cells (Figure 1a). Each grid cell predicts the number of bounding boxes (B) and confidence scores for those boxes (Figure 1b).

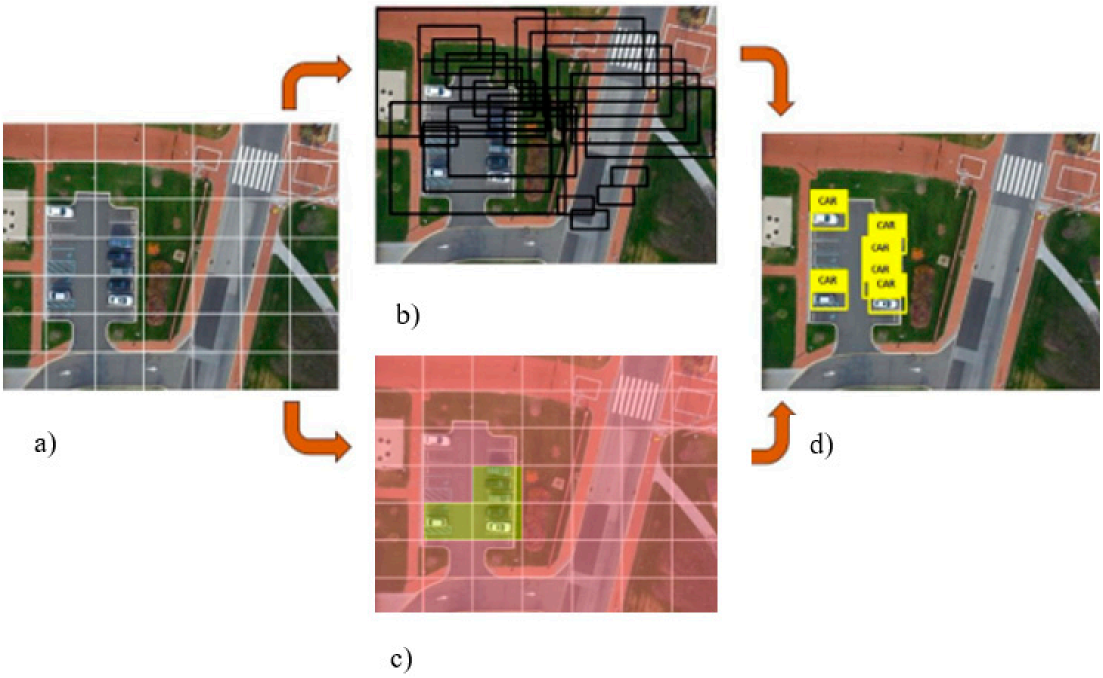


Figure 1. Images captured by unmanned aerial vehicles (UAVs) (**a**) divided into cells using an equally sized grid (**b**,**c**) to uncover key features in the underlying landscape (**d**). The example shows the network designed with grid size S = 7 and number of cells N = 49.

At this point, each bounding box contains the following information: x and y coordinates of the bounding box, width (w), height (h), and the probability that the bounding box contains the object of interest (Pr (Object)). The (x, y) coordinates are calculated to be at the center of the bounding box but relative to the bounds of the grid cell (Figure 1c). The width and height of the bounding box are predicted relative to the whole image. The final output of the process is $S \times S \times (B \times 5 + C)$ tensor, where C stands for the number of classes the network is classifying and B is a number of hypothetical object bounding boxes. Non-maximal suppression is used to remove duplicate detections. During the network training phase, the following loss function was implemented:

$$\lambda_{\text{coor}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \tag{1}$$

where w_i is the width of the bounding box, while h_i is the height of the bounding box, 1_{ij}^{obj} is the function that counts if the jth bounding box predictor in cell i is responsible for the prediction of the object.

The proposed detection network has only 24 convolutional layers followed by two fully-connected layers. This condensed architecture significantly decreases the time for the object detection, while marginally reducing the classification accuracy of detected objects. The 26-layer configuration shown in Figure 2 is preferred for UAV applications due its high computational speed. According to the "YOLO" authors, alternating 1×1 convolutional layers reduces the feature space from that of the preceding layers. The final layer makes object classification supplemented by the probability that the selected object belongs to the class in question and the bounding box coordinates. Both bounding box height (h) and width (w); and x and y coordinates, are normalized to have values between 0 and 1.

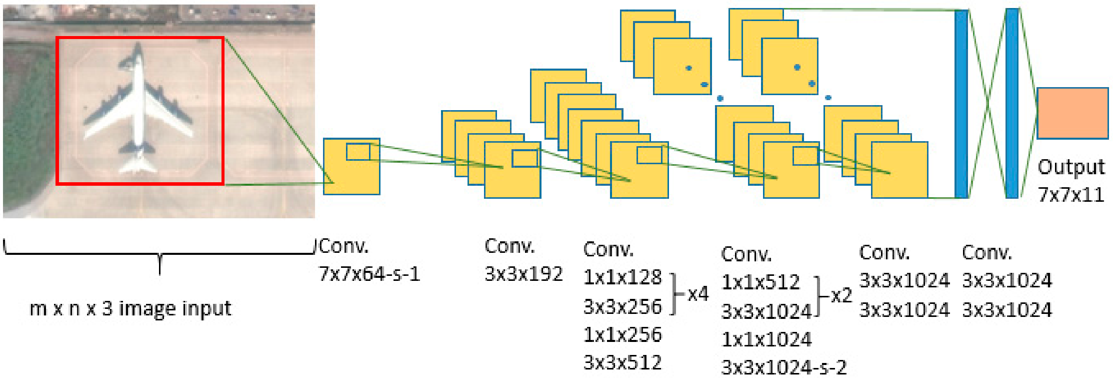


Figure 2. Graphical representation of multilayered convolutional neural network (CNN) architecture.

2.2. Network Training

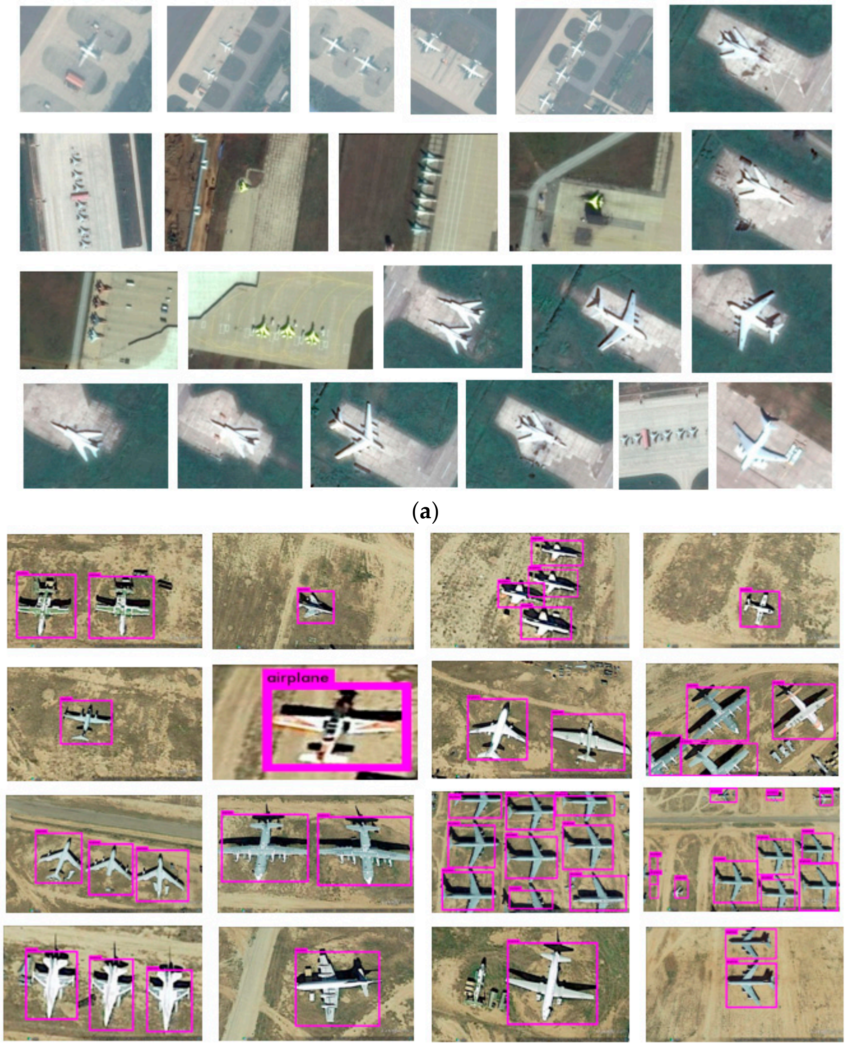
While "YOLO" provides a platform for object detection and classification, the CNN still needs to be trained and the correct parameters need to be determined. The batch size, momentum, learning rate, decay, iteration number, and detection thresholds are all task-specific parameters (defined by the user) that need to be inputted into the "YOLO" platform. The number of epochs that our network needed to be trained with was determined empirically. "Epoch" refers to a single presentation of the entire data set to a neural network. For batch training, all of the training samples pass through the learning algorithm simultaneously in one epoch before weights are updated. "Batch size" refers to a number of training examples in one forward/backward pass. "Learning rate" is a constant used to control the rate of learning. "Decay" refers to the ratio between learning rate and epoch, while a momentum is a constant that controls learning rate improvement. Our network was designed to have 7 × 7 grid structure (S = 7), and was tested on only one object class; i.e., "airplane" (C = 1). This network architecture gives an output tensor with dimensions 7 × 7 × 11.

It is important to note that highly utilized and cited image databases such as PASCAL VOC 2007 and 2012 [15,16], were not used for the training purposes. Preliminary results showed that images taken by UAVs differed significantly from the images available at the PASCAL VOC databases in terms of the scene composition and angle at which images were taken. For example, many images from the PASCAL VOC database were taken from the frontal view, while the images taken by the UAV consist mostly from the top-down view. Therefore, it was not a coincidence that the networks trained on the PASCAL VOC database images alone when tested on UAV acquired video feeds proved to be very unstable with very low recognition confidence (~20%). However, a recognition confidence of 84% was reached when a database containing satellite and UAV-acquired images was used for the training purposes. The learning rate schedule also depended on the learning data set. While it has been suggested that the learning rate rises slowly for the first epochs, this may not be true for our network training. It is known that starting the learning rate at high levels causes models to become unstable. This project provided a unique opportunity to learn how networks behave when exposed to new data sets. To avoid overfitting, dropout and data augmentation were used during network training. For a dropout layer, a rate of 0.5 was used, while for data augmentation, random scaling and translations of up to 35% of the original image size were implemented. Furthermore, saturation and exposure of the image were randomized by up to a factor of 2.5 in the hue saturation value (HSV) color space.

3. Results

3.1. Neural Network Validation

Validation of the CNN was carried out by testing the classification accuracy on a class of objects labeled "airplanes". The class of object "airplanes" was created by downloading satellite images of airplanes grounded on civil and military airfields across the globe from Google Maps (Figure 3a). Images from Google Maps were used due to current restrictions on operating UAVs in airfield proximity. These images consisted of a variety of airplane types and a wide range of image scales, resolutions, and compositions. For example, images were selected in a way to show airplanes up-close and from large distances. There was also variation based on the image composition, with most images having one airplane while others had multiple airplanes. Image quality also varied from high-resolution images (900 dpi) to very low-resolution images (72 dpi), and so on. An airplane object category was created using these images for training the network. There were a total of 152 images containing 459 airplane objects in this training dataset.



(**a**)

(**b**)

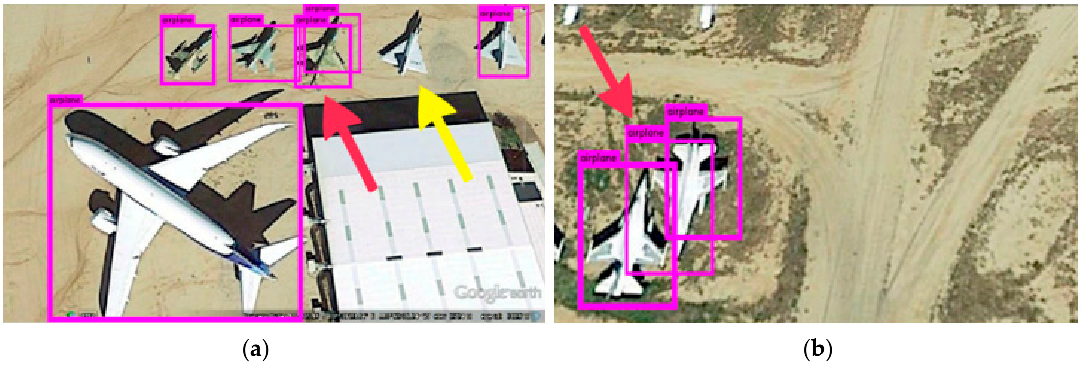
Figure 3. (**a**) Training set of object class "Airplane"; (**b**) Testing set of object class "Airplane".

The open-source tool known as Bounding Box Label [17] was used to label all airplane instances in this dataset, creating ground truth bounding boxes. Throughout the training, batch sizes ranging from 42 to 64 were used, while the momentum and decay parameters were determined empirically by trial and error. The best results were obtained when batch size was 64, momentum = 0.5, decay = 0.00005, learning rate = 0.0001, and iteration number = 45,000. For testing, the detection threshold was set to be 0.2.

For testing CNN recognition accuracy, a new dataset of 267 images containing a total of 540 airplanes was used (Figure 3b). Results showed (Table 1) that the CNN was able to recognize "airplane" objects in the data set with 97.5% of accuracy (526 out of 540 "airplane" objects), while only 16 instances were incorrectly categorized (14 airplanes were not identified and 2 objects were wrongly identified). An incorrectly categorized instance refers to a situation when the image contains an airplane but is not recognized by the network(Figure 4a), or if there is no airplane in the image but one is labeled by the network as being present in the image (Figure 4b). The positive prediction value for our CNN was calculated to be 99.6%, false discovery rate was 0.4%, true positive rate was 97.4%, and false negative rate was 2.6%. More detailed analysis of "YOLO" performance and its comparison to other CNN algorithms was conducted by Wang et al. [9].

Table 1. Confusion matrix for the object class "Airplane".

Classification	**Class**	**Detected**	
		Airplane	**Not Airplane**
Actual	Airplane	526	14
	Not Airplane	2	NA



(**a**) (**b**)

Figure 4. (**a**) Yellow arrow points at the instances where "airplane" object is present but not detected by the CNN, **(b)** red arrow points to the instance where "airplane" object is wrongly identified.

3.2. *Real-Time Object Recognition from UAV Video Feed*

After validation and training, network accuracy was tested in real-time video feed from the UAV flight. Additionally, detection and recognition of multi-object scenarios were also evaluated. "Multi-object scenarios" refers to recognizing more than one class/object in a given image [11]. Real-time/ field testing and results can be viewed using this link [18].

Preliminary tests showed that the CNN was able to detect and identify different object classes in multi-object scenarios in real time from the video feed provided by UAVs with an accuracy of 84%. Figure 5 shows results from testing the algorithm on multi-object scenarios from UAV supplied video feed. Figure 5 shows the image sequence from the video feed in which the CNN is able to detect and recognize two types of objects: "car" and "bus". The CNN was able to accurately detect and classify an object (class) in the image, even if the full contours of the object of interest were obscured by third object, for example, a tree was obscuring the full image of the bus. Furthermore, the CNN was able to classify and detect objects even if they were not fully shown in the image. For example, at the bottom image in Figure 5, only partial contours of two cars were shown and a full contour of the third car. Nevertheless, the CNN was able to accurately detect and classify all three objects as a "car" class.

Based on the high level of detection and classification accuracy attained, there are limitless opportunities in both commercial and military applications. With simple modifications, the approach can be successfully applied in many transportation-related projects. Existing applications in this field including construction site management and infrastructure asset inspections can be greatly enhanced by leveraging the additional intelligence provided by our approach.

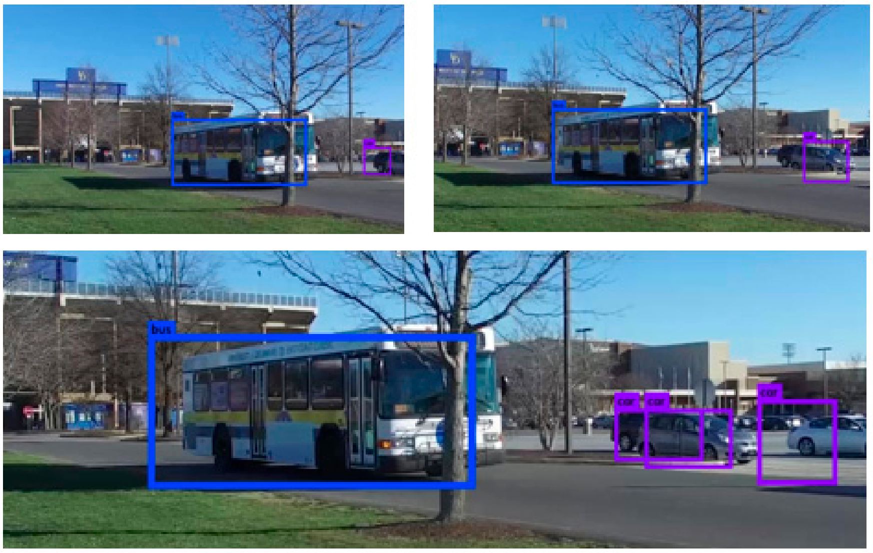


Figure 5. Sample detection of the two object classes: bus (blue bounding box) and car (blue-violet bounding box) from the video sequence obtained by drone and recognized in real time by CNN trained on aerial images.

4. Conclusions

The CNN approach for object detection and classification in aerial images presented in this paper proved to be a highly accurate (97.8%) and efficient method. Furthermore, authors adapted and then tested "YOLO"—a CNN-based open-source object detection and classification platform—on real-time video feed obtained from a UAV during flight. The "YOLO" has been proven to be more efficient compared to the traditionally employed machine learning algorithms [9], while it was comparable in terms of detection and classification accuracy (84%), making it the ideal candidate for UAV autonomous flight applications. To put that into perspective, "YOLO" is capable of running networks on video feeds at 150 frames per second. This means that it can process video feeds from a UAV image acquisition system with less than 25 milliseconds of latency. This nearly-instantaneous response time allows UAVs to perform time-sensitive and complex tasks in an efficient and accurate manner.

Potential Applications in the Transportation and Civil Engineering Field

It is recommended that future work focuses on testing the approach on various images with a combination of different object classes considering the fact that advancements will have major beneficial impacts on how UAVs implement complex tasks. Specifically, the focus will be on construction site management, such as road and bridge construction, where site features can be tracked and recorded with minimal human intervention. Considering that 3D object reconstruction of construction sites is gaining ground in the construction industry, the ability for the CNN to recognize 3D image reconstructed objects must be assessed. Additionally, UAVs and CNNs could be used to improve the performance of existing vehicle counting and classification tasks in traffic management with minimum interference. Another application area in the transportation field will also be in the automated identification of roadway features such as lane departure features, traffic and road signals, railway crossings, etc. These applications could greatly transform transportation asset management in the near future.

Author Contributions: Matija Radovic, Offei Adarkwa and Qiaosong Wang conceived and designed the experiments; Matija Radovic performed the experiments; Qiaosong Wang analyzed the data; Matija Radovic, Offei Adarkwa and Qiaosong Wang contributed materials and analysis tools; Matija Radovc and Offei Adarkwa wrote the paper. Qiaosong Wang reviewed the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Barrientos, A.; Colorado, J.; Cerro, J.; Martinez, A.; Rossi, C.; Sanz, D.; Valente, J. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *J. Field Robot.* **2011**, *28*, 667–689. [CrossRef]
2. Andriluka, M.; Schnitzspan, P.; Meyer, J.; Kohlbrecher, S.; Petersen, K.; Von Stryk, O.; Roth, S.; Schiele, B. Vision based victim detection from unmanned aerial vehicles. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 1740–1747.
3. Huerzeler, C.; Naldi, R.; Lippiello, V.; Carloni, R.; Nikolic, J.; Alexis, K.; Siegwart, R. AI Robots: Innovative aerial service robots for remote inspection by contact. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; p. 2080.
4. Ortiz, A.; Bonnin-Pascual, F.; Garcia-Fidalgo, E. Vessel Inspection: A Micro-Aerial Vehicle-based Approach. *J. Intell. Robot. Syst.* **2014**, *76*, 151–167. [CrossRef]
5. Snavely, N.; Seitz, S.M.; Szeliski, R. Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph. (TOG)* **2006**, *25*, 835–846. [CrossRef]
6. Sherrah, J. Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery. Available online: https://arxiv.org/pdf/1606.02585.pdf (accessed on 8 June 2017).
7. Qu, T.; Zhang, Q.; Sun, S. Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. *Multimed. Tools Appl.* **2016**, 1–13. [CrossRef]
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
9. Wang, Q.; Rasmussen, C.; Song, C. Fast, Deep Detection and Tracking of Birds and Nests. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 12–14 December 2016; pp. 146–155.
10. Howard, A.G. Some Improvements on Deep Convolutional Neural Network Based Image Classification. Available online: https://arxiv.org/ftp/arxiv/papers/1312/1312.5402.pdf (accessed on 8 June 2017).
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. Available online: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (accessed on 8 June 2017).
12. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. Available online: https://arxiv.org/pdf/1312.6229.pdf (accessed on 14 June 2017).
13. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. Decaf: A Deep Convolutional Activation Feature for Generic Visual Recognition. Available online: http://proceedings.mlr.press/v32/donahue14.pdf (accessed on 8 June 2017).
14. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. Available online: http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf (accessed on 8 June 2017).
15. Visual Object Classes Challenge 2012 (VOC2012). Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/ (accessed on 8 June 2017).
16. Visual Object Classes Challenge 2007 (VOC2007). Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/ (accessed on 8 June 2017).
17. BB Boxing Labeling Tool. 2015. Available online: https://github.com/puzzledqs/BBox-Label-Tool (accessed on 12 April 2015).
18. Civil Data Analytics. Autonomous Object Recognition Software for Drones & Vehicles. Available online: http://www.civildatanalytics.com/uav-technology.html (accessed 2 February 2016).