Mobility Aids Detection using Convolution Neural Network (CNN)

Amir Mukhtar*, Michael J. Cree, Jonathan B. Scott, Lee Streeter
School of Engineering, University of Waikato, Hamilton, New Zealand
*am301@students.waikato.ac.nz, {michael.cree, jonathan.scott, lee.streeter}@waikato.ac.nz

***Abstract*—The automated detection of disabled persons in surveillance videos to gain data for lobbying access for disabled persons is a largely unexplored application. We train You Only Look Once (YOLO) CNN on a custom database and achieve an accuracy of 92% for detecting disabled pedestrians in surveillance videos. A person is declared disabled if they are detected in the close proximity of a mobility aid. The detection outcome was further categorised into five classes of mobility aids and precision was calculated.**

***Index Terms*—convolutional neural network, mobility aids, computer vision, YOLO**

I. Introduction

We investigate computer vision and machine learning techniques for identifying physically disabled people appearing in surveillance footage. This is a challenging task as surveillance video acquired outside is subject to clutter, varying illumination, changing camera view, and is often of poor quality and resolution.

We focus on exploring machine learning techniques for identifying disabled people in surveillance videos by detecting visible mobility aids. Existing computer vision datasets unfortunately lack annotated images and videos for disabled pedestrians and mobility aids thus making it difficult to train a system for the task. A great deal of research has been conducted however on vision based pedestrian detection [1], human behaviour detection [2], [3] and gait recognition [4].

In an earlier study [5] we extracted motion information crucial for differentiating a disabled person from healthy person from videos. Manual extraction of gait signals revealed that there is useful information in the gait of a walking person for detecting unusual motion patterns. However, an automated scheme (based on motion detection [6] and skeletonization [7]) failed to reproduce the same information due to problems of shadow and segmentation leading to inaccuracies in extracting the silhouette of the moving person. In this research, we now propose a different approach: detect the presence of a mobility aid used by a disabled person in addition to the person.

In this work, an image database for mobility aids is formed and YOLO architectures are trained and tested on that. We propose grouping together a person and mobility aid based on their closeness in space and consider the unit as a disabled person. Both versions of YOLO (v2 and v3) are trained and performance is compared. The system was also tested on images and video data. The rest of this paper is organised as follows, section II gives an overview of Convolution Neural Networks (CNN) followed by YOLO training methodology in Section III. Results are reported in Section IV with discussion in Section V. Finally, the conclusions drawn from this research study are presented in Section VI.

This study was funded by the project TRAD1401 of Callaghan Innovation, New Zealand.

978-1-7281-0125-5/18/$31.00

II. Convolution Neural Networks

CNN is a class of deep and feed-forward artificial neural networks [8]. They dominate image detection and classification tasks in computer vision and require relatively little preprocessing compared to other image classification algorithms. CNN learns the filters that in conventional algorithms were hand-engineered thus are independent of prior information and eliminate manual effort in feature design.

An image is input directly to the network, and this is followed by several stages of convolution, pooling and/or normalization layers [9]. Convolutional layers apply a convolution operation on the input data and pass the result to the next layer. Convolution function is specified by a vector of weights and a bias which are adjusted during the training stage depending on the loss function and learning rate. A prominent feature of CNNs is that many neurons share the same filter and thus require less memory. Pooling layers combine the outputs of neuron clusters at one layer into a single neuron in the next layer. Thereafter, representations from these operations feed one or more fully connected layers which give class label(s). Fully connected layers connect every neuron in one layer to every neuron in another layer and are same as those in multilayer perceptron neural network (MLP) [10].

Although fully connected feed-forward neural networks can be trained for learning features in image classification, their architecture is practically not suitable for images since a very high number of neurons would be required. Even a low resolution image with a shallow architecture results in large input sizes associated as each pixel corresponds to a relevant variable. The convolution operation solves this problem by reducing the number of free parameters, allowing the network to be deeper with fewer parameters [11]. For instance, regardless of image size, tiling regions of size 7×7, each with the same shared weights, requires only 49 learnable parameters. It

also resolves complexities of vanishing or exploding gradients during the training phase by using back-propagation.

You Look Only Once (YOLO) was first introduced in 2015 as a state of art real-time object detection system [12]. A single neural network predicts bounding boxes and class probabilities for detected objects in images in a single evaluation. YOLO makes more localization errors compared to state of the art detection systems based on CNNs. YOLOv3 [13], the latest version of YOLO series, has 53 convolutional layers and is a hybrid of YOLOv2, Darknet-19, and residual networks. YOLOv3 is a good detector being accurate and runs at speed faster than that of existing CNNs [13]. The new network is more powerful than Darknet19 and more efficient than ResNet-101 or ResNet-152. YOLOv3 has network structure that better utilizes the GPU, making it more efficient to evaluate and thus faster. YOLOv3 has performance comparable to state of the art CNNs and outperforms them on processing time criteria [13].

III. Methodology

A. Training Phase

We select YOLO to perform detection of mobility aids in the images. Here, we explain the training phase of YOLO in Darknet framework and refer readers to the original paper [13] for detailed technical information. Pre-trained weights available at the author's website[1] are utilized and transfer learning concept is enforced. Transfer learning changes the dimensions of output layers depending on the number of classes to be detected. Training parameters are tuned to cater the custom database and number of classes in training set. YOLO resizes images to extract features at multiple scales adding robustness to its learning.

YOLO predicts bounding boxes using dimension clusters as anchor boxes [13]. The network predicts location and dimensions for each bounding box along with an objectness score for each bounding box using logistic regression. This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior. A prediction is ignored if the bounding box prior is not the best but does overlap a ground truth object by more than some threshold.

The last layer predicts a three dimensional tensor encoding bounding box, objectness and class predictions. In our experiments with ImageNet data, the output tensor shape is $N \times N \times [3 \times (4 + 1 + 8)]$ for the four bounding box offsets, one objectness prediction, and eight class predictions. N is the grid size which refers to all possible locations for bounding box. 'Objectness' is the probability that a given bounding box encloses an object belonging to any class in the training dataset, and 'class predictions' is the classification score for a particular object inside a bounding box. The objectness prediction and class prediction scores are combined into one final score that specifies the probability that the bounding box contains a specific type of object. Both YOLOv2 and YOLOv3 are trained on the same dataset and evaluated for detecting mobility aids in test images. Their performance is also compared to look for better performing network for our particular application. The network was trained for 97 147 batches (at learning rate of 0.001) and weights were saved after every 10 000 iterations. The batch size was set to 64 images and a total of 12 434 816 images were processed during the whole training phase for 100 000 iterations. Two NVIDIA GeForce GTX 1080 Ti graphics cards did extensive number crunching and proved handy in reducing times for retraining YOLOv2 and YOLOv3 on different dataset formations.

B. Dataset

We are not aware of any publicly available databases suitable for training a CNN to recognise mobility aids, therefore, we sourced images from ImageNet, Google Images and INRIA's pedestrian database to create a custom dataset. This image dataset has a total of eight classes inclusive of five mobility aids, pedestrians and the rest (car and bicycle) act as distractors having structures similar as those in wheelchair and mobility scooters. Pre-trained weights are useful in reducing training times when there are features common between the source (database on which YOLO was originally trained) and target (our custom images) datasets. Car, bicycle and pedestrian objects featuring in both domains reduce the number of epochs required to minimize the loss function. The majority of images sourced were not labelled with ground truth and were manually annotated. Our database contains 5 819 images of which 4 653 images (6 715 training examples) were labelled manually. The labelling task was performed by two PhD students at the School of Engineering, University of Waikato. The annotated boxes were re-drawn by a computer program on the labelled images and were examined manually to ensure the correctness and consistency in labelling.

The dataset was randomly split into three sets being training, validation and test images. It was observed that YOLO does not require validation or test images at the training phase but uses the loss function based on the prediction box overlap with the ground truth, therefore, the validation part of the dataset was not utilized and the test images were used for producing results shown in Table II. Test images comprise from 9.6% to 10.5% of the total (Training+Testing) images used in this research. This breakdown is shown in Table I. Once annotated, the bounding box information (location and size) for all labelled objects in images was saved.

TABLE I
Dataset Description

Object Type	Training	Validation	Testing
Wheelchair	931	150	100
Crutch	514	150	55
Walking Frame	513	80	55
Walking Stick	573	90	65
Mobility Scooter	512	80	60
Person	663	100	75
Car	324	60	35
Bicycle	500	79	55
Total	4530	789	500

[1]https://pjreddie.com/darknet/yolo/

IV. Results

The trained network was tested on ImageNet images and videos with people using mobility aids. These surveillance videos were collected in August, 2017 for research purposes. The confusion matrix summarizing the detection result using intersection over union (IOU) has been provided in Table II. IOU is defined by,

$$\text{IOU} = \frac{A \cap B}{A \cup B}, \tag{1}$$

where A is the area of bounding box for the detected object and B is that of ground truth used to evaluate the system. Dividing the area of overlap by the area of union gives IOU score.

Several video clips were tested and accuracy of 92% was obtained from results summarised in Table III. Entries in 'Others' row refer to objects not belonging to any of the eight classes in our datasets but classified as one of mobility aids, person, car and bicycle. These objects were incorrectly classified because of their appearance similar to those in training database. Detection results from videos is shown in Fig. 1. Processing time was also recorded for a collection of short video clips and experiments revealed that it was dependent on the number of objects detected in a given frame. A list of processing speeds and times for different videos we tested is provided in Table IV. We also had few incorrect detections which are displayed in Fig. 2. These false detections were caused by the objects with structure similar to those in the training dataset. For instance, the yellow box with crutch prediction (in lower right image) contains vertical shaped structure having appearance features that are common in crutches. A few errors also resulted due to poor localization by YOLO leading to low IOU score.

YOLO v2 and v3 are trained and tested on the same datasets for performance comparison. Version 3 has 0.89 precision and 0.92 recall while those of version 2 are 0.81 and 0.86 respectively. Precision and recall values for this multi-class problem are calculated in the nomal manner [14].

V. Discussion

Our custom dataset for mobility aid detection is unique and sets a foundation for future machine learning applications designed for disabled pedestrian detection. YOLO trained on this database shows good performance in picking mobility aids from outdoor surveillance videos. An accuracy of 92% is a decent detection rate given that we are unaware of any other mobility aid detection system to compare against. Most test images/video frames had objects from multiple classes in it and YOLOv3 was able to detect most of the mobility aids along with the pedestrians using them.

An adequate number of the images is also important to yield statistically meaningful results, therefore we gathered hundreds of training images for each category to empower the system's detection performance. Numbers in person column of Table II are higher than those for other classes due to the fact that a person using a mobility aid leads to overlapping bounding boxes. This affects the false detection count for making a confusion matrix but actually useful in regarding a person disabled since he/she lie too close to a mobility aid. Cropping the training images to the mobility aid size can prevent imbalance but may lead to classification errors specially in occlusion. This is due to partial appearance of human body parts visible in training images and scanned as negative instances. Plenty of wheelchair images are available in ImageNet database and make 20.5% of the total database images thus outnumbering those of other individual classes. Class imbalance can not be ruled out for wheelchair versus the rest of the classes and we aim to investigate further in future experiments.

The confusion matrix in Table II reports incorrect detections and pedestrians account for most false detections due to the fact that part of a person always appears overlapped with mobility aid. For example, an image with a wheelchair person has two objects; wheelchair and a person. Their annotation is two overlapped ground truth boxes with human legs and torso assigned as part of wheelchair thus affecting the classifier performance. Training on mobility aid images without a person may not work since classifier is not trained on a realistic depiction of real world scenarios. Therefore, our dataset is designed to contain images of mobility aids with and without persons using them.

We believe that the actual frames per second (FPS) for the system could be higher as YOLO run-time parameters were not fully optimised. Our original work was based on retraining YOLO's version 2 for detecting mobility aids but during the course of experiments, YOLOv3 was announced so we upgraded the current state of the system to version 3. Results show that the version 3 performs slightly better than version 2.

VI. Conclusion

Automated Detection of disabled pedestrians from surveillance videos is a challenging task. In this paper, YOLOv3 is customized and retrained to identify visible mobility aids for detecting disabled people. Our self collected and annotated image dataset from ImageNet, web-search and INRIA has proved adequate for detecting mobility aids with reliable accuracy. Addition of car and bicycle classes in training set empowered the system to learn similar features among different objects thus increasing the robustness and reducing false detections. The evaluation part demonstrates that system has decent performance when tested on images and outdoor videos. The system successfully detects all five modes of mobility aids with occasional mix-up of crutch/stick and wheelchair/walking frame classes because of their similar build. This problem can be avoided by grouping similar objects (crutch and walking sticks) in the same class.

In future work, we shall continue on improving results and bring the detection result close to the human count. A human count could be set as a benchmark for calibrating detection counts resulting from the integration of YOLOv3 and object tracker. For upcoming research experiments, additional

TABLE II
CONFUSION MATRIX FOR 8 CLASS (IOU=0.5)

Actual Class \ Predicted Class	Wheelchair	Crutch	Walking Frame	Walking Stick	Mobility Scooter	Car	Person	Bicycle	Total
Object Type									
Wheelchair	112	0	0	0	0	2	18	2	134
Crutch	2	67	0	3	0	4	5	0	81
Walking Frame	0	2	57	1	0	0	2	0	62
Walking Stick	0	2	0	81	0	0	1	0	84
Mobility Scooter	0	0	0	0	63	0	1	0	64
Car	0	0	0	0	0	96	5	0	101
Person	8	6	2	2	0	3	479	2	502
Bicycle	1	0	0	1	0	1	15	79	62
Others	1	4	0	7	1	3	23	2	41
Total	124	81	59	95	64	109	549	85	1166

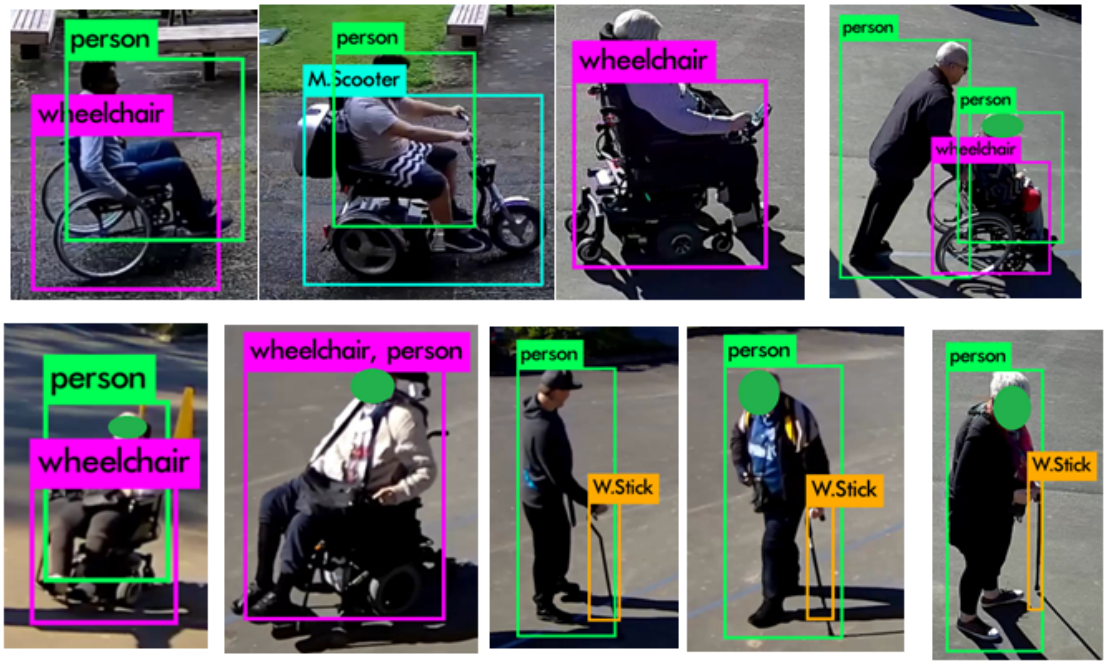

Fig. 1. Correct mobility aids detection in test videos

TABLE III
MOBILITY AIDS DETECTION IN TEST VIDEOS

Object	Actual	Detected
Wheelchair	7	7
Stick	6	5
Walking Frame	3	1
Mobility Scooter	2	2
Person	20	20

TABLE IV
CNN PROCESSING TIMES ON TEST VIDEOS

objects per frame	frames tested	processing speed (FPS)	processing time (ms)
1-2	570	22.80	43.86
2-3	900	22.50	44.44
4-5	390	19.50	51.28
6-7	2 071	15.34	65.19

surveillance videos from different parts of the Hamilton city are expected from Stantec. The dataset images collected and annotated in this study, will continue to be our CNN training source in future experiments.

ACKNOWLEDGEMENT

We are thankful to Stantec and Stroke Foundation for their support in data collection. Special thanks to Bridget Burdett (at Stantec) who assisted us to gather test data from Hamilton City Council and other places in Hamilton.

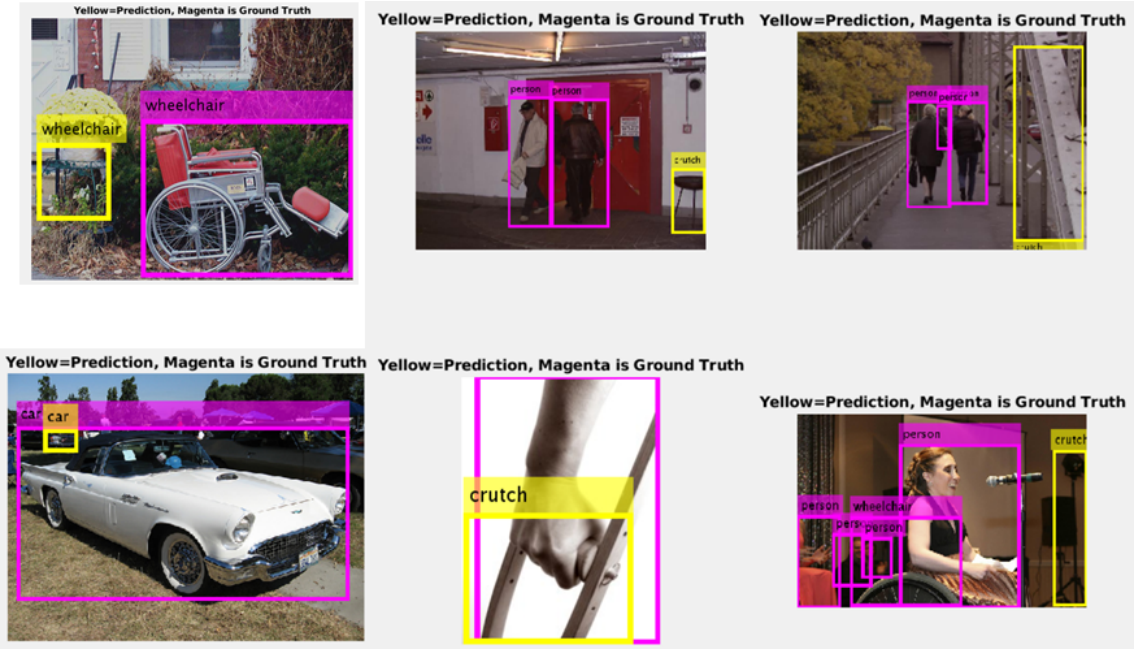

Fig. 2. Incorrect Detections

REFERENCES

[1] M. Paul, S. M. Haque, and S. Chakraborty, "Human detection in surveillance videos and its applications-a review," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 1–16, 2013.

[2] P. Afsar, P. Cortez, and H. Santos, "Automatic visual detection of human behavior: A review from 2000 to 2014," *Expert Systems with Applications*, vol. 42, no. 20, pp. 6935–6956, 2015.

[3] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognition–a review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 865–878, 2012.

[4] T. K. M. Lee, M. Belkhatir, and S. Sanei, "A comprehensive review of past and present vision-based techniques for gait recognition," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 2833–2869, 2013.

[5] A. Mukhtar, M. J. Cree, J. B. Scott, and L. Streeter, "Gait analysis of pedestrians with the aim of detecting disabled people," *Applied Mechanics and Materials*, vol. 884, pp. 105–112, 9 2018.

[6] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Ft. Collins, CO, USA, 1999.

[7] H. Fujiyoshi and A. J. Lipton, "Real-time human motion analysis by image skeletonization," in *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, 1998, p. 15.

[8] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. Cambridge, MA: MIT, 2016.

[9] I. Hadji and R. P. Wildes, "What do we understand about convolutional networks?" *arXiv preprint arXiv:1803.08834*, 2018.

[10] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[11] H. H. Aghdam and E. J. Heravi, *Guide to convolutional neural networks: a practical application to traffic-sign detection and classification*. Springer, 2017.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, June 2016.

[13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[14] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.