Assessing Crowd Size at Essential Stores in the Age of COVID-19

Steve Burke

Stanford University

sfburke@stanford.edu

1 Introduction

In response to COVID-19 spreading around the globe, people have been trying to distance themselves as much as possible from others in order to limit the spread of infection and stay healthy. One necessity that cannot be avoided for many people is going out to buy essential goods, including groceries and prescription drugs. There have been many documented cases in the news of extremely long lines outside of these buildings. This project explores developing a deep learning algorithm capable of accurately detecting the number of people waiting in lines outside of essential stores during COVID-19, which can be deployed to send periodic updates to local citizens so that they can plan the best times to go out and get essential goods. The input to this system is an RGB image of an outside location of an essential store, and the output is a prediction on the number of people in the image.

This task includes challenges that are present in many other applications involving counting people in small crowds. One principle challenge is that of occlusion: individuals may occlude one another and therefore obscure features that would be useful in determining if a person is present in a scene. This particular COVID-related application poses potential additional challenges - individuals can be obscured by objects like umbrellas and shopping carts, and in many instances, individuals' faces are covered by facemasks.

2 Related Work

Gao et al. [1] provides a comprehensive survey of past research in crowd estimation and counting, and the paper includes discussion on over 220 different works on the subject. Many of these methods involve using CNN-based density map estimation, which has proven especially useful on datasets that include very dense or congested crowds, on the order of hundreds to thousands of individuals. Our task is slightly different – classifying groups on the order of 0-25 individuals, who are often separated due to social distancing recommendations, but sometimes occlude one another in the image. Stewart et al. [2] proposed an end-to-end approach for detecting people in small crowds and occluded scenes that builds upon past work with standard object detection approaches like R-CNNs, and tested these algorithms on image datasets of coffee shop scenes and pedestrians in crosswalks. This work provided much of the motivation for this project, by indicating that object detection methods could be successful in assessing small crowd sizes.

3 Dataset

For this project, I compiled a test dataset of 50 images from a google search of COVID-19 grocery store lines [3]. Across these images, the average number of people visible in the images is 9. The minimum number of people is zero and the maximum is 22. An example image is shown below. The images are all different sizes, but on average are 1200 x 780 pixels.

As described in the next section, the baseline test for this project consisted of using a model that was pre-trained on the COCO dataset [4], a large object detection dataset consisting of >200k labeled images, 1.5 million object instances, and 80 object categories – including a "person" category. One important point described in the COCO dataset paper is that annotations were capped at 15 instances of objects per image. In the case of a larger number of instances in an image (e.g. for human crowds), they annotated only 15 instances and labeled the rest of the instances collectively as a crowd. This has some performance implications for conducting inference on the COVID test images, where 6 out of the 50 images in the test set contain crowds of a size larger than 15 people.

A larger dataset was used for training a second object detection model, specifically for detecting the heads of people in the COVID images. This dataset was introduced as part of work done by Vu et. Al [5] on context-aware CNNs for person head detection. The dataset consists of 224,740 images from 16 different Hollywood films, with a total of 369,846 bounding box label instances of human heads. In some of these images, there are multiple actors in the scene, and in others there is only one or zero human heads visible. The images take place in a variety of scenes and a few of the films are in black and white. Due to time constraints, I reduced the number of training images down to 226 images, with a corresponding total number of labeled bounding boxes of 369. Images were chosen roughly equally from each of the 16 films. An important distinction to note is that this dataset had bounding boxes of just the person's head, whereas the baseline pre-trained model had been trained on bounding box annotations around the entire person.

Two example images from the Hollywood training dataset [5]

4 Methods

For the baseline model, I used a pre-trained model from the Deep-Learning Coursera YOLO car detection assignment [6]. The pre-trained weights for the keras model (YOLOv2) in this case had been trained with the COCO dataset, described above. The pre-trained model and weights can be found here [7]. I modified the provided coursera jupyter notebook to be able to perform inference on a series of test images (in this case, the COVID test set of 50 images) of different dimensions. After tuning some parameters for this model and analyzing results, I chose to train a second model for detecting just people's heads in images, in order to use that second model in an ensemble with the baseline model, to try and capture as many people in the scenes as possible.

For the second model, I made use of an existing codebase [8] for training datasets on a YOLOv3 architecture. The codebase included scripts for downloading pre-trained weights and converting them into keras

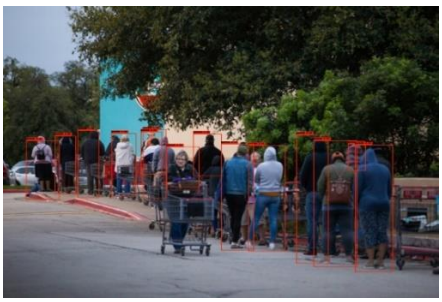
format. The weights were pre-trained on the Imagenet dataset [9]. I wrote scripts to select a subset of 236 images from the Hollywood image dataset and transform the .xml bounding box annotations to a .csv format that was compatible with the yolov3 codebase. The codebase transfer learning methodology involved first training the last layers of the model for a set of epochs and then unfreezing all layers to fine-tune the model by updating weights on all layers. I trained this process for 51 epochs on the final layers of the model, and then another 34 epochs of fine-tuning before early stopping.

For context, the baseline and second model both made use of the YOLO architecture, which is part of a group of algorithms, like SSD or R-CNN, that are effective in object detection tasks. The YOLO model takes an image as input and uses a convolutional neural network to output predictions for each region of the image. The predictions indicate whether an object is present in that region of the image, where the object is, and which class of objects it is. The location of the object is defined by one of the pre-defined anchor box shapes for the algorithm, and the prediction indicates which anchor box best indicates the location and size of the object. Non-max suppression and intersection over union calculations are used to filter out the resulting predictions for all anchor boxes and select the one which is most likely to define the object, without repeats.

5 Results and Discussion

Two example results from running the baseline pre-trained YOLOv2 model on the COVID test images are shown below. Across the test dataset of 50 images, there are 443 total instances of people present. In the first baseline test, the model correctly detected 204 instances of people in the images, which is a corresponding **recall performance of 46%.** The model only made four false positive predictions of people, and the resulting **precision was 98%.** In these four false prediction cases, the bounding boxes were close to a person in the image but not capturing a majority of the person in the bounding box. In many of the images, the model had better success in detecting people when they were closer to the camera, larger, and less occluded by other people.

Changing the threshold for the confidence of bounding box predictions from 60% to 40% and maintaining the original non-maximum suppression (NMS) value of 0.5 led to a resulting improved **recall performance of 69%** while the **precision decreased to 94%.** Two example images of the improved performance are below, with a thinner bounding box visualization for clarity in cases of occlusion.



The image in the COVID test set with the lowest recall (33%, only 7 correct detections out of 21 people in the image) is below on the left. It is representative of a consistent performance issue across the test dataset – when people are densely packed in an image, the algorithm does not always detect them properly. The image below on the right is an example image where setting the confidence threshold lower reduced the precision – in this case, two stickers on the windows of the store were mis-classified as people. Other challenging aspects in the test set that I monitored for issues were the presence of facemasks, reflections in store windows, darker images taken at night-time, and people holding umbrellas. None of these attributes appeared to affect the performance of the baseline model, but these should be evaluated with a much larger test dataset in order to draw meaningful conclusions.

Although parts of the human body are often occluded in the cases with missed detections in this test set, the front or back of the head of the person was still visible in 86 out of the 137 cases of missed detection. I therefore theorized that detecting human heads in these images with a second model and combining those results with the baseline person-level baseline inference results could potentially improve the recall performance from 69% to 89% on the test set.

After training a second YOLOv3 model on images of Hollywood actors' heads, I found that performing inference on the COVID test dataset with this model alone led to a **recall of 37.9% and precision of 77.5%.** There were 22 instances where the heads of people in the test set were detected by this model, where the corresponding person was not detected by the baseline YOLOv2 model. When using the baseline model and head detector model in ensemble to detect people in an image, the **combined recall was improved to 74% from the baseline 69%, and the combined precision decreased from baseline 94% to 88.7%.** The images below show a comparison between the baseline model (on the left) and the head-detection model (on the right). In this specific image in the test set, there were 5 cases of heads that were detected where the corresponding person was not identified by the baseline model.

Future Work

The results above demonstrate that combining the results of two separate YOLO models together for detecting the number of people in small crowds shows promise. However, it's important to highlight that the test dataset was rather small, so trials on much larger sets of test data, perhaps from a day's worth of images taken outside of an actual grocery store, would provide a deeper understanding of the model performance. It is promising that the head detector model performed reasonably well given the limited size of the 236 images used from the Hollywood dataset for transfer learning. In the future, the head detection model could be trained with thousands of images from the Hollywood dataset, and perhaps the images from the black and white films could be removed from the datasets because they are not relevant to the task at hand. With larger datasets, I would also split the images into more thorough train/validation/test sets to facilitate hyperparameter tuning and optimize performance further. Other possible future directions could be to compare this architecture to other object-detection model architectures, as described earlier in this paper, or to annotate one of the datasets further and train a single YOLO model to predict two classes – a person and a person's head – in order to reduce the number of models needed from two to one.

This system does not need to achieve perfect recall in order to provide a close estimate of the number of people in small crowds outside essential stores. For this reason, I see the results presented in this paper as promising that this type of system could be effectively deployed and would provide useful information for positively impacting the health and well-being of citizens in our community, who are trying to socially distance themselves as effectively as possible while procuring essential goods.

References

[1] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang. CNN-based density estimation and crowd counting: A survey. In arXiv preprint arXiv:2003.12783, 2020.

[2] R. Stewart, M. Andriluka, and A. Y. Ng. End-to-end people detection in crowded scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2325–2333, 2016

[3] Sources for these images can be found in the accompanying github repository for the project: https://github.com/stevefburke/covid_crowds

[4] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., and Zitnick, C. L. (2014b). Microsoft COCO: Common Objects in Context. In ECCV.

[5] Vu, T.H., Osokin, A., Laptev, I.: Context-aware cnns for person head detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2893–2901 (2015)

[6] A. Ng, K. Katanforoosh, and Y. Mourri. Convolutional Neural Networks, Car Detection for Autonomous Driving. 2020. https://www.coursera.org/

[7] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. In: arXiv preprint arXiv:1804.02767, 2018. https://pjreddie.com/darknet/yolo

[8] Muehlemann, Anton. How to train your own YOLOv3 detector from scratch. Medium, 2019. https://blog.insightdatascience.com/how-to-train-your-own-yolov3-detector-from-scratch-224d10e55de2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009