RAMAIAH UNIVERSITY OF APPLIED SCIENCES

ASSIGNMENT

Course Code	19CSC312A
Course Name	Artificial Intelligence
Programme	B.Tech.
Department	Computer Science and Engineering
Faculty	FET

Name of the Student	Subhendu Maji
Reg. No	18ETCS002121
Semester/Year	6^{TH} / 2018
Course Leader/s	

Declaration Sheet				
Student Name	Subhendu Maji			
Reg. No	18ETCS002121			
Programme	B.Tech.	Semester/Year	6th /2018	
Course Code	19CSC312A			
Course Title	Artificial Intelligence			
Course Date		to		
Course Leader				

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

Declaration Sheet ii
Contents iii
Faculty of Engineering and Technology iv
Question No. 1 5
1.1 Executive Summary 5
1.2 Background and Objectives 7
1.3 Comparative analysis of state-of-the-art methods 8
1.4 Conclusion and Recommendations 17

Faculty of Engineering and Technology			
Ramaiah University of Applied Sciences			
Department	Computer Science and Engineering	Programme	B. Tech.
Semester/Batch	6th/2018		
Course Code	19CSC312A	Course Title	Artificial Intelligence
Course Leader(s)	Dr. Subarna Chatterjee, Gp. Capt. Rath and Santoshi Kumari		

Assignment-1						
Register No		18ETCS002121		Name of Student	Subhendu Maji	
Question	Marking Scheme				Marks	
					Max Marks	First Examiner Marks	Moderator
1							
	1	Executive summary			**03**		
	2	Background and Objectives			**04**		
	3	Comparative analysis of state-of-the-art methods			**10**		
	4	Conclusion and Recommendations			**03**		
	5	Presentation			**05**		
	Max Marks				**25**		
Total Assignment Marks					25		

Course Marks Tabulation				
Question	**First Examiner**	**Remarks**	**Moderator**	**Remarks**
1				
Marks (Max 25)				

Signature of First Examiner **Signature of Moderator**

Question No. 1

Solution to Question No. 1:

Artificial intelligence (AI) is a rapidly growing field of technology that is capturing the attention of commercial investors, defense intellectuals, policymakers, and international competitors alike, as evidenced by a number of recent initiatives.

1.1 Executive Summary

Artificial intelligence has penetrated almost all civilian industries that one can think of. It has transformed the way individuals and businesses work, and now it is quickly making its way in becoming a critical part of modern warfare.

The strength of its army is one of the factors indicating how powerful the country is. In some of the most developed nations, investment in this sector is the highest as compared to other sectors. A major part of this investment goes towards rigorous research and development in modern technology such as AI in military applications. AI-equipped military systems are capable of handling volumes of data efficiently. In addition to that, such systems have improved self-control, self-regulation, and self-actuation due to its superior computing and decision-making capabilities.

Use Cases of AI in Military

Training: Training and simulation are multidisciplinary fields which utilize system and software engineering principles to construct models that can help soldiers train on various combat systems deployed during the actual military operations. The US Navy and Army have already initiated several sensor simulations programs.

Further, augmented and virtual reality techniques can be used to create effective, realistic, and dynamic simulations for training purposes. The reinforcement techniques enhance combat training for both virtual agents and human soldiers.

Arms and Ammunition: New-age weaponry now comes with AI-embedded technology. For example, sophisticated missiles have the capability to determine and analyses the target range for kill zones without any human intervention.

Cybersecurity: In defence circles, cyberspace is now being considered as the third war-front after land, sea, and air. A compromised and malicious network can severely compromise the security of the whole region. Defence establishments are using machine learning to predict and protect from unauthorized intrusions. This intrusion detection is usually done by classifying the network as normal or intrusive. AI-based techniques help in increasing the accuracy of such classification.

Logistics: One of the most important factors that play a role in determining the success of a military operation is logistics. Integration of machine learning and geospatial analysis with the military's logistical systems reduce the amount of effort, time, and error.

Surveillance: AI with geospatial analysis can help in the extraction of valuable intelligence from connected pieces of equipment such as radars and automatic identification systems. This information can help in detection of any illegal or suspicious activities and alert the concerned authority. Robots with AI and computer vision with IoT can also help in target identification and classification.

Military face new and escalating challenges relating to the extraction of intelligence from the volume of data collected by multi-spectral, overhead sensors.

Automated target recognition (ATR)—also referred to as object detection—will play a central role in automating and augmenting tasks such as broad area search, persistent site monitoring, discovery and tracking of construction sites, land management analysis, humanitarian aid and disaster relief, and monitoring oil and gas fracking wells (Figure 1). The output of such object detection algorithms—the target class and bounding box coordinates—are then fed into numerous downstream applications and analytics.

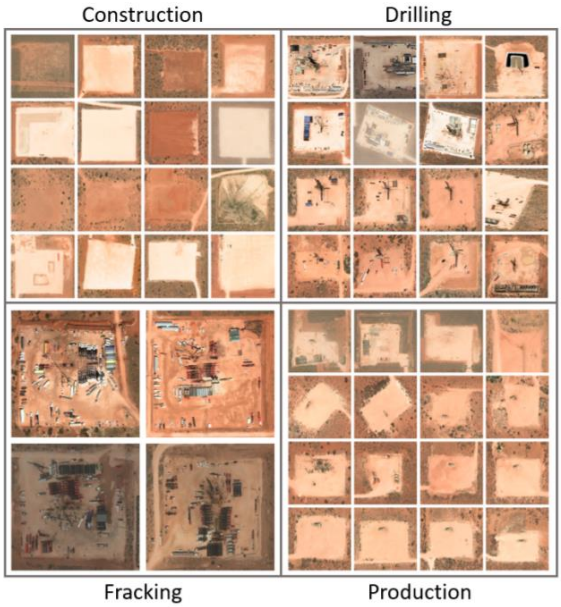

Figure 1 Examples of fracking wells in various phases in New Mexico, USA.

Deep learning models, more specifically convolutional neural networks (CNN), are increasingly being used as the core enabling technology for detecting and classifying objects within electro-optical (EO), and synthetic aperture radar (SAR) imagery. Due to the simplicity of its implementation, the sliding window algorithm is often used in place of more modern approaches for object detection found in the literature. However, sliding window suffers from a fundamental trade-off between speed and localization accuracy.

The highest accuracy models have almost always employed a two-stage approach to object detection, which can be broken down into the detection of candidate objects (Stage 1) followed by object classification and location refinement using a CNN (Stage 2).

Single-stage models perform object detection in a single shot, accomplishing this by performing regression on the bounding box parameters along with classification simultaneously using a single

network. Lastly, multi-stage models typically package several techniques together—whether it be of similar or disparate types—to accomplish object detection.

Challenges faced by Military with AI

One of the major concerns with using AI-based technology is an investment in terms of both money and skills. Especially for middle-income countries like India, where a lot of the population is still living under the poverty line, how much can we afford to provide infrastructure capabilities for building such technology is a big question. The possible solution is that policymakers must decide on what AI programme is necessary for the security of the nation and work towards them.

The use of AI in defence also presents an ethical dilemma. Experts and organizations around the world have raised such technology unintentionally escalating the tensions between countries. One of the arguments is that if an AI system fails to perform as intended may result in catastrophic implications. In fact, several human and civil rights groups call for an absolute ban on autonomous devices in defence, especially weaponry.

1.2 Background and Objectives

The accurate and timely detection of objects within imagery produced by Earth observing satellites is a key technology leveraged by many sub-disciplines within the field of Earth science. As the volume of data increases rapidly, so too will the need for automated detection algorithms to continue to improve in speed and accuracy. The necessity for real time object detection and tracking within frames of high-resolution video streams originates from the requirement that autonomous vehicles be able to swiftly respond—on the order of milliseconds—to avoid deadly collision. Out of this need arose an incredibly active area of research and development—in both algorithms and hardware—primarily leveraging progress in deep learning to advance the field of ground-based machine perception. Rather than applying these techniques to individual frames of high-resolution video data, we treat each chip within a much larger satellite image as independent input data.

we compare the detection accuracy and speed measurements of several state-of-the-art models—RetinaNet, GHM, Faster R-CNN, Grid R-CNN, Double-Head R-CNN, and Cascade R-CNN —for the task of object detection in commercial EO satellite imagery. To fully explore the solution space, we use ResNet-50, ResNet101, and ResNeXt-101 CNN options as each model's backbone—to serve as the primary feature extraction mechanism. We then compare our timing results to the sliding window algorithm for baseline comparison.

Datasets

The objective is to explore the detection accuracy and inference speed of a number of modern object detection models, used primarily by the autonomous driving industry, and how they perform when reapplied to the task of detecting objects in overhead satellite imagery. We select two object types—oil and gas fracking wells and small cars—to represent detection on various physical scales.

i. WorldView-3: Oil and Gas Fracking Wells

Hydraulic fracturing, or "fracking", is one of several types of unconventional oil and gas extraction techniques. This process involves drilling into shale formations (a type of sedimentary rock) followed by hydraulically fracturing the rock in order to liberate the oil and gas trapped within. Large well pads (50m - 250m) composed of crushed rock are constructed in order to support the heavy equipment needed in the drilling, fracking, and pumping phases of the well, and this is what our algorithms aim to detect.

ii. xView: Small Cars

The second object we focus on is the small car category from the xView dataset. Within the dataset, there are 210,938 examples of small cars—roughly 35 percent of all images contain at least one car—with contextual environments ranging from dense urban scenes to sparsely populated country roads. At the native spatial resolution of the imagery (30cm GSD), small cars may only span perhaps 7 pixels on a side (median: 14 pixels on a side), and thus pose a real challenge for most out-of-the box algorithms due to size and crowding.

Figure 2 Example prediction output from Grid R-CNN (with ResNet-101 backbone) on an xView validation image containing 3929 small cars. Yellow boxes are ground truth bounding boxes and red are predicted bounding boxes.

1.3 Comparative analysis of state-of-the-art methods

To ensure we properly benchmark each model's performance, we utilize a library called MMDetection. MMDetection is an open-source object detection toolbox (created using the PyTorch framework) which provides pretrained weights and model definitions for over 200 high performing object detection models, along with tools to train and test them.

Though there are a numerous architectures and techniques, MMDetection standardizes deep learning object detection models by expressing single-stage, two-stage, and multi-stage detection models into common functional pieces.
For instance, all models share a common component called a **backbone**, which is typically a deep convolutional neural network (CNN) used to extract features from the input image to be used in downstream bounding box regression and classification components.

We select two single-stage detectors (RetinaNet and GHM), three two-stage detectors (Faster R-CNN, Grid R-CNN, and Double-Head R-CNN), and a single multi-stage detector (Cascade R-CNN) for training and testing on our objects of interest. Though each model was created to solve a particular problem associated with object detection—whether it be speed, classification, or localization accuracy.

State-of-the-Art Models

RetinaNet: Until the introduction of RetinaNet, single-stage object detection models historically lagged in performance compared to more complex architectures. This model combined several ideas from previous works, namely the concept of anchor boxes and the use of feature pyramids, with a novel loss function referred to as Focal Loss.

Focal Loss was designed to address the severe foreground background class imbalance which was discovered to be the cause of single-stage detector performance degradation. By modifying the cross-entropy loss function to focus on the more difficult training examples, RetinaNet achieved state-of-the-art performance.

GHM: With the notion that several training imbalances exist, for example between easy and hard examples, and between positive and negative examples, a gradient harmonizing mechanism (GHM) was explored which modifies the standard classification and regression loss functions, GHM-C and GHM-R, respectively. Coupled with model architectures like RetinaNet, it was shown to achieve state-of-the-art performance.

Faster R-CNN: Faster R-CNN is the latest in a series of two-stage detection algorithms which originally stems from the combination of region proposals with convolutional neural network feature extraction, R-CNN. Faster R-CNN improves on its predecessor, Fast R-CNN, by using a second neural network for region proposal, rather than using selective search.

Grid R-CNN: This two-stage model approach to object detection differs from others in one key area—the bounding box localization technique. While most models improve object detection in one way or another, most perform bounding box localization in a similar way.

Rather than formulating localization as a regression problem performed by a set of fully connected layers which take in high-level feature maps to predict candidate boxes, Grid R-CNN replaces this approach by utilizing a grid point guided localization mechanism, and demonstrates state-of-the-art performance on the MS COCO benchmark, especially when tightening up the object localization constraint.
(Microsoft Common Objects in Context (COCO) is a large-scale object detection, segmentation, and captioning dataset used for benchmarking algorithms.)

Double-Head R-CNN: Models based on R-CNN tend to apply a single head to extract regions of interest (RoI) features for both classification and localization tasks. Instead, the Double-Head R-CNN uses a fully-connected head for classification while using a convolutional head for bounding box regression. With this modification, they were able to achieve gains over previous architectures on the MS COCO dataset.

Cascade R-CNN: Cascade R-CNN stands as the only multi-stage detection model we explore in this work. This model builds off of R-CNN by daisy-chaining a series of detectors together—each stricter about object localization than the last—in order to be sequentially more selective against close false positives. The detectors are trained together stage by stage, with the notion that the output of one detector is a good distribution for training the subsequent detector in the series

Sliding Window Model: We also benchmark the Sliding Window detector as a basis for comparison against the models selected above. Sliding window represents perhaps the simplest and most intuitive object detection system one might implement, leveraging a single CNN model which performs classification on a predefined window within the image data array. This window is then slid across the input image in both dimensions to achieve global detection coverage. Figure 3 exhibits several steps of this technique.

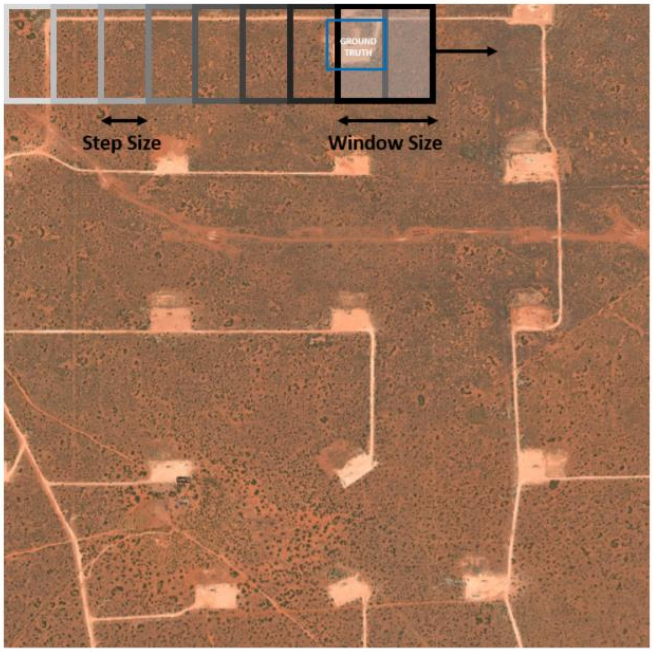

Figure 3 Schematic of the sliding window object detection technique. Blue box represents a notional ground truth bounding box.

Training Details

The MM-Detection library provides default training configuration files and pre-trained ImageNet weights for each model architecture, which we built off of for each CNN backbone chosen. Generally, most hyperparameter settings (except for learning rate) were left in their default setting.
A Stochastic Gradient Descent (SGD) optimizer with momentum, learning rate decay, and a learning rate schedule was used in the training of all models.

Sliding Window CNN backbones (ResNet-50 and ResNet101) are trained as standalone classifiers using target image chips rather than full scenes.

Evaluation Metrics

We choose the **average precision** (AP), **max** F1 score, and **area coverage rate** metrics to score our object detection models.
The prediction outputs of each model are the bounding box locations and associated class probabilities for each input image. Consequently, a decision must be made in order to concretely define what constitutes a detection in terms of probability and box localization accuracy. These parameters are the **probability score threshold** and **Jaccard Index** (or intersection over union, IoU). Given two bounding boxes, $B1$ and $B2$, the IoU is defined as

$$IoU = \frac{|B1 \cap B2|}{|B1| + |B2| - |B1 \cap B2|}$$

With some particular choice of probability score and IoU thresholds, the precision and recall are defined as follows

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively.

The average precision is defined as the area under the precision-recall curve—formed by plotting precision vs recall as one sweeps over all possible probability scores.

$$AP = \int_0^1 P(R)dR$$

In the case of multi-class object detection performance, the **mean average precision** (mAP) is typically used, which is simply the average of each class-specific AP score. The IoU which we choose to report the AP with is 0.50 $(AP@IoU = 0.50)$ for fracking wells, and 0.25 $(AP@IoU = 0.25)$ for cars.

The max F1 score is the point on the precision-recall curve with the largest harmonic mean of the precision and recall. This roughly corresponds to being the point closest to the perfect classifier (P=1, R=1).

$$F_1 = \frac{2PR}{P + R}$$

Finally, the area coverage rate measures the amount of data each model can process, and is measured in square kilometers per second.

Results

Timing results are collected using a single NVIDIA TITAN Xp GPU. The area coverage rate, measured in $km^2/sec.$, is obtained by measuring the run-time speed of the models in frames per second (FPS) on non-overlapping images of a fixed area. Regardless of the backbone network used (which dictates the amount of down-sampling performed) each WorldView3 image used for the oil and gas fracking wells were each 2.4025 km^2. Each image used for the xView small cars was 0.0324 km^2 . It should be noted that timing results for the sliding window detector are obtained with overlapping windows into the image data, which is precisely how sliding window achieves its object localization (and primarily why it is slower).

Oil and Gas Fracking Wells

Overall, and somewhat surprisingly, the single-stage detectors RetinaNet and GHM provide the highest overall detection performance, while also providing the best area coverage rates. If highest possible performance is desired, RetinaNet with a ResNeXt-101 backbone is the top choice with an average precision of 0.92. If one optimizes strictly for area coverage rate, then RetinaNet with a significantly smaller backbone (ResNet50) provides the best value.

Figure 5 shows precision-recall curves for each model using ResNet-50, ResNet-101, and ResNeXt-101 feature extractors, trained for detecting oil and gas fracking wells.

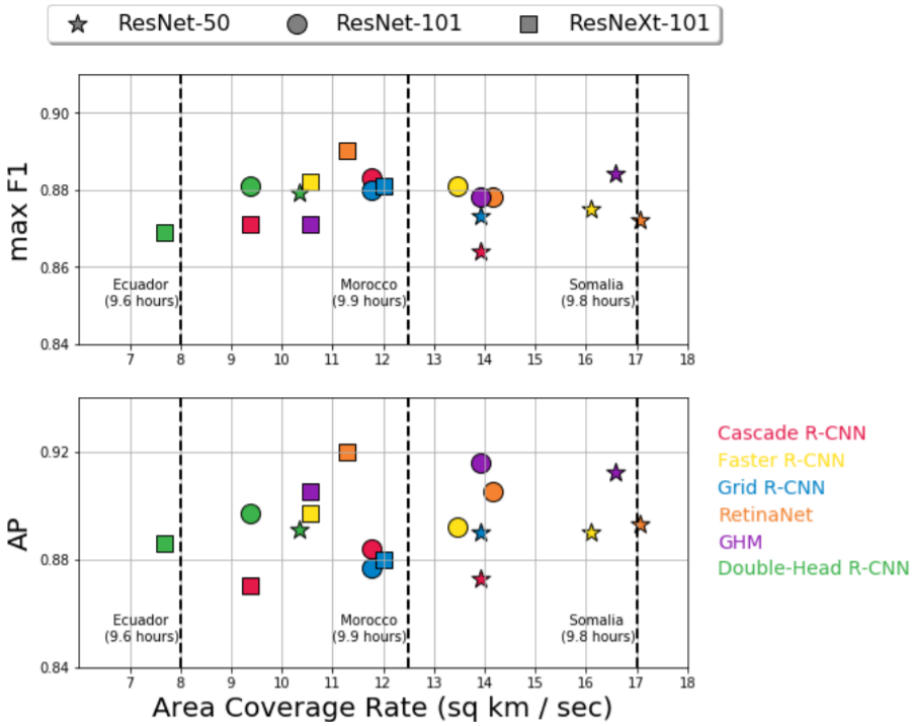

Figure 2 Oil and Gas Fracking Wells: Speed and performance results for oil

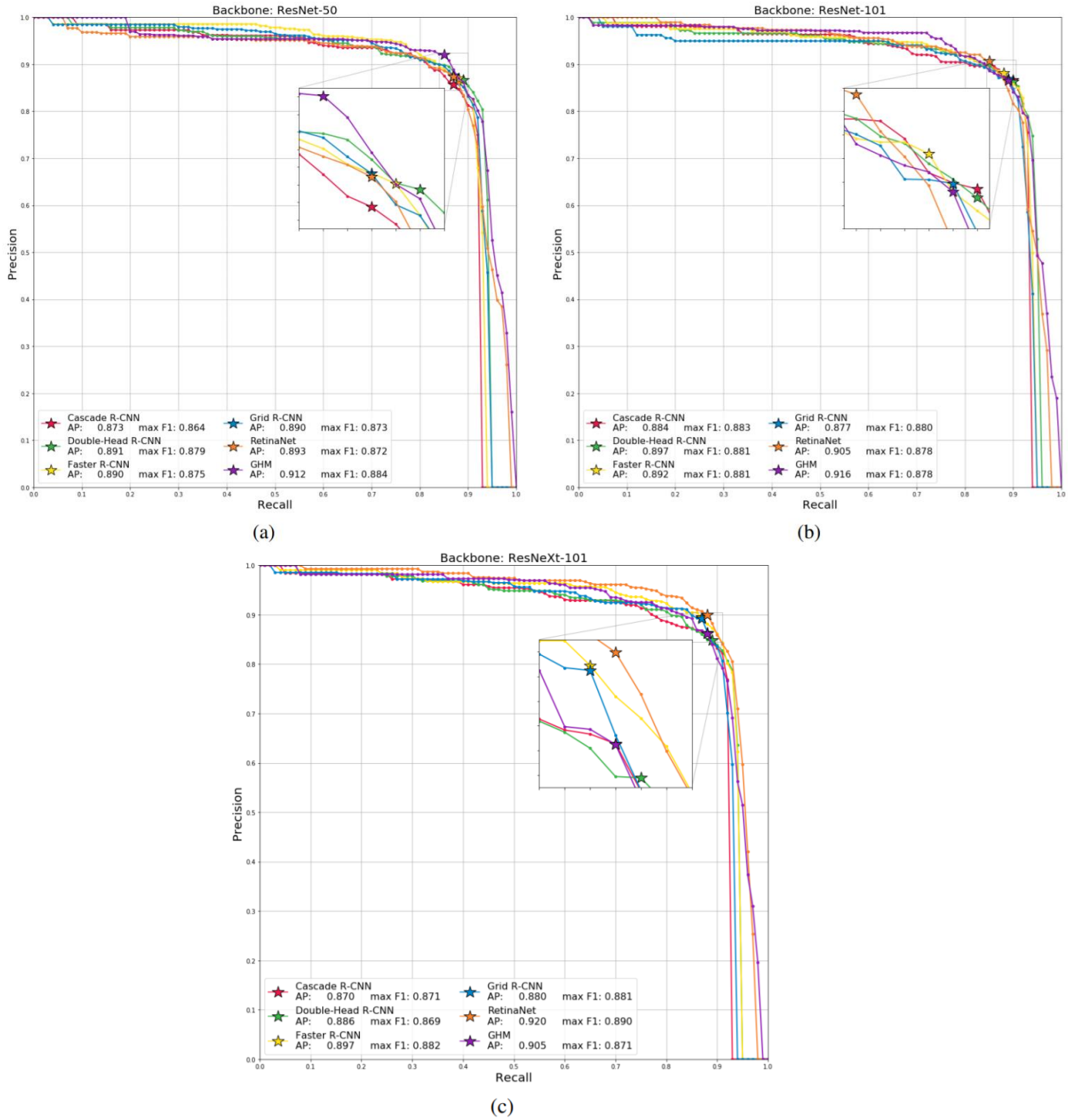

Figure 3 Oil and Gas Fracking Wells: Precision-Recall (PR) curves for all object detection models for the following CNN backbones: (a) ResNet-50, (b) ResNet-101, (c) ResNeXt-101. Stars indicate the max F1 score, while average precision (AP) scores reported represent the area under the curve.

Model Architecture	CNN Backbone	max F_1	AP	FPS	Cov. Rate (sq km/sec.)
RetinaNet	ResNet-50	0.872	0.893	**7.1**	**17.06**
	ResNet-101	0.878	0.905	5.9	14.17
	ResNeXt-101	**0.890**	**0.920**	4.7	11.29
GHM	ResNet-50	**0.884**	0.912	**6.9**	**16.58**
	ResNet-101	0.878	**0.916**	5.8	13.93
	ResNeXt-101	0.871	0.905	4.4	10.57
Faster R-CNN	ResNet-50	0.875	0.890	**6.7**	**16.10**
	ResNet-101	0.881	0.892	5.6	13.45
	ResNeXt-101	**0.882**	**0.897**	4.4	10.57
Grid R-CNN	ResNet-50	0.873	**0.890**	**5.8**	**13.93**
	ResNet-101	0.880	0.877	4.9	11.77
	ResNeXt-101	**0.881**	0.880	5.0	12.01
Cascade R-CNN	ResNet-50	0.869	0.870	**5.8**	**13.93**
	ResNet-101	0.864	0.873	5.0	12.01
	ResNeXt-101	**0.883**	**0.884**	4.0	9.61
Double-Head R-CNN	ResNet-50	0.879	0.891	**4.3**	**10.33**
	ResNet-101	**0.881**	**0.897**	3.9	9.37
	ResNeXt-101	0.869	0.886	3.2	7.69

Figure 4 Oil and Gas Fracking Wells: PERFORMANCE AND TIMING RESULTS.

Small Cars

For the detection of cars, two-stage and multi-stage detectors all achieve average precisions of just under 0.70 and max F_1 scores of around 0.75, while single-stage detectors lag in performance—varying in average precision from 0.661 (RetinaNet) down to 0.583 (GHM). The highest performing model is Grid R-CNN with ResNet-101 and ResNeXt-101 backbones, though several model and backbone combinations are very comparable. The speediest models are RetinaNet, GHM, and Faster R-CNN with ResNet-50 backbones.

However, the best combination of speed and accuracy is Faster R-CNN with a ResNet-50 backbone.

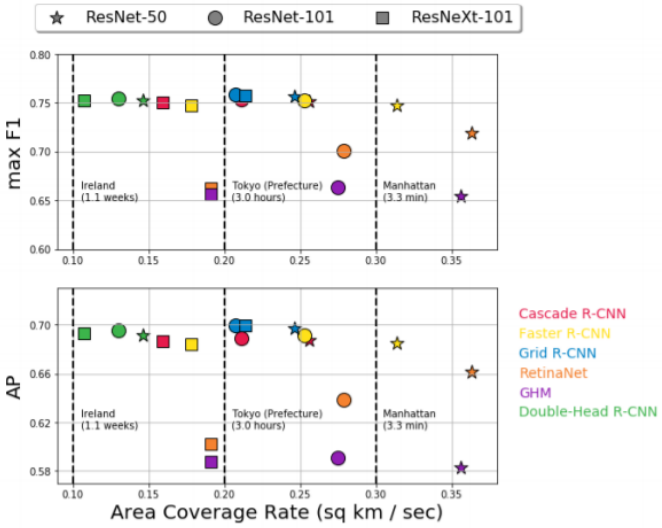

Figure 5 Small Cars: Speed and performance results for small cars.

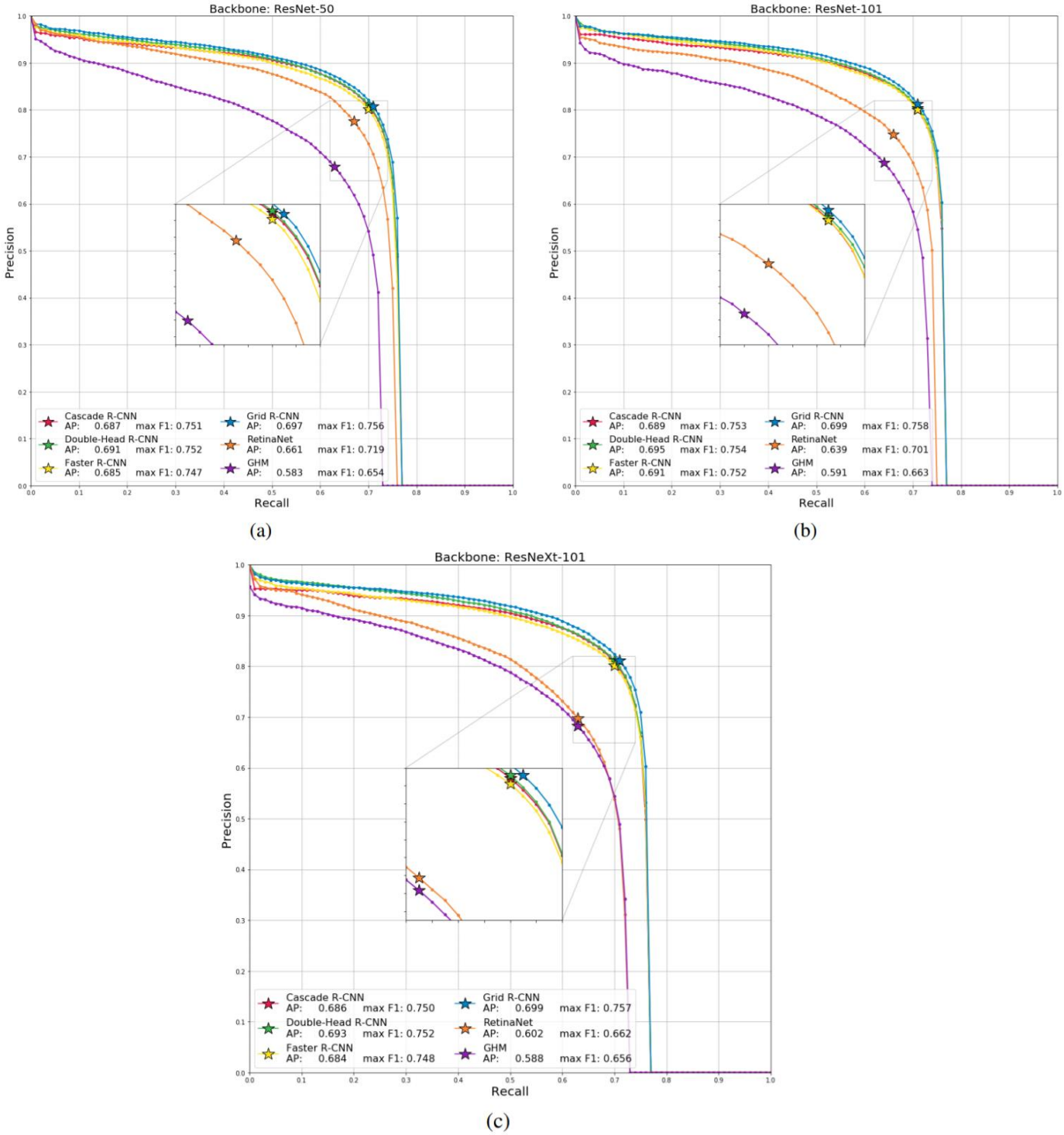

Figure 6 Small Cars: Precision-Recall (PR) curves for all object detection models for the following CNN backbones: (a) ResNet-50, (b) ResNet-101, (c) ResNeXt-101. Stars indicate the max F1 score, while average precision (AP) scores reported represent the area under the curve.

Model Architecture	CNN Backbone	max F_1	AP	FPS	Cov. Rate (sq km/sec.)
RetinaNet	ResNet-50	**0.719**	**0.661**	**11.2**	**0.363**
	ResNet-101	0.701	0.639	8.6	0.279
	ResNeXt-101	0.662	0.602	5.9	0.191
GHM	ResNet-50	0.654	0.583	**11.0**	**0.356**
	ResNet-101	**0.663**	**0.591**	8.5	0.275
	ResNeXt-101	0.656	0.588	5.9	0.191
Faster R-CNN	ResNet-50	0.747	0.685	**9.7**	**0.314**
	ResNet-101	**0.752**	**0.691**	7.8	0.253
	ResNeXt-101	0.747	0.684	5.5	0.178
Grid R-CNN	ResNet-50	0.756	0.697	**7.6**	**0.246**
	ResNet-101	**0.758**	**0.699**	6.4	0.207
	ResNeXt-101	0.757	**0.699**	6.6	0.214
Cascade R-CNN	ResNet-50	0.751	0.687	**7.9**	**0.256**
	ResNet-101	**0.753**	**0.689**	6.5	0.211
	ResNeXt-101	0.750	0.686	4.9	0.159
Double-Head R-CNN	ResNet-50	0.752	0.691	**4.5**	**0.146**
	ResNet-101	**0.754**	**0.695**	4.0	0.130
	ResNeXt-101	0.752	0.693	3.3	0.107

Figure 7 Small Cars: PERFORMANCE AND TIMING RESULTS.

Sliding Window

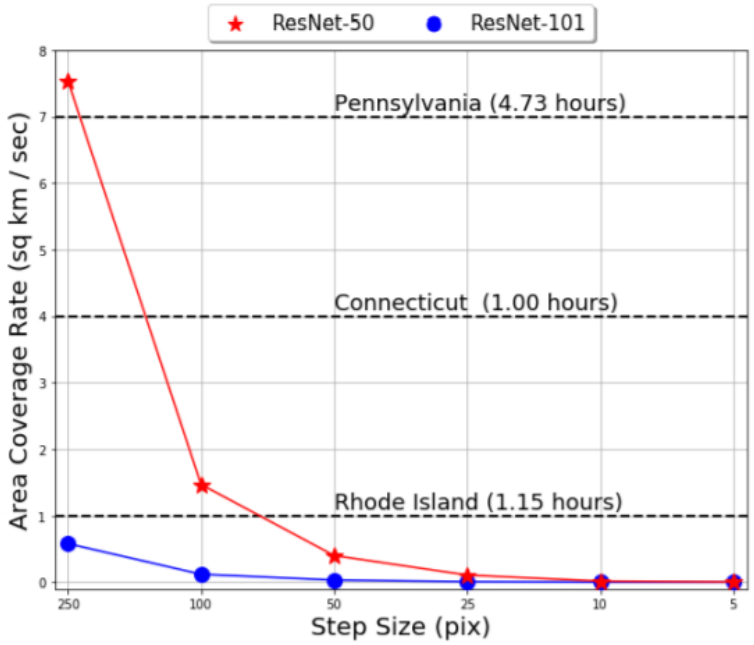

Figure 8 Sliding window area coverage rates against step size for a window

Figure 10 highlights the trade-off between run-time and step size (and by proxy bounding box localization accuracy) one must make with the sliding window algorithm using ResNet50 and ResNet-101 backbones.

1.4 Conclusion and Recommendations

we have studied in which speed and performance of modern object detection algorithms were measured for the automatic detection of oil and gas fracking wells and small cars in commercial EO satellite imagery datasets.

For the detection of wells, nearly all models perform satisfactorily, achieving both high marks in max F_1 and average precision scores, though single-stage models seem to outperform their two-stage and multi-stage counterparts in both performance and speed. However, two-stage and multistage models significantly outperform single-stage models for the detection of small cars. A possible explanation for sub-par single-stage performance could be due to the difference in the amount of training data used for each object. The data support for xView small cars is larger by two orders of magnitude than the manually curated fracking well dataset.

In general, we find that the choice of CNN backbone used within each model architecture has a weak impact on performance metrics for detecting targets like fracking wells and small cars. It is likely the case that even the least complex backbone (ResNet-50) is still capable of effectively extracting the correct features to pass along to the model's head component, especially since we only consider models which aim to detect one object at a time. For models looking to detect 60 classes or more simultaneously, the backbone choice likely has more of a direct impact on performance. The decision about which backbone to use, however, does have a large impact on inference speed.

1. M. Pritt and G. Chern, “Satellite image classification with deep learning,” in 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Oct 2017, pp. 1–7.
2. A. V. Etten, “You only look twice: Rapid multi-scale object detection in satellite imagery,” CoRR, vol. abs/1805.09512, 2018. [Online]. Available: http://arxiv.org/abs/1805.09512
3. R. J. Soldin, “Sar target recognition with deep learning,” in 2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Oct 2018, pp. 1–8.
4. M. Pritt, “Deep learning for recognizing mobile targets in satellite imagery,” in 2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Oct 2018, pp. 1–7.
5. T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr, “Focal loss for dense object detection,” in 2017 IEEE International Conference on Computer Vision (ICCV), Oct 2017, pp. 2999–3007.
6. B. Li, Y. Liu, and X. Wang, “Gradient harmonized single-stage detector,” in AAAI, 2018.
7. S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster r-cnn: Towards realtime object detection with region proposal networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1137–1149, 2015.
8. X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, “Grid r-cnn,” in CVPR, 2018.
9. Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, “Double-head rcnn: Rethinking classification and localization for object detection,” 2019.
10. Z. Cai and N. Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 6154–6162.
11. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016. [Online].
12. S. Xie, R. B. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual ´ transformations for deep neural networks,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995, 2016.
13. 13.G. Chern, A. Groener, M. Harner, T. Kuhns, A. Lam, S. O’Neill, and M. Pritt, “Globally-scalable automated target recognition (gatr),” in 2019 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Oct 2019.
14. L. Gandossi and U. V. Estorff, “An overview of hydraulic fracturing and other formation stimulation technologies for shale gas production update 2015,” Scientific and Technical Research Reports (Report). Joint Research Centre of the European Commission; Publications Office of the European Union, 2015.
15. D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, “xview: Objects in context in overhead imagery,” ArXiv, vol. abs/1802.07856, 2018.
16. 16.K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, “Mmdetection: Open mmlab detection toolbox and benchmark,” ArXiv, vol. abs/1906.07155, 2019.