AI-based Pilgrim Detection using Convolutional Neural Networks

Marwa Ben Jabra

Adel Ammar

Anis Koubâa*

Omar Cheikhrouhou

Habib Hamam

*CISTER Research Centre

CISTER-TR-200109

2020/09/02

AI-based Pilgrim Detection using Convolutional Neural Networks

Marwa Ben Jabra, Adel Ammar, Anis Koubâa*, Omar Cheikhrouhou, Habib Hamam

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: aska@isep.ipp.pt

https://www.cister-labs.pt

Abstract

Pilgrimage represents the most important Islamicreligious gathering in the world where millions of pilgrimsvisit the holy places of Makkah and Madinah to perform theirrituals. The safety and security of pilgrims is the highest priorityfor the authorities. In Makkah, 5000 cameras are spread aroundthe holy for monitoring pilgrims, but it is almost impossibleto track all events by humans considering the huge number ofimages collected every second. To address this issue, we proposeto use artificial intelligence technique based on deep learningand convolution neural networks to detect and identify Pilgrimsand their features. For this purpose, we built a comprehensivedataset for the detection of pilgrims and their genders. Then, wedevelop two convolutional neural networks based on YOLOv3and Faster-RCNN for the detection of Pilgrims. Experimentsresults show that Faster RCNN with Inception v2 featureextractor provides the best mean average precision over allclasses of 51%.

AI-based Pilgrim Detection using Convolutional Neural Networks

Marwa Ben Jabra [1] , Adel Ammar[2], Anis Koubaa [3], Omar Cheikhrouhou[4] , Habib Hamam[5]

Abstract— **Pilgrimage represents the most important Islamic religious gathering in the world where millions of pilgrims visit the holy places of Makkah and Madinah to perform their rituals. The safety and security of pilgrims is the highest priority for the authorities. In Makkah, 5000 cameras are spread around the holy mosques for monitoring pilgrims, but it is almost impossible to track all events by humans considering the huge number of images collected every second. To address this issue, we propose to use an artificial intelligence technique based on deep learning and convolutional neural networks to detect and identify Pilgrims and their features. For this purpose, we built a comprehensive dataset for the detection of pilgrims and their genders. Then, we develop two convolutional neural networks based on YOLOv3 and Faster-RCNN for the detection of Pilgrims. Experiment results show that Faster RCNN with Inception v2 feature extractor provides the best mean average precision over all classes (51%). A video demonstration that illustrates a real-time pilgrim detection using our proposed model is available at [1].**

Index Terms— **Pilgrim Detection, Convolutional Neural Networks, Deep Learning, You Only Look Once (Yolo), Faster R-CNN.**

I. INTRODUCTION

Artificial Intelligence (AI) represents the hottest technology nowadays ever with a huge impact on the societies and services provided in different types of applications. One of the main driving factors of artificial intelligence in the last decade is the emergence of deep learning in computer vision applications and, more particularly, with convolutional neural networks (CNNs). In fact, with the emergence of AlexNet [2] in 2012, the computer vision community aggressively moved to the application of CNN for image classification, detection, recognition, and semantic segmentation. Deep learning approaches have been used in a variety of use cases namely people behavior monitoring [3], vehicles detection [4], [5], semantic segmentation of urban environments [6], self-driving vehicles [7], object detection and classification [8], [9], semantic segmentation [10], [11], [12].

In this paper, we address the problem of developing AI-based solutions for pilgrims detection and monitoring in Hajj and Umrah events in Saudi Arabia. Hajj and Umrah attract annually millions of pilgrims from all over the world. In fact, According to the Ministry of Hajj, the number of Umrah Visas issued in 2019 is around 7.5 million, and the number of pilgrims during the five days of the annual Pilgrimage reached 2.5 million. The Vision 2030 of the Kingdom of Saudi Arabia aims to reach 30 million pilgrims annually. The increasing number of pilgrims induces several challenges in terms of the security and safety of pilgrims. Although there are more than 5000 cameras spread around the holy places, it is impossible for humans to track every activity of action that would need a special intervention from security forces or from civil defense agents. There are several use cases that would need an AI-based assistive technology to monitor pilgrims, including: (1) search and find of lost people, (2) real-time discovery of people in emergency services, (3) assisting pilgrims in their rituals, and several others. To address this gap, we propose to develop AI-based monitoring techniques dedicated to pilgrims. We aim at the effective use of convolutional neural networks algorithms applied to video streams collected from CCTV cameras of any video source containing pilgrims. The ultimate goal would be to provide assistive technology to the authorities to promote the safety of pilgrims.

In this paper, the contribution is three-folded. First, we built a large dataset of pilgrim and non-pilgrim instances for different genders and in different environments. Second, we have trained two state-of-the-art CNN algorithms for the specific use case of pilgrim detection, namely YOLOv3 [13] and Faster R-CNN. YOLOv3 is a one-stage detector that is known to be the fastest detection algorithm, whereas Faster R-CNN [14] is an improvement of R-CNN [15] that represents the most efficient region-based CNN algorithm for image detection. Third, we conducted a comparative study between these two algorithms to evaluate their performance in the context of pilgrim detection.

To the best of our knowledge, this is the first paper that addresses the problem of pilgrim detection using deep learning with state-of-the-art convolutional neural networks.

The remainder of the paper is organized as follows. Section II discusses related works on deep learning for people monitoring and existing non-AI techniques for pilgrim monitoring. Section III presents a brief background on both states of the art CNN algorithms, namely YOLOv3 and Faster R-CNN. Section IV presents details on the Pilgrim dataset that we built for this study. Section V presents and discussed the main results. Section VI concludes the paper and outlines future works.

II. RELATED WORKS

Several recent works have used CNN for people's behavior monitoring, but there were applied to contexts different from

*This work is supported by the Robotics and Internet-of-Things Lab of Prince Sultan University

[1] AITU American International Thelogy University in Florida, USA./ Robotics and Internet-of-Things Unit (RIoT) Labo, Saudi Arabia

[2]Prince Sultan University, Saudi Arabia.

[3] Prince Sultan University, Saudi Arabia /Gaitech Robotics,China /CISTER, INESC-TEC, ISEP, Polytechnic Institute of Porto,Portugal "anis.koubaa@gmail.com"

[4]Taif University, Taif, Kingdom of Saudi Arabia

[5]Faculty of Engineering, University of Moncton, CANADA

pilgrim detection.

For the detection of occluded pedestrians, Zhang et al.[16] proposed a simple and compact method based on Faster R-CNN and an attention mechanism based network with self or external guidance to represent various occlusion patterns in one single model. They achieved a miss rate of 56.66% on CityPersons and 45.18% on Caltech.

Molchanov et al.[17] proposed a classification approach that combines pedestrian detection and classification task in real scenes. The approach uses a YOLO neural network to overcome the problem of the low image resolution and the high density of people in a small area.

These works present several limitations, such as (*i.*) The use of high computational complexity that can be time-consuming. To solve this problem, we use the YOLOv3, which is orders of magnitude faster. (*ii.*) The low accuracy when using the RGB dataset or when dealing with a low-resolution image and the difficulty of detecting a small pedestrian. To solve this problem of detection, we used Faster R-CNN, with two different features extractors (Inception-v2 and ResNet50) that give us the best feature map that helps us to do the detection task.

On the other hand, several techniques [18], [19] were applied for pilgrim detection using sensing and mobile technologies, but not using deep learning methods.

Teduh et al.[18] proposed architecture of geo-fencing emergency alerts system for Hajj pilgrims. The proposed architecture is based on mobile phones with GPS module, which is used as pilgrims' tracking devices.

Mohandes et al.[19] developed a prototype of a wireless sensor network for tracking pilgrims in the Holy areas during Hajj. They used a principle delay tolerant network. In this system, a network of fixed master units is installed in the Holy area. Besides, every pilgrim will be given a mobile sensor unit that includes a GPS unit, a Microcontroller, antennas, and a battery that aims to sends its UID number, latitude, longitude, and time.

These works that were applied for pilgrims' detection using sensing and mobile technologies also present several problems such as, (*i.*) the difficulty to receive the GPS signal in some area, which hinders the pilgrim tracking system using GPS. (*ii.*) the difficulty of applying such systems in large crowds, since they cannot easily deal with big data.

To solve these problems, we propose to use a computer vision deep learning system for pilgrim detection in real-time. Also, it can be easily integrated to monitor pilgrims using the CCTV camera infrastructure in holy mosque areas.

III. ALGORITHMS BACKGROUND

For the pilgrims' detection, we are using Faster R-CNN [14] and YOLOv3[13] algorithms. In this section, we present the different versions of these algorithms and the difference between them.

A. *Faster R-CNN*

In this section, we provide an overview of the Faster R-CNN [14] algorithm, which we used for the detection of pilgrims. It is an improved version of R-CNN [15], which has been conceived to bypass the problem of selecting a huge number of regions. This problem is inherent to the use of the conventional CNN algorithms for object detection.

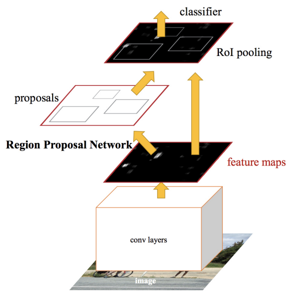

Fig. 1. Faster R-CNN

The Faster R-CNN [14] algorithm presented in figure 1 is the improved version of R-CNN. This algorithm contains two modules that share the same convolutional layers. These modules are:

- The region proposal network (RPN).
- A Fast R-CNN detector.

The RPN module is a fully convolutional network that aims to generate the region proposals, which are the bounding boxes that possibly include the candidate object, using multiple scales and object ratios. Each region proposal has an objectness score that measures the belonging of the region to the set of objects versus the background [5].

The Fast R-CNN detector is composed of the two following steps:

- The extraction of feature vectors from the region of interest (ROI) using the ROI pooling.
- The feature vector obtained is the input of the classifier composed of fully connected layers.

The classification step output is:

- A sequence of probabilities estimated of the different objects considered.
- The coordinates of the regions proposals.

B. *YOLOv3*

YOLO or You Only Look Once is an improved version of convolutional neural network CNN, which is used especially for object detection, because the CNN, as originally conceived, is very time-consuming. There are three versions of YOLO. YOLOv3 [13], which is an improved version of YOLOv2 [20] and YOLOv1 [21]. It is characterized by:

- The use of multi-label classification based on logistic regression instead of the Softmax function.
- The use of cross-entropy loss function instead of the mean square error for the classification loss.
- The prediction of different bounding boxes based on the overlapping of the bounding box anchor with the ground truth object.
- The use of the concept of Feature Pyramid Network for the prediction by predicting boxes at three different

scales and then extracting features from these scales. And the result of the prediction is a 3D tensor encoding the bounding box, the objectness score, and the prediction over classes.

- The use of Darknet-53 CNN features extractor, which is composed of 53 convolutional layers Instead of Darknet-19, using 3x3 and 1x1 filters and skip connections inspired by ResNet [22].

IV. THE PILGRIMS DATASET

In this paper, we are interested in building a comprehensive dataset for the detection of pilgrims and their genders.

A. Number of Classes

In this particular use case, we initially considered four classes: *Man/Woman* and *Pilgrim/Not-Pilgrim*. The visual appearance of the Pilgrim male person is undoubtedly clear as he wears special white two-piece clothes called *Ihram*. Thus, for any man, it is possible to visually differentiate between whether he is in a Pilgrim state (*Muhrim*) or not. However, for a woman, there is no particular visual appearance or clothes to determine whether she is in a Pilgrim state or not. As a consequence, Pilgrim and non-Pilgrim states only apply to men, not to women. As such, without loss of generality, we reduced the number of classes from four classes to three classes, namely: *Pilgrim*, *Not Pilgrim* that implicitly designate a male, and the third class is *Woman* with no additional feature. The results that we present in this paper are for three classes, although we run experiments with four classes, but were slightly less accurate than what we present in this paper.

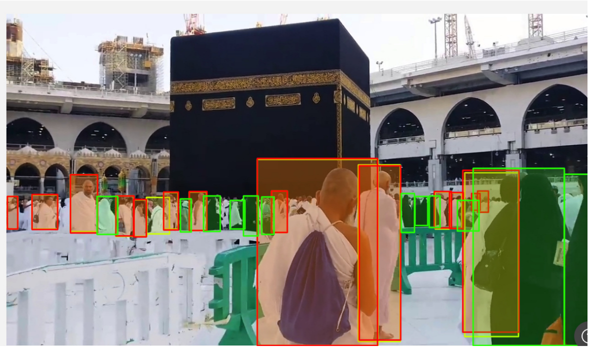

Fig. 2. Labeled image

B. Data Collection and Labeling

To create our dataset, we collected 622 images of people in the holy places of Makkah and Madinah. We chose images of people in different environments and situations, taken from various views and illuminations. Then, using the LabelImg software [23], we labeled the collected dataset into three chosen labels, namely woman, pilgrim, and not-pilgrim, as shown in Figure 2. We obtained a dataset composed of 1165 women and 2291 men instances, which are subdivided into 1339 pilgrim and 952 not-pilgrim instances. The statistics of the training and testing datasets are presented in Table I.

TABLE I
NUMBER OF IMAGES AND INSTANCES IN THE TRAINING AND TESTING DATASETS

		Training	Testing	Total
Number of images		560	62	622
Number of instances	Pilgrim men	1228	111	1339
	Non-pilgrim men	859	111	970
	Women	1016	162	1178

Our dataset follows the Pascal VOC [24] (Pascal object classes) dataset annotation scheme, and is composed of 3 classes (woman, pilgrim, not pilgrim). We choose the Pascal VOC annotation scheme because it enables evaluating our proposed YOLOv3 and Faster R-CNN pilgrim detection algorithms with significant variability in terms of object size, orientation, pose, illumination, position, and occlusion [24].

V. EXPERIMENTAL EVALUATION

In this section, we describe the results of the experimental study that we conducted to evaluate the performance of the pilgrim detection use case using two state-of-the-art algorithms, namely YOLOv3 and Faster RCNN. We start by describing the experimental setup, and we present the metrics used for the evaluation of the proposed algorithm. Finally, we analyze the results obtained for each algorithm to compare their performances. The video demonstration of a real-time pilgrim detection is available at [1].

A. Experimental Setup

In this experimental study, the training was done on two machines. The configurations of these two machines are presented in Table II.

TABLE II
CONFIGURATION TABLE

	Machine 1	**Machine 2**
CPU	Intel Core i7-8700K (3.7 GHz)	Intel Core i9-9900K (Octa-core)
Graphics card	NVIDIA GeForce 1080 (8 GB) GPU	NVIDIA GeForce RTX 2080T (11 GB) GPU
RAM	32GB	64GB
Operating system	Linux (Ubuntu 16.04 TLS)	Linux (Ubuntu 16.04 TLS)

For Faster R-CNN, we chose to test two different CNN architectures for the feature extraction, namely Inception-v2 [25] and ResNet50 [22], because these are currently among the best feature extractors for the detection task [26]. For YOLOv3, we chose to evaluate it with different resolutions, which has an impact on the accuracy and the speed of the system. We chose to use three different input sizes that have values of (320x320), (416x416), and (608x608) pixels. These settings result in five classifiers trained and tested on our pilgrim dataset. The training of these two algorithms is made to detect and recognize three classes of persons that are (`Woman`, `Pilgrim`, and `Not-Pilgrim`). To optimize these two algorithms, we used Stochastic Gradient Descent (SGD) with a default value of momentum (0.9). For the learning rate, we used an initial rate of 0.001 for YOLOv3, and for Faster R-CNN, we used an initial rate of 0.0002 with Inception-v2 and 0.0003 with ResNet50, which are the default value of each feature extractor network. We used the weight decay value of 0.0005.

B. Performance evaluation and metrics

For the evaluation of our proposed algorithms, we have used six metrics based on the following parameters:

- **True Positive (TP)**: it is the number of instances (woman, pilgrim, and not-pilgrim) successfully detected and classified.
- **False Positive (FP)**: it refers to the number of instances that are wrongly classified.
- **False Negative (FN)**: it is the number of non-detected instances.

The seven metrics used for the evaluation are:

$$\textbf{Precision} = TP/(TP+FP) \tag{1}$$

$$\textbf{Recall} = TP/(TP+FN) \tag{2}$$

$$\textbf{F1score} = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{3}$$

$$\textbf{Quality} = TP/(TP+FP+FN) \tag{4}$$

- **mIoU:** mean of the Intersection over Union that measures the overlap between the predicted and the ground-truth bounding boxes.
- **mAP:** mean Average Precision. Or AP (Average Precision) when it is measured on one class. It is an approximation of the area under the precision-recall curve [5].
- **FPS:** frame per second. It measures the inference speed of the algorithm.

C. Comparison between Faster R-CNN and YOLO v3

For the evaluation of the proposed algorithms, we compared the values of the six metrics for each algorithm shown in Table III and Table IV.

1) FN, TP and FP: Figure 3 shows that when we used YOLOv3, the number of false negatives was revealed to be much higher than the number of false positives on all classes, and also much higher than the number of true positives, which indicates that most instances go undetected. And when using Faster R-CNN, the number of true positives is much higher than the number of false positives and the number of false negatives on all classes, which indicates that most instances are detected.

2) Average Precision: When analyzing the results, it appears that YOLOv3, with an input size of (608x608) gave a better mAP with a ratio of 53.98% for the Pilgrim Class and Faster R-CNN with Inception-v2 gave a better mAP with a ratio of 59.45% for Non-Pilgrim Class (Figure 4). Figure 4 also shows that Faster R-CNN with Inception-v2 gave a much better mAP over classes. This ratio of mAP is good compared to other algorithms of the related work [16].

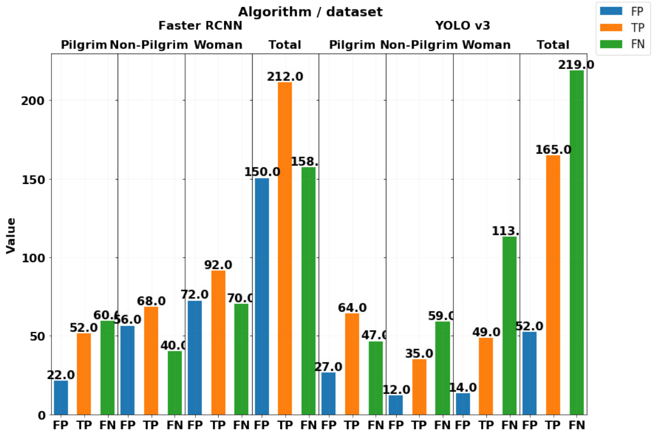

Fig. 3. Average number of false positives (FP), false negatives (FN), and true positives (TP) for YOLOv3 and Faster R-CNN.

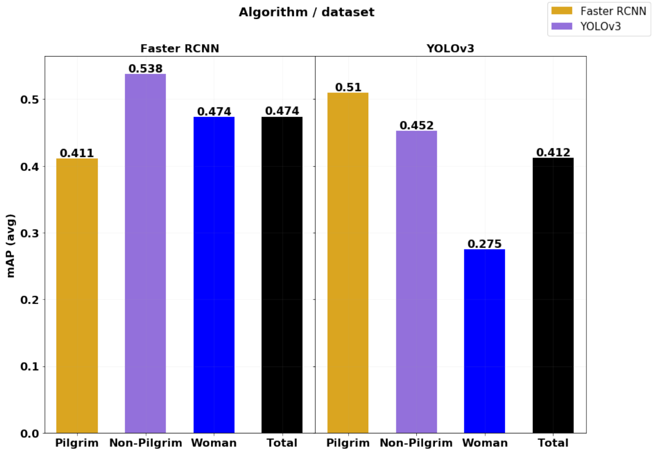

Fig. 4. Comparison of the mean AP between YOLOv3 (Input size of (608x608)) and Faster R-CNN (Inception-v2 features extractor).

3) Precision and mIoU: The results of Average IoU, show that YOLOv3 gave a better IoU over classes than Faster R-CNN. And the results of precision show that YOLOv3, with an input size of (320x320)px, gave a better precision for the Non-Pilgrim Class and Faster R-CNN with Inception-v2 gave a better precision on Pilgrim Class. It also shows that YOLOv3, with an input size of (320x320) gave a much better precision over classes with a ratio of 80.58%.

4) Recall: Analyzing the average recall results, we found that Faster R-CNN outperforms YOLOv3 in this metric with a slightly better performance with the ratio of 59.29% for Inception-v2 feature extractor over Resnet50, and a marked inferior performance for YOLOv3 with an input size of (320x320).

5) Robustness: When analyzing the quality that measures the robustness of the algorithms, we observe that YOLOv3 gave a better quality for the Non-Pilgrim Class, and Faster R-CNN gave a better Precision on Pilgrim Class. The results show that Faster R-CNN with Inception-v2 gave a much better precision over classes with a ratio of 41.72%.

The F1score that also measures the robustness based on

TABLE III

EVALUATION METRICS OF FASTER R-CNN AND YOLOV3 FOR EACH CLASS

Algorithm		**YOLOv3 (320x320)px**	**YOLOv3 (416x416)px**	**YOLOv3 (608x608)px**	**Faster R-CNN (Inception v2)**	**Faster R-CNN (ResNet 50)**
Class "Pilgrim"	FP	20	33	27	19	24
	TP	64	61	68	55	48
	FN	47	50	43	56	63
	Precision	0.7619	0.6489	0.7157	0.7432	0.6666
	Recall	0.5765	0.5495	0.6126	0.4954	0.4324
	Quality	0.4885	0.4236	0.4927	0.4230	0.3555
	F1score	0.6564	0.5951	0.6601	0.5945	0.5245
	AP	0.5098	0.4788	0.5398	0.4462	0.3751
	mIoU	0.6352	0.5988	0.6192	0.5710	0.5850
Class "Non-Pilgrim"	FP	6	16	14	71	42
	TP	50	51	55	76	61
	FN	61	60	56	35	50
	Precision	0.8928	0.7611	0.7971	0.5170	0.5922
	Recall	0.4504	0.4594	0.4954	0.6846	0.5495
	Quality	0.4273	0.4015	0.44	0.4175	0.3986
	F1score	0.5988	0.5730	0.6111	0.5891	0.5700
	AP	0.4407	0.4373	0.4786	0.5985	0.4770
	mIoU	0.6352	0.5988	0.6192	0.5710	0.5850
Class "Woman"	FP	14	17	10	74	71
	TP	45	59	42	97	86
	FN	117	103	120	65	76
	Precision	0.7627	0.7763	0.8076	0.5672	0.5477
	Recall	0.2777	0.3641	0.2592	0.5987	0.5308
	Quality	0.2556	0.3296	0.2441	0.4110	0.3690
	F1score	0.4072	0.4957	0.3925	0.5825	0.5391
	AP	0.2493	0.3295	0.2458	0.5041	0.4428
	mIoU	0.6352	0.5988	0.6192	0.5710	0.5850

TABLE IV

EVALUATION METRICS OF FASTER R-CNN AND YOLOV3 OVER CLASSES

Algorithm	YOLOv3 (320x320)px	YOLOv3 (416x416)px	YOLOv3 (608x608)px	Faster R-CNN (Inception v2)	Faster R-CNN (ResNet 50)
FP	40	66	51	164	137
TP	159	171	165	228	195
FN	225	213	219	156	189
Precision	0.8058	0.7288	0.7735	0.6091	0.6022
Recall	0.4349	0.4577	0.4557	0.5929	0.5042
Quality	0.3905	0.3849	0.3923	0.4172	0.3744
F1score	0.5541	0.5546	0.5546	0.5887	0.5446
mAP	0.3999	0.4152	0.4214	0.5162	0.4317
mIoU	0.6352	0.5988	0.6192	0.5710	0.5850
FPS	91.28	65.31	43.84	3.35	3.8

the precision and recall ratios reveals that YOLOv3, with an input size of (608x608) gave a better performance with a ratio of 66.01% for the Pilgrim Class and Faster R-CNN gave a better precision also on Pilgrim Class with a ratio of 59.45%. Over all classes, Faster R-CNN with Inception-v2 gave a much better score with a ratio of 58.87%.

6) Inference Processing time: The results of the average Inference speed measured in Frames per Second (FPS), for each of the tested algorithms, show that YOLOv3 is 12 to 27 times faster than Faster R-CNN in the inference phase (Table IV). Yolov3 has a real-time inference speed, even for the highest input size (608x608)px.

7) Effect of the feature extractor: When analyzing the effect of the feature extractor for Faster R-CNN, the results shows that Resnet50 feature extractor is slightly faster than Inception-v2 because it is less computationally complex. But, Inception-v2 outperforms Resnet50 on almost all other metrics.

8) Effect of the input size: Table IV shows a significant gain in YOLOv3's AP when moving from a (320x320) input size to (608x608), but with a substantial loss in precision. The input size has also an important impact on the inference processing speed of YOLOv3 because a larger input size generates a higher number of network parameters and operations (FPS from 44 FPS for (608x608) up to 91 FPS for (320x320)).

In this section, we compared the performance of YOLOv3 (with three different input sizes) and Faster R-CNN (with two different feature extractors) and the impact of the input size and the feature extractors. Figure 5 summarizes the main results of this comparison study. It compares the trade-off between AP and inference time for YOLOv3 (with three different input sizes) and Faster R-CNN (with two different feature extractors). It can be observed that YOLOv3 (with input size (320x320)) gave the best inference speed with low AP, contrary to Faster R-CNN (with Inceptionv2 as feature extractor) which gave the lowest inference speed with the best AP. This emphasizes that neither algorithm surpasses the other in all cases.

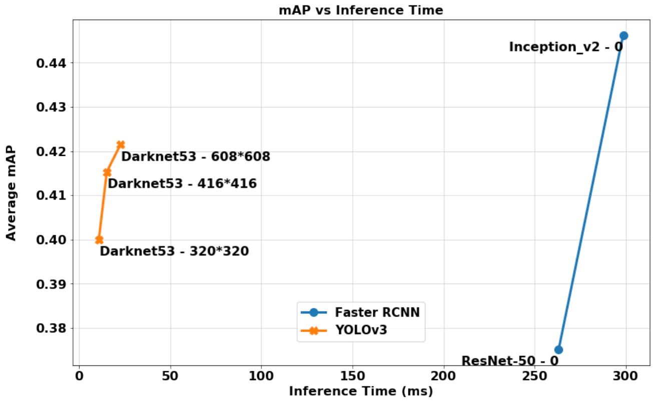

Fig. 5. Comparison of the trade-off between mAP and inference time for YOLOv3 (with 3 different imput sizes) and Faster R-CNN (with two different feature extractors),

VI. Conclusion

In this paper, we developed convolutional neural network models for pilgrim detection for Al-Hajj based on YOLOv3 and Faster RCNN. We have built a dataset containing three classes of pilgrims, non-pilgrims and women. Experimental results show that Faster RCNN with Inception v2 feature extractor provides the best mean average precision over all classes with 51%, comparable to state-of-the-art object detection algorithms. In future work, we will extend the dataset to have several tens of thousands of instances to improve the overall accuracy and precision, and we will consider more classes. We also aim at developing a search application for lost people during Hajj and Umrah.

Acknowledgment

This work is supported by the Robotics and Internet-of-Things Lab of Prince Sultan University.

References

[1] A. Koubaa, “Ai-based pilgrim detection and monitoring using deep learning, https://www.youtube.com/watch?v=L-nmYBY2pvE.”

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[3] A. Koubaa, A. Ammar, A. A.-H. Bilel Benjdira, B. Kawaf, A. B. Saleh Ali Al-Yahri, K. Assaf, and M. B. Ras, “Activity Monitoring of Islamic Prayer (Salat) Postures using Deep Learning,” *arXiv pre-print 1911.xxxxx*, November 2019.

[4] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, “Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3,” in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pp. 1–6, IEEE, 2019.

[5] A. Ammar, A. Koubaa, M. Ahmed, and A. Saad, “Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3,” *arXiv pre-print 1910.07234*, October 2019.

[6] B. Benjdira, Y. Bazi, A. Koubaa, and K. Ouni, “Unsupervised Domain Adaptation Using Generative Adversarial Networks for Semantic Segmentation of Aerial Images,” *Remote Sensing*, vol. 11, 2019.

[7] B. Schoettle and M. Sivak, “A survey of public opinion about autonomous and self-driving vehicles in the us, the uk, and australia,” tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute, 2014.

[8] I. Ševo and A. Avramović, “Convolutional Neural Network Based Automatic Object Detection on Aerial Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, pp. 740–744, May 2016.

[9] K. S. Ochoa and Z. Guo, “A Framework for the Management of Agricultural Resources with Automated Aerial Imagery Detection,” *Computers and Electronics in Agriculture*, vol. 162, pp. 53 – 69, 2019.

[10] M. Kampffmeyer, A. Salberg, and R. Jenssen, “Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 680–688, June 2016.

[11] S. M. Azimi, P. Fischer, M. Körner, and P. Reinartz, “Aerial LaneNet: Lane-Marking Semantic Segmentation in Aerial Imagery Using Wavelet-Enhanced Cost-Sensitive Symmetric Fully Convolutional Neural Networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 2920–2938, May 2019.

[12] L. Mou and X. X. Zhu, “Vehicle Instance Segmentation From Aerial Image and Video Using a Multitask Learning Residual Fully Convolutional Network,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, pp. 6699–6711, Nov 2018.

[13] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2017.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[16] S. Zhang, J. Yang, and B. Schiele, “Occluded pedestrian detection through guided attention in cnns,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6995–7003, 2018.

[17] V. Molchanov, B. Vishnyakov, Y. Vizilter, O. Vishnyakova, and V. Knyaz, “Pedestrian detection in video surveillance using fully convolutional yolo neural network,” in *Automated Visual Inspection and Machine Vision II*, vol. 10334, p. 103340Q, International Society for Optics and Photonics, 2017.

[18] T. Dirgahayu and S. Hidayat, “An architectural design of geo-fencing emergency alerts system for hajj pilgrims,” in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, pp. 1–6, IEEE, 2018.

[19] M. Mohandes, M. A. Haleem, A. Abul-Hussain, and K. Balakrishnan, “Pilgrims tracking using wireless sensor network,” in *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pp. 325–328, IEEE, 2011.

[20] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[21] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 779–788, 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Arxiv.Org*, 2015.

[23] D. Tzutalin, “Labelimg. git code (2015). https://github.com/tzutalin/labelimg.”

[24] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[26] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7310–7311, 2017.