Measuring Busyness of University Facilities

ADA University
School of IT and Engineering

Galib Aili
galili2018@ada.edu.az

Narmin Orujova
norujova2018@ada.edu.az

Isa Mammadov
ismammadov2018@ada.edu.az

ABSTRACT
Lack of highly functional occupancy measurement systems on actively used facilities may lead to various scheduling, efficiency and cost-saving complications. Hence, introducing the application, which will measure busyness of actively used facilities, students, alongside with universities` academic, administrative, and technical staff, will benefit from decreasing amount of times spent on busy facilities, and do appropriate utilization related policies.

CCS CONCEPTS
Machine Learning Approaches → Neural Networks; Image Manipulation → Image processing;

KEYWORDS
Utilization, Occupancy Measurement, Image Classification, Object Detection, Convolutional Neural Networks, Proposal Generation

1. INTRODUCTION

In discussed proposal, occupancy measurement is achieved by identifying number of empty (circle_empty, rectangle_empty) and full (circle_full, circle_empty) circle and rectangular tables, which are most used two types of tables at University canteens, and deriving occupancy factor, which is percentage of total empty tables relative to total tables. Identification is based on machine learning techniques, neural networks, namely convolutional neural networks, the Faster R-CNN model. The model was trained and tested with self-build "ADA University Tables" dataset, comprised from pictures of university canteen's tables.

2. IMPLEMENTATION

To use for model training, the dataset of 3500+ pictures in total, taken at different angles, level of brightness, and times of the day was build. Approximately 1700 of them were filtered and labeled manually by team members. The components of the dataset pictures have 4 possible labels, "circle_empty", "circle_full", "rectangle_empty", "rectangle_full". The model used for training combines the Fast R-CNN classifier with RPN and wins in time-accuracy competition. Two networks, RPN and Fast R-CNN share the convolutional layer. The training process is tuned, so RPN`s output is input for classifier. The proposal- generating network takes the feature map of last convolutional layer and maps it to a feature of lower dimension. In this ways anchors are proposed which later are sorted with non-maximum suppression (NMS). Since the number of proposals is redundant, the elimination of part of them does not worsen the accuracy, on the contrary fastens the speed [1].

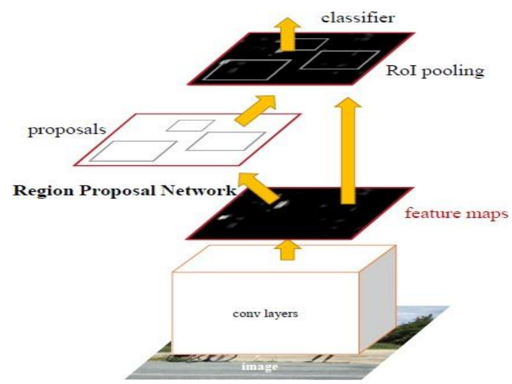

Figure 1: The Faster R-CNN Model

During the training, the loss function is calculated separately for the box-regression and box-classification layers and combined together according to the following formula [1].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

3. RESULTS AND DISCUSSION

Loss function's graph was reference point to observe the ongoing process and its efficiency while training. Eventually, we got the value of this function as 0.066, almost 0, indicating successful results. The model was able to detect the number of full tables at accuracy of 0.8/1, and number of empty tables with 0.76/1.

4. FUTURE WORK

As a future work of the project we are planning to enlarge the application scope of the system and increase the accuracy. For now, the model and dataset are tuned to classify tables at university's canteen taken as an input from security cameras. However, the problem of measuring occupancy, and empty place finding during day's peak times is actual at universities, along with other places of people's congestion, such as Malls. This ensures the fact that empty place tracker can have very productive application at malls as well. This is an objective that motivates us to enlarge our project and expand its application to other public areas.

REFERENCES

[1] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,*39*(6), 1137-1149. doi:10.1109/tpami.2016.2577031