GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

OBJECT DETECTION AND CLASSIFICATION BETWEEN CRAWLING

A Report for the Evaluation 3 of Project 2

Submitted by

PRIYANK MISHRA

(1613101523/16SCSE101846)

in partial fulfillment for the award of the degree of

Bachelor of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

MR. ANURAG SINGH,

Professor

APRIL / MAY- 2020

TABLE OF CONTENTS

Abstract

Efficient and accurate CRAWLING detection has been an important topic in the advancement of computer vision systems. With the advent of deep learning techniques, the accuracy for CRAWLING detection has increased drastically. The project aims to incorporate state-of-the-art technique for CRAWLING detection with the goal of achieving high accuracy with a real-time performance. A major challenge in many of the CRAWLING detection systems is the dependency on other computer vision techniques for helping the deep learning based approach, which leads to slow and non-optimal performance. In this project, we use a completely deep learning based approach to solve the problem of CRAWLING detection in an end-to-end fashion. The network is trained on the most challenging publicly available dataset (PASCAL VOC), on which a CRAWLING detection challenge is conducted annually. An optical camera-based intrusion classification system (Light Intrusion DeTection systEm named as acronym LITE) for an outdoor setting was recently developed by a superset of the authors. The system classified between human and animal images captured in a side-view manner based on the height. Based on the system and algorithm design, most probably human-crawl would be classified as animal by the LITE. In this paper, classification between human-crawl and animal is addressed. In addition to this work, classification of person with weapon versus person with vehicle is also addressed (referred as person with CRAWLING) to provide more information about the type of intrusions. A Convolutional Neural Network (CNN) based approach is used to solve the above stated two problems. In the case of "person with CRAWLING" classification, a study of different CNN architectures was carried out and analysis corresponding to that is presented. In case of human crawl vs animal movement, performance results corresponding to only the best architecture model is provided among the many tried models. Further on, additional insights are provided about the classification using the attention heat maps and t-SNE plots. The test classification accuracies for human-crawl vs animal and person with CRAWLING classification on the recorded data are close to 95.65% and 90%, respectively. The LITE, having the Odroid C2 (OC2) Single-Board Computer (SBC) with CNN-based classification algorithm for human-crawl versus animal task ported on it, was deployed in an outdoor setting for a realtime deployment. It provided a classification accuracy close to 92%.

Introduction

The goal of this article is to provide an easier human-machine interaction routine when user authentication is needed through CRAWLING detection and recognition. With the aid of a regular web camera, a machine is able to detect and recognize a person's CRAWLING; a custom login screen with the ability to filter user access based on the users' CRAWLING features will be developed. The CRAWLING gives of this thesis are to provide a set of detection algorithms that can be packaged in an easily portable framework among the different processor architectures we see in machines (computers) today. These algorithms must provide at least a 95% successful recognition rate, out of which less than 3% of the detected CRAWLINGs are false positive is processed to crop and extract the person's CRAWLING for easier recognition. CRAWLING Recognition where that detected and processed CRAWLING is compared to a database of known CRAWLINGs, to decide who that person is.

Since 2002, CRAWLING detection can be performed fairly easily and reliably with Intel's open source framework called OpenCV . This framework has an inbuilt CRAWLING Detector that works in roughly 90-95% of clear photos of a person looking forward at the camera. However, detecting a person's CRAWLING when that person is viewed from an angle is usually harder, sometimes requiring 3D Head Pose Estimation. Also, lack of proper brightness of an image can greatly increase the difficulty of detecting a CRAWLING, or increased contrast in shadows on the CRAWLING, or maybe the picture is blurry, or the person is wearing glasses, etc. CRAWLING recognition however is much less reliable than CRAWLING detection, with an accuracy of 30-70% in general. CRAWLING recognition has been a strong field of research since the 1990s, but is still a far way away from a reliable method of user authentication. More and more techniques are being developed each year.

The EigenCRAWLING technique is considered the simplest method of accurate CRAWLING recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate. OpenCV was started at Intel in 1999 by Gary Bradski for the purposes of accelerating research in and commercial applications of computer vision in the world and, for Intel, creating a demand for ever more powerful computers by such applications. Vadim Pisarevsky joined Gary to manage Intel's Russian software OpenCV team.

Over time the OpenCV team moved on to other companies and other Research. Several of the original team eventually ended up working in robotics and found their way to Willow Garage. In 2008, Willow Garage saw the need to rapidly advance robotic perception capabilities in an open way that leverages the entire research EigenCRAWLINGs is considered the simplest method of accurate CRAWLING recognition, but many other (much more complicated) methods or combinations of multiple methods are slightly more accurate.

.Most resources on CRAWLING recognition are for basic Neural Networks, which usually don't work as well as EigenCRAWLINGs does. And unfortunately there are only some basic explanations for better type of CRAWLING recognition than EigenCRAWLINGs, such as recognition from video and other techniques at the CRAWLING Recognition Homepage or 3D CRAWLING Recognition Wikipedia page and Active Appearance Models page . But for other techniques, you should read some recent computer vision research papers from CVPR and other computer vision conferences. Most computer vision or machine vision conferences include new advances in CRAWLING detection and CRAWLING recognition that give slightly better accuracy. So for example you can look for the CVPR10 and CVPR09 conferences.

Figure 1

Existing system

CRAWLING detection methods can be grouped in five categories, each with merits and demerits: while some are more robust, others can be used in real-time systems, and others can be handle more classes, etc. Table 1 gives a qualitative comparison.

Method	Coarse-to-fine and boosted classifiers	Dictionary based	Deformable part-based models	Deep learning	Trainable image processing architectures
Accuracy	++	+=	++	++	+=
Generality	==	++	+=	++	+=
Speed	++	+=	==	+=	+=
Advantages	Real-time, it can work at small resolutions	Representation can be shared across classes	It can handle deformations and occlusions	Representation can be transfered to other classes	General-purpose architecture that can be used is several modules of a system
Drawbacks/ requirements	Features are predefined	It may not detect all object instances	It can not detect small objects	Large training sets specialized hardware (GPU) for efficiency	The obtained system may be Too specialized for a particular setting
Typical applications	Robotics, security	Retrieval, search	Transportation pedestrian detection	Retrieval, search	HCI, health, robotics

Accuracy: ++, High; +=, Good; ==, Low.
Speed: ++, real-time (15 fps or more); +=, online (10–5 fps); ==, offline (5 fps or more).
Generality: ++ (+=), applicable to many (some) object classes; ==, depend on features designed for specific classes.

3.1 Coarse-to-Fine and Boosted Classifiers

The most popular work in this category is the boosted cascade classifier of Viola and Jones (2004). It works by efficiently rejecting, in a cascade of test/filters, image patches that do not correspond to the CRAWLING. Cascade methods are commonly used with boosted classifiers due to two main reasons: (i) boosting generates an additive classifier, thus it is easy to control the complexity of each stage of the cascade and (ii) during training, boosting can be also used for feature selection, allowing the use of large (parametrized) families of features. A coarse-to-fine cascade classifier is usually the first kind of classifier to consider when efficiency is a key requirement.

3.2. Dictionary Based

The best example in this category is the Bag of Word method [e.g., Serre et al. (2005) and Mutch and Lowe (2008)]. This approach is basically designed to detect a single CRAWLING per image, but after removing a detected CRAWLING, the remaining CRAWLINGs can be detected [e.g., Lampert et al. (2009)]. Two problems with this approach are that it cannot robustly handle

well the case of two instances of the CRAWLING appearing near each other, and that the localization of the CRAWLING may not be accurate.

3.3. Deformable Part-Based Model

This approach considers CRAWLING and part models and their relative positions. In general, it is more robust that other approaches, but it is rather time consuming and cannot detect CRAWLINGs appearing at small scales. It can be traced back to the deformable models (Fischler and Elschlager, 1973), but successful methods are recent (Felzenszwalb et al., 2010b). Relevant works include Felzenszwalb et al. (2010a) and Yan et al. (2014), where efficient evaluation of deformable part-based model is implemented using a coarse-to-fine cascade model for faster evaluation, Divvala et al. (2012), where the relevance of the part-models is analyzed, among others [e.g., Azizpour and Laptev (2012), Zhu and Ramanan (2012), and Girshick et al. (2014)].

3.4. Deep Learning

One of the first successful methods in this family is based on convolutional neural networks (Delakis and Garcia, 2004). The key difference between this and the above approaches is that in this approach the feature representation is learned instead of being designed by the user, but with the drawback that a large number of training samples is required for training the classifier. Recent methods include Dean et al. (2013), Huval et al. (2013), Ouyang and Wang (2013), Sermanet et al. (2013), Szegedy et al. (2013), Zeng et al. (2013), Erhan et al. (2014), Zhou et al. (2014), and Ouyang et al. (2015).

3.5. Trainable Image Processing Architectures

In such architectures, the parameters of predefined operators and the combination of the operators are learned, sometimes considering an abstract notion of fitness. These are general-purpose architectures, and thus they can be used to build several modules of a larger system (e.g., CRAWLING recognition, key point detectors and CRAWLING detection modules of a robot vision system). Examples include trainable COSFIRE filters (Azzopardi and Petkov, 2013, 2014), and Cartesian Genetic Programming (CGP) (Harding et al., 2013; Leitner et al., 2013).

Proposed System

To improve the recognition performance, there are many things that can be improved here, some of them being fairly easy to implement. For example, you could add color processing, edge detection, etc. You can usually improve the CRAWLING recognition accuracy by using more input images, at least 50 per person, by taking more photos of each person, particularly from different angles and lighting conditions. If you can't take more photos, there are several simple techniques you could use to obtain more is that at the heart of the algorithm, it is matching images by basically doing the equivalent of subtracting the testing image with a training image to see how similar they are. This would work fairly well if a human performed it, but the computer just thinks in terms of pixels and numbers. So if you imagine that it is looking at one pixel in the test image, and subtracting the gray scale value of that pixel with the value of the pixel in the EXACT same location of each training image, and the lower the difference then the better the match. So if you just move an image by a few pixels across, or use an image that is just a few pixels bigger or has a few more pixels of the forehead showing than the other image, etc, then it will think they are completely different images! This is also true if the background is different, because the code doesn't know the difference between background and foreground (CRAWLING), which is why its important to crop away as much of the background as you can, such as by only using a small section inside the CRAWLING that doesn't include any background at all. Since the images should be almost perfectly aligned, it actually means that in many cases, using small low-res images (such as by shrinking the images to thumbnail size) can give better recognition results than large hi-res images! Also, if the images are perfectly aligned, if the testing image is a bit brighter than the training image then it will still think there is not much of a match. Histogram Equalization can help in many cases but it can also make things worse in other cases, so differences in lighting is a difficult & common problem. There are also issues such as if there was a shadow on the left of the nose in the training image and on the right in the test image then it will often cause a bad match, etc. That's why CRAWLING recognition is relatively easy to do in realtime if you are training on someone and then instantly

Implementation Details

The project is implemented in python 3. Tensor ow was used for training the deep network and OpenCV was used for image pre-processing.

The system speci cations on which the model is trained and evaluated are mentioned as follows: CPU - Intel Core i7-7700 3.60 GHz, RAM - 32 Gb, GPU - Nvidia Titan Xp.

4.2.1 Pre-processing

The annotated data is provided in xml format, which is read and stored into a pickle le along with the images so that reading can be faster. Also the images are resized to a xed size.

4.2.2 Network

The entire network architecture is shown in Fig. 13. The model consists of the base network derived from VGG net and then the modied convolutional layers for ne-tuning and then the classier and localizer networks. This creates a deep network which is trained end-to-end on the dataset.

Source Code

```
fromimageai.Detection import CRAWLINGDetection
importos
execution_path = os.getcwd()
detector = CRAWLINGDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath(os.path.join(execution_path , "resnet50_coco_best_v2.0.1.h5"))
detector.loadModel()
detections = detector.detectCRAWLINGsFromImage(input_image=os.path.join(execution_path ,
"image.jpg"), output_image_path=os.path.join(execution_path , "imagenew.jpg"))
foreachCRAWLING in detections:
```

print(eachCRAWLING["name"] , " : " , eachCRAWLING["percentage_probability"])

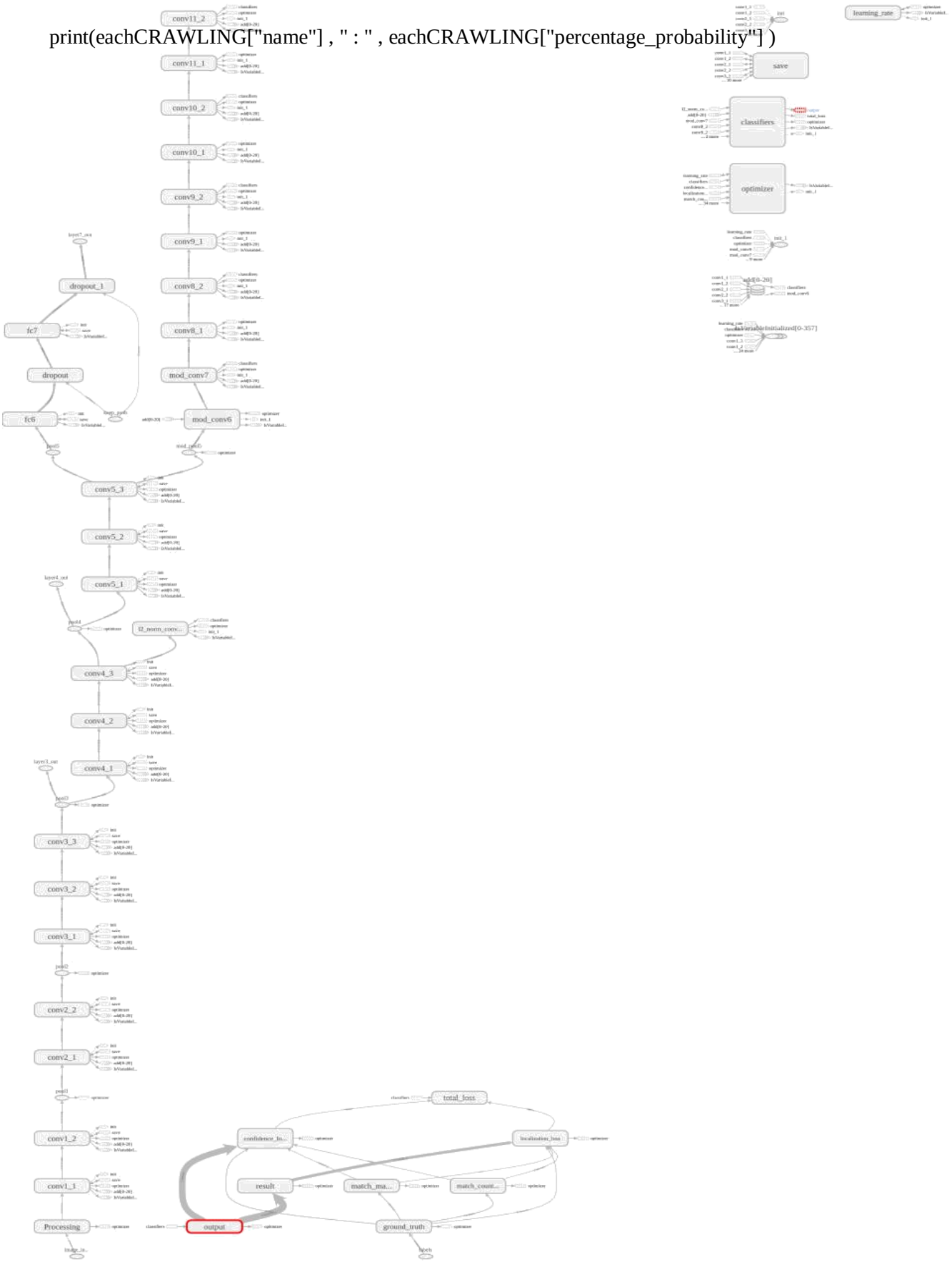


Figure 2

Architecture Diagrams

The network used in this project is based on Single shot detection (SSD) [5]. The architecture is shown in Fig. 7.

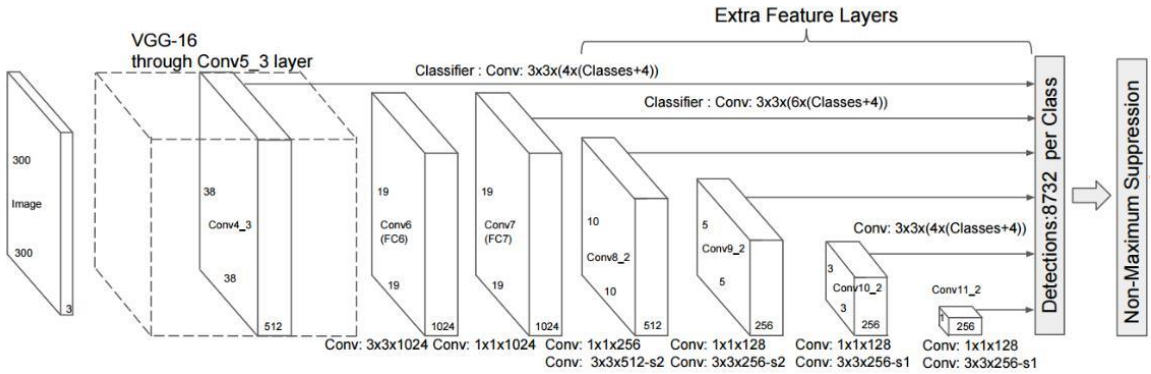


Figure 3: SSD Architecture

The SSD normally starts with a VGG [6] model, which is converted to a fully convolu-tional network. Then we attach some extra convolutional layers, that help to handle bigger CRAWLINGs. The output at the VGG network is a 38x38 feature map (conv4 3). The added layers produce 19x19, 10x10, 5x5, 3x3, 1x1 feature maps. All these feature maps are used for predicting bounding boxes at various scales (later layers responsible for larger CRAWLINGs). Thus the overall idea of SSD is shown in Fig. 8. Some of the activations are passed to the sub-network that acts as a classifier and a localizer.

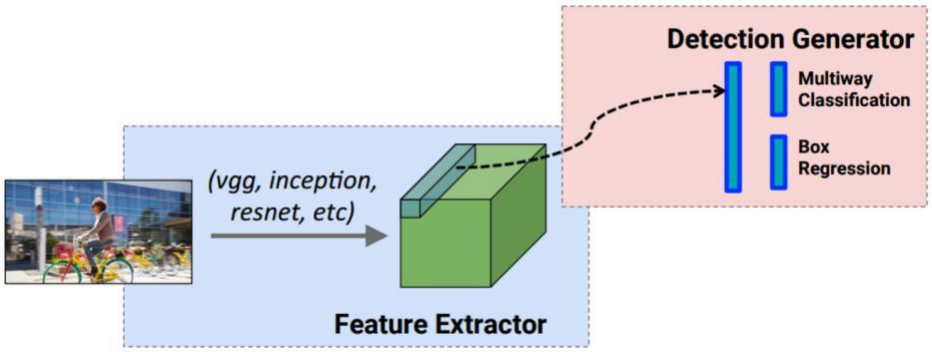


Figure 4: SSD Overall Idea

Anchors (collection of boxes overlaid on image at different spatial locations, scales and aspect ratios) act as reference points on ground truth images as shown in Fig. 9.

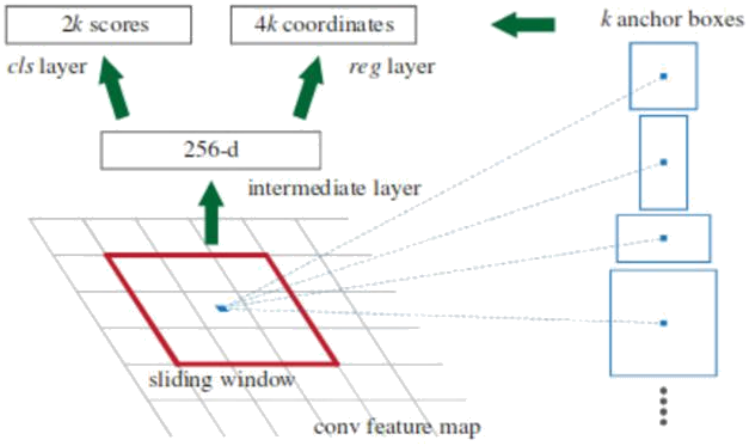


Figure 5: Anchors

A model is trained to make two predictions for each anchor:

A discrete class

A continuous o set by which the anchor needs to be shifted to t the ground-truth bounding box

During training SSD matches ground truth annotations with anchors. Each element of the feature map (cell) has a number of anchors associated with it. Any anchor with an IoU (jaccard distance) greater than 0.5 is considered a match. Consider the case as shown in Fig. 10, where the cat has two anchors matched and the dog has one anchor matched. Note that both have been matched on different feature maps.

Result

Input Ground Truth Prediction

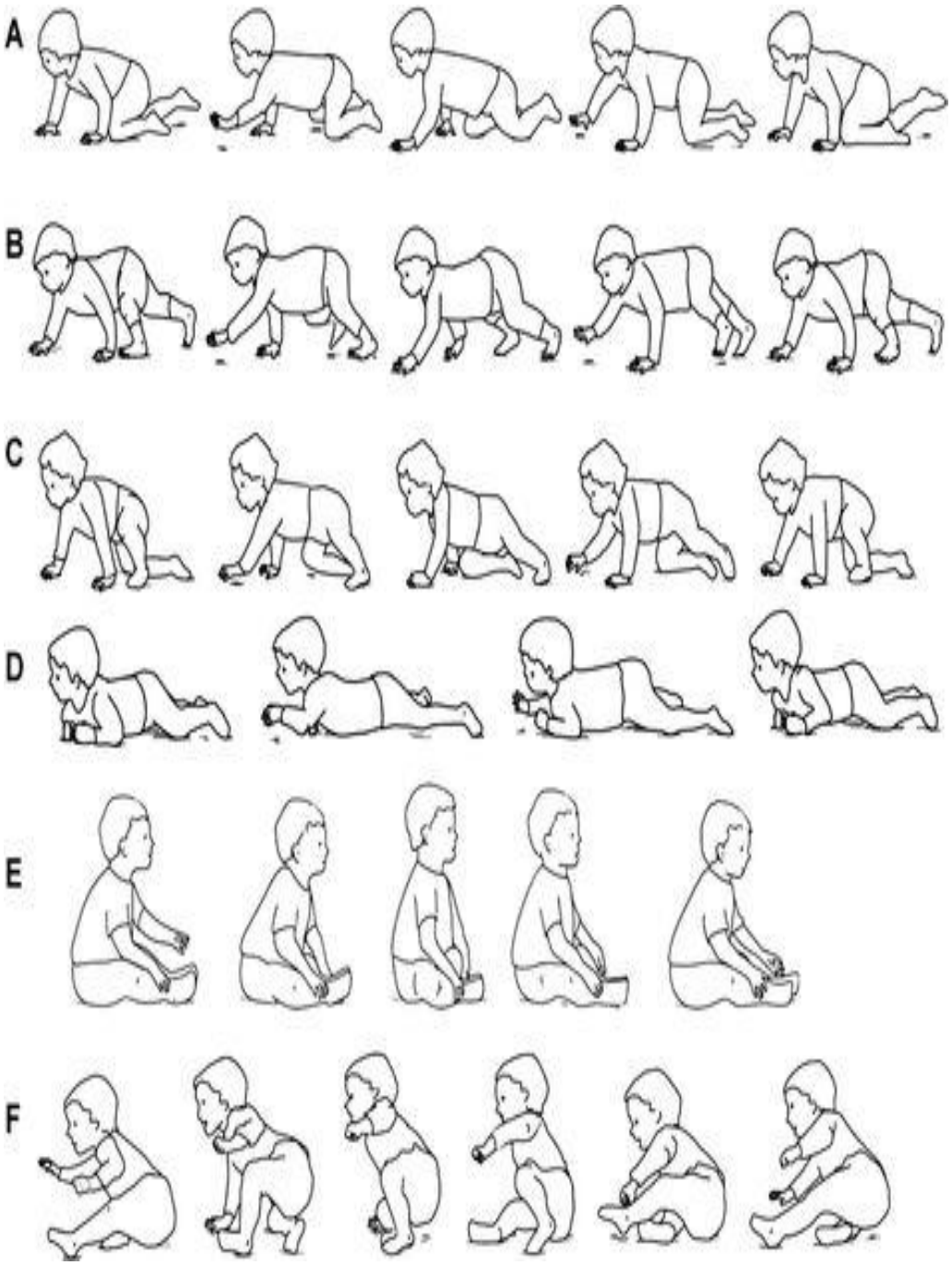


Table 1: Detection results

The results on custom dataset are shown in Table 2.

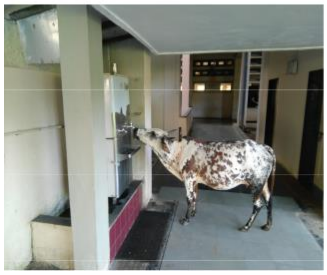
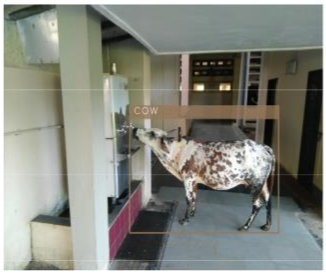
Input	Ground Truth	Prediction
		
		

Table 2: Detection results on custom dataset.

The system handles illumination variations thus providing a robust detection. In Fig. 14 the same person is standing in the shade and then in the sunny environment.

a) High illumination (b) Low illumination

Figure 6: Detection robust to illumination variation

However, occlusion creates a problem for detection. As shown in Fig. 15, the occluded birds are not detected correctly.

Also larger CRAWLING dominated when present along with small CRAWLINGs as found in Fig.

16. This could be the reason for the average precision of smaller CRAWLINGs to be less when compared to larger CRAWLINGs. This has been reported in the next section.

(a) Only small CRAWLING in image (b) Small and large CRAWLING in image

Figure 8: Domination of larger CRAWLING in detection

Conclusion

An accurate and efficient CRAWLING detection system has been developed which achieves compa-rable metrics with the existing state-of-the-art system. This project uses recent techniques in the field of computer vision and deep learning. Custom dataset was created using labelImg and the evaluation was consistent. This can be used in real-time applications which require CRAWLING detection for pre-processing in their pipeline.

An important scope would be to train the system on a video sequence for usage in tracking applications. Addition of a temporally consistent network would enable smooth detection and more optimal than per-frame detection.

References

- "What is CRAWLING Recognition? - Definition from Techopedia". Techopedia.com.
- "CRAWLING Recognition: Who's Tracking You in Public?". Consumer Reports
- https://www.upwork.com/hiring/for-clients/pros-cons-CRAWLING-recognition-technology-business/
- https://en.wikipedia.org/wiki/CRAWLING_recognition_system#References
- Ross Girshick, Je Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate CRAWLING detection and semantic segmentation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
- Ross Girshick. Fast R-CNN. In International Conference on Computer Vision (ICCV), 2015.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time CRAWLING detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), 2015.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unied, real-time CRAWLING detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- Wei Liu, DragomirAnguelov, DumitruErhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In ECCV, 2016.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2