Implementation of a Computer Vision Framework for Tracking and Visualizing Face Mask Usage in Urban Environments

Gabriel T.S. Draughon
Civil and Environmental Engineering
University of Michigan
Ann Arbor, MI, USA
draughon@umich.edu

Peng Sun, PhD
Civil and Environmental Engineering
University of Central Florida
Orlando, FL, USA
peng.sun@ucf.edu

Jerome P. Lynch, PhD
Civil and Environmental Engineering
University of Michigan
Ann Arbor, MI, USA
jerlynch@umich.edu

***Abstract*—The COVID-19 pandemic is an evolving situation in the United States and is spreading at alarming rates. The adoption of public health-informed hygienic practices can have a large impact on community transmission of COVID-19 including the wearing of face masks in public settings. Convolutional Neural Networks (CNN) can be trained to classify people wearing face masks with impressive accuracy. However, current face mask datasets contain clear, high-resolution close-up images of individuals with face masks which is unrepresentative of the lower fidelity images of distant faces more prominent in urban camera images. This paper proposes a practical deep learning computer vision framework for detection and tracking of people in public spaces and the use of face masks. A custom 6,000 image face mask dataset curated from over 50 hours of urban surveillance camera footage is created in this work. CNN-based detectors trained using the dataset are used to perform person detection and face mask classification. Then, a multi-target tracking module extracts individual trajectories from frame-by-frame detection. By associating detected face masks with tracked individuals, overall face mask usage can be estimated. The framework is implemented on several surveillance cameras along the Detroit RiverWalk, a 5-kilometer pedestrian park connecting various greenways, plazas, pavilions, and open green spaces along the Detroit River in Detroit, Michigan. The detection of park user types is shown to have an average precision of 89% and higher for most person classes with the mask detector having an accuracy of 96%. An interactive web application visualizes the data and is used by park managers to inform management decisions and assess strategies used to increase face mask usage rates.**

***Index Terms*—computer vision, multi-object tracking, face masks, COVID-19**

I. Introduction

On March 11, 2020 the World Health Organization (WHO) declared the COVID-19 outbreak a pandemic [1]. In the weeks that followed, federal and state governments in the United States made efforts to suppress the spread of the disease. State governments issued stay-at-home orders and retailers and restaurants were temporally restricted to delivery or curbside pick-up orders only. Due to economic pressures, states began reopening in stages at the start of summer allowing in-person dining and shopping, and other non-essential services to return. As states have reopened, confirmed daily COVID-19 cases have once again risen within the United States, setting a record for new cases in a day on July 16, 2020 [2]. The earliest and most optimistic estimates for when a vaccine may be available is still months away [3]. As restrictions are eased, it is imperative that cities adopt public health-informed strategies that aim to help mitigate the spread of COVID-19. Strategies include wearing face masks and social distancing (maintaining 1.5 meters from others) when in public [4]. Maintaining these protective measures are essential to resumption of public life including the use of public spaces like city parks. For these reasons, it is important for cities to track public face mask usage and to adopt strategies that encourage increasing face mask usage.

Recent work in computer vision shows face mask detection tools based on Convolutional Neural Networks (CNN) are capable of achieving state-of-the-art results when trained and tested on popular public face mask datasets [5]. However, the application space of these tools are currently limited. Public face mask datasets, such as the MAsked FAces (MAFA) dataset [6], are primarily composed of high resolution close-up images of faces, with a large number of pixels detailing the face and mask. A face mask detection tool trained on such a dataset will inevitably struggle when working with much lower fidelity images which are much more common in practice. For example, cities wishing to utilize such a framework would likely need to pull images from pre-installed surveillance cameras in the city, which sit distant from crowds and are usually incapable of producing close-up images of faces. Furthermore, as presented, these previously proposed frameworks only work at the instance level, designed to only compute how many face masks are detected in the current frame. Without a tracking component, it would be difficult to more accurately measure the number of *unique* face masks detected and usage rates could only be a rough estimate.

This paper presents a practical computer vision sensing framework for person and face mask *tracking*. The face mask detection module was trained and validated on a challenging low resolution face mask dataset curated from 50+ hours of surveillance camera footage from the Detroit Riverfront. The face mask dataset is added to the authors' existing

978-1-7281-8294-0/20/$31.00 ©2020 IEEE

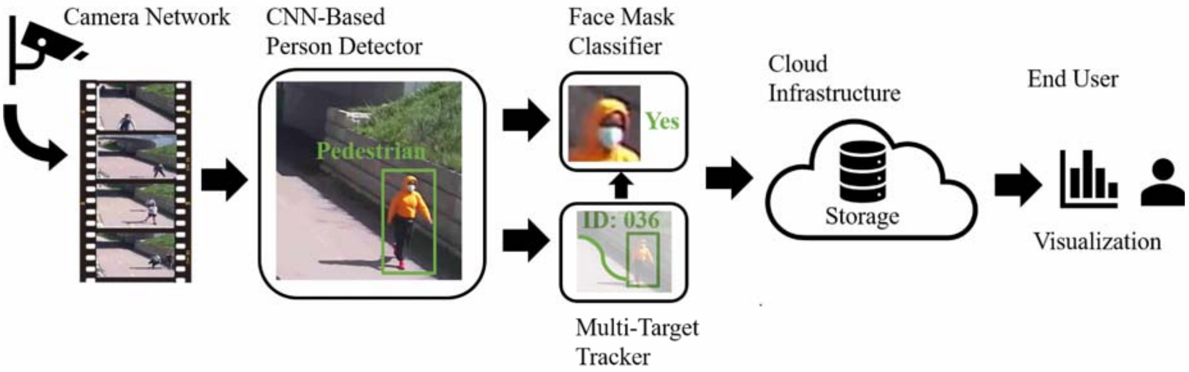


Fig. 1: Computer vision based framework for tracking and visualizing face mask usage rates by people in public spaces.

Objects in Public Open Spaces (OPOS) [7] dataset and is so named OPOS-FM. The proposed framework also includes a cloud infrastructure that hosts an interactive data visualization web application for public officials like park managers. The framework is implemented across several key locations along the Detroit Riverfront parks and accesses image data from an existing surveillance camera network installed in the space for security reasons. Park managers are able to access the processed data through the interactive online dashboard to track effectiveness of various strategies used to increase patron adherence to CDC face mask guidelines. The rest of the paper is as follows. Section II describes the components of the person and face mask tracking framework. In section III, the accuracy of this framework is presented along with results found when implementing the framework at the Detroit RiverWalk. Lastly, concluding remarks and direction of future work are given in Section IV.

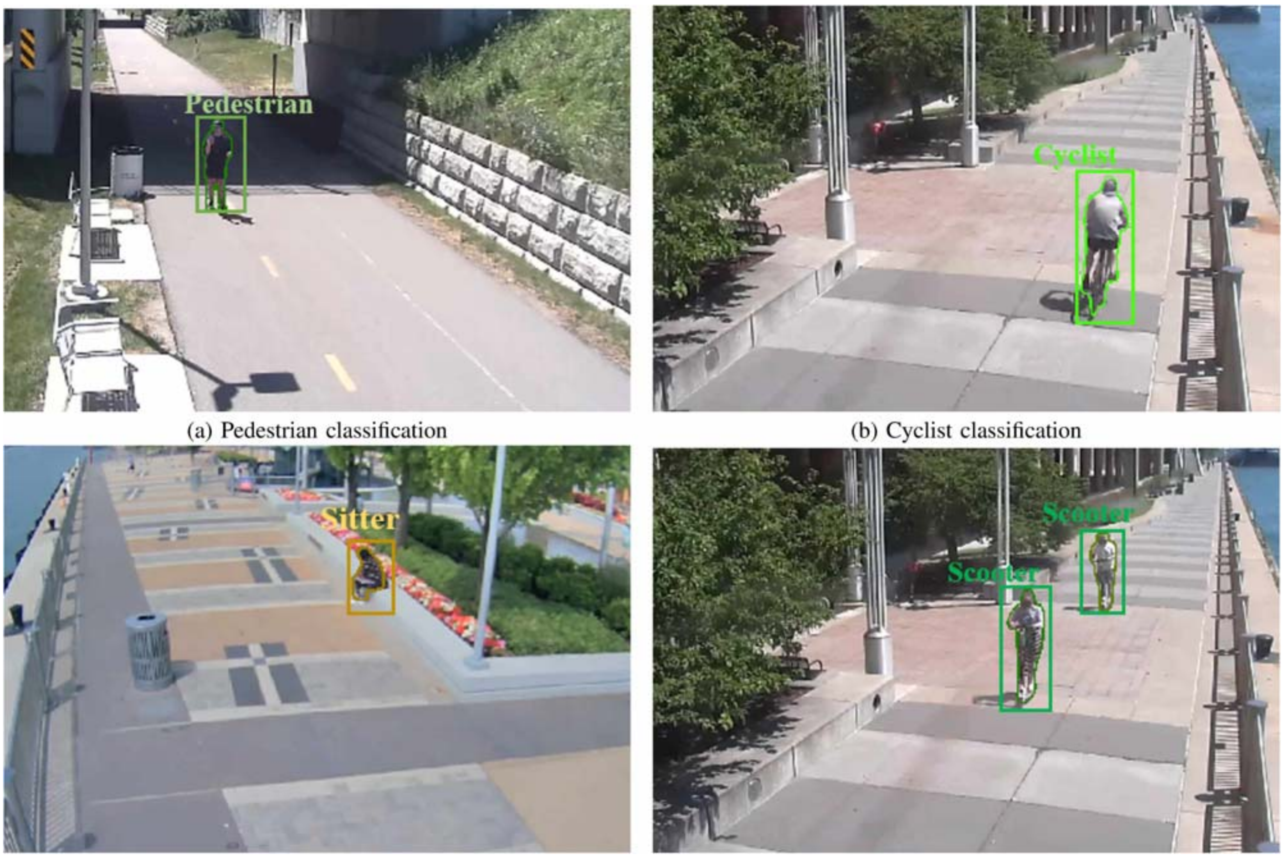
II. Methods

The face mask tracking framework proposed herein uses state-of-the-art computer vision methods for person and face mask detection and tracking. Additionally, it leverages Amazon Web Services (AWS) cloud infrastructure to store and visualize the data online. An overview of the framework is shown in Fig. 1. First, images from a data source (*e.g.*, surveillance camera), are passed into a CNN-based detector for person detection. The CNN-based detector is trained to classify people by their various activities such as “cyclist”, “sitter”, or “skater”. Table I contains the definition of the major person classes based on the OPOS dataset. Detected classes are then passed into a multi-target tracker which utilizes Kalman filtering, CNN feature extraction, and intersection over union (IoU) matching to assign unique IDs and track people movement. Additionally, cropped images of faces from the detected classes are passed into a separate CNN-based binary classifier to mark if the detected classes and their corresponding IDs are wearing a face mask. The number of detected people, activity classifications, and percentage wearing face masks is stored in Amazon’s Simple Storage Service (AWS S3) [8]. An interactive dashboard, which pulls data from AWS S3, is hosted online through Amazon’s Elastic Beanstalk service [9].

A. *CNN-based person detector*

While CNN architectures are the most popular choice for object detection today, there are other well known methods within the computer vision field. More traditional methods such as the Viola-Joins detector [10], which is popular with face detection, utilize handcrafted feature extractors such as Haar [11] or histogram of oriented gradients (HOG) [12]. These features can then be passed into a cascaded detection algorithm [10] or into a State Vector Machine (SVM) [12] for object classification. Recently, deep learning CNN architectures for object classification have become popular due to their ability to learn feature extractors (convolution kernels) that convolve with the original image. Additionally, convolution kernels which convolve with extracted features can be used to produce more complex and higher-level features. CNN architectures for object detection are usually classified as either single-stage [13], [14] or two-stage [15], [16]. Two-stage detectors first use a region proposal network (RPN) to generate regions of interest which are then passed to another network for object classification and bounding box regression. The popular single-stage architecture such as that used in YOLO [13] uses a single CNN to segment the input image into a grid of cells and predict bounding boxes and class probabilities for each cell. While single-stage detectors are faster than two-stage, they are less accurate.

Our framework uses the state-of-the-art Mask R-CNN architecture [15] to perform person detection and classification. Mask R-CNN strikes a good balance between inference speed and detection accuracy. The Mask R-CNN model is a two-stage detector. First, a CNN backbone structure such as ResNet [17] is used to extract features from the input image. A RPN then utilizes the extracted features to generate initial bounding boxes and region proposals (areas in the image suspected of containing objects). In the second stage, another neural network classifies the regions of interest and generates refined bounding boxes and pixel-wise masks of the detected objects.



(a) Pedestrian classification

(b) Cyclist classification

(c) Sitter classification

(d) Scooter classification

Fig. 2: Examples of Mask R-CNN person detection and activity classifications on frames taken from surveillance cameras at the Detroit Riverfront.

Examples of detections from the Mask R-CNN model are shown in Fig. 2.

Loss functions are used in Neural Network training as a measure of the error between prediction and ground truth. Loss functions evaluate candidate weights for the network and are used to find optimal solutions which minimize error. In this study, all parts of the Mask R-CNN model are trained together in an end-to-end fashion. The total loss function used for training is defined as:

$$L_{\text{Mask R-CNN}} = (L_{\text{rpn obj}} + L_{\text{rpn bbox}}) + (L_{\text{cls}} + L_{\text{bbox}} + L_{\text{mask}}) \quad (1)$$

where $L_{\text{rpn obj}}$ represents the loss of the RPN classifier (i.e., object or not a object) and $L_{\text{rpn bbox}}$ represents the loss for the bounding box regression within the RPN. L_{cls}, L_{bbox}, and L_{mask} represent the losses for activity classification (*e.g.*, cyclist, sitter), bounding box regression, and mask segmentation that is performed in the second stage of Mask R-CNN, respectively.

B. *Deep Sort tracker*

The Mask R-CNN detector processes each frame independently. A multi-object tracker is needed to associate objects detected in the current frame with the same objects detected in sequential frames. By doing so, objects can be assigned unique IDs and total number of unique objects (*e.g.*, cyclists) can be recorded in time and space. In this study, the Deep Sort algorithm [18] is used for person tracking and unique ID assignment. While other deep learning tracking methods exist [19], the Deep Sort algorithm is selected as it is more robust, especially in situations with visual occlusions (*e.g.*, when one pedestrian temporarily occludes, or blocks, the view of another pedestrian when crossing in front of them).

The Deep Sort algorithm utilizes a Kalman filter, feature vectors extracted from cropped detections, intersection over union (IoU) scores, and a Hungarian data association algorithm [20] for multi-object tracking, as shown in Fig. 3. Appearance feature vectors for each cropped detection are extracted using a CNN. The CNN used for feature extraction was trained on the MARS [21] dataset for person re-identification and outputs an 1×128 appearance feature vector. It is assumed that an appearance vector describing a detected person should not vary strongly from frame to frame (given the absence of any occlusions). Hence, appearance vectors of current detections can be compared against appearance vectors of previous detections by computing their cosine similarity. Cosine similarity is the cosine angle between two vectors. It is computed by

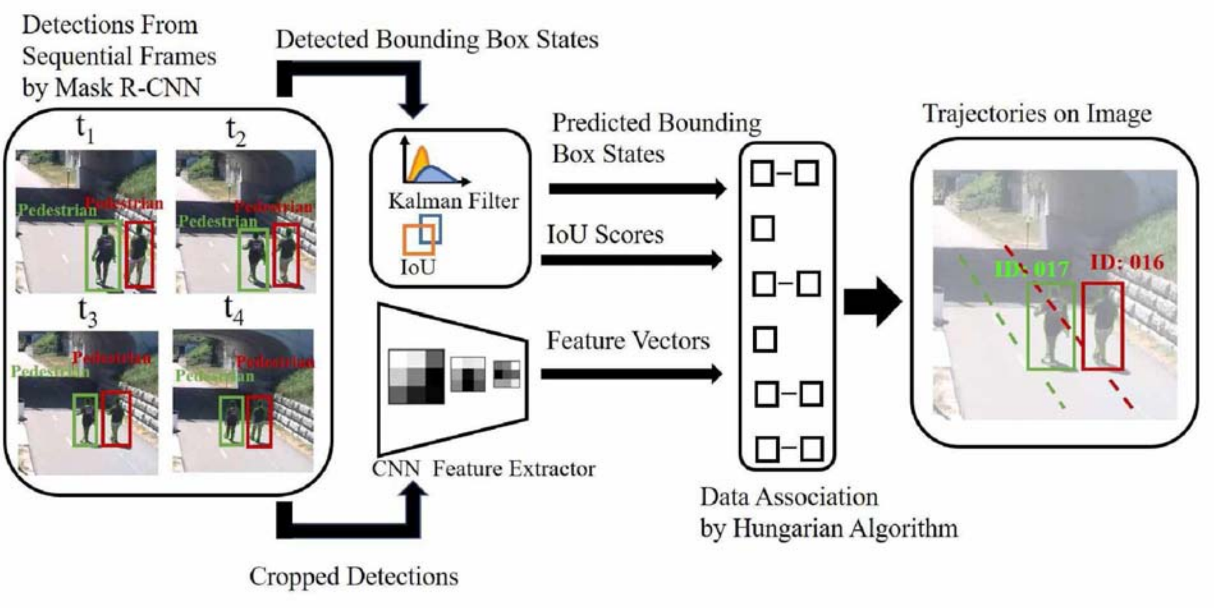

Detections From Sequential Frames by Mask R-CNN
Detected Bounding Box States
t1
t2
t3
t4
Pedestrian
Pedestrian
Kalman Filter
IoU
Predicted Bounding Box States
IoU Scores
Feature Vectors
CNN Feature Extractor
Trajectories on Image
ID: 017
ID: 016
Data Association by Hungarian Algorithm
Cropped Detections

Fig. 3: Schematic of the tracking module using detection results from sequential frames.

calculating the dot product of the two vectors and dividing by the product of the magnitudes of the vectors. IoU scores for detection bounding boxes in the current frame and detection bounding boxes in the previous frame are also computed. The IoU between two bounding boxes is computed by taking the area of the intersection and dividing by the area of the union.

Additionally, for each detection, a track is created that has state $X = \{x_c, y_c, a, h, \dot{x_c}, \dot{y_c}, \dot{a}, \dot{h}\}$, describing the bounding box center coordinates, aspect ratio, height, and time derivatives, respectively. A Kalman filter [22] is used to track detected object states as a dynamical system and recursively generates the current state (X_k) on the k-th video frame using the previous state (X_{k-1}). Predicted states of tracks from the Kalman filter are compared against detections in the current frame by computing the squared Mahalanobis distances. The Mahalanobis distance calculates the distance between a vector (*e.g.*, state of current detection) and a distribution (*e.g.*, predicted state from the Kalman filter).

The Hungarian algorithm uses cosine similarities between appearance vectors, IoU scores, and squared Mahalonobis distances for data association and ID assignment. By leveraging motion information and appearance information, the tracker is more resilient to issues caused by occlusion. Additionally, the Deep Sort algorithm is tuned with parameters controlling when to delete tracks that have not had detection matches for consecutive frames and how many matches a track must have before an ID is assigned.

C. Face mask detection module

In this framework, Mask R-CNN handles person detection and will forward the cropped images of detections to the face mask detection module. Therefore, the face mask detector only needs to solve a binary classification problem: is the detected person wearing a face mask or not? Recent work [5], [6] shows CNN-based detectors are able to detect facial occlusions with impressive accuracy when tested on face mask datasets such as MAFA. The datsets used for training and validation contain clear, high resolution images of faces. Unfortunately, CNN-based face mask detectors that have been trained on higher resolution images with close-up views of faces will struggle when applied to lower resolution crops of faces taken from distant cameras. To address this issue, a low resolution face mask dataset was manually curated from over 50 hours of surveillance camera footage from the Detroit riverfront. The dataset is intgrated into the authors' Objects in Public Open Spaces (OPOS) [7] dataset and is titled OPOS-FM. OPOS-FM contains 6,039 images of cropped faces. The cropped images are on average 3200 px^2 with the defining feature (face mask) typically between 100-400 px^2.

The face mask detector is a CNN-based binary classifier which utilizes a wide residual network [23] architecture. The face mask detection module uses the bounding boxes and tracking information from tracked people. If the velocities from a track state vector indicate a tracked person is headed towards the camera, the bounding box coordinates are used to forward a cropped image of the tracked person to the face mask detection tool. The top 6th of the cropped image is further cropped to isolate the head area. The wide residual network extracts a feature vector from the cropped head through a series of convolutional layers and then follows with a final classification layer; this is illustrated in Fig. 4. Face mask classification results are associated with the unique person IDs generated from the tracker, so the number of *unique* masks and total usage rate can be calculated. If a tracked person never faces the camera, the face mask classification associated with the ID is left as "NA", and the ID is not included in face mask

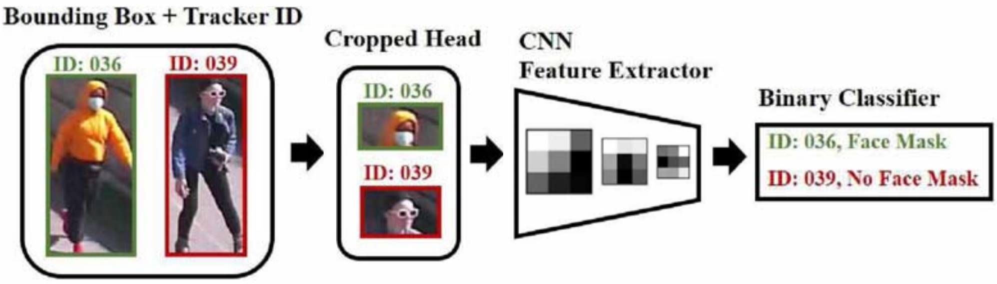


Fig. 4: Schematic of face mask detection process using a CNN-based classifier.

usage calculations.

D. Cloud infrastructure for web-based visualization

Cloud infrastructure, powered through AWS, is implemented to enable real-time data visualizations which can be made accessible online to community members and stakeholders. The IDs of tracked people, their activity classifications (*e.g.*, pedestrian, cyclist, sitter), and a marker denoting if they are wearing a face mask is stored in the AWS S3 object storage service. The data on AWS S3 is then visualized on an interactive web application which is hosted through AWS Elastic Beanstalk which is an easy to use service for deploying and scaling web services and applications. The dashboard is coded in Python with Dash [24], an open-source productive Python framework for building web applications. Dash is an attractive solution for developing interactive dashboards as it allows for easy integration of powerful Python data visualization packages.

III. Results

A. Person detection

The Mask R-CNN model used for person detection was trained and validated with the OPOS dataset. The OPOS dataset was curated from 18 different surveillance cameras and contains 7,826 images of scenes of people in public open spaces. The dataset includes over 17,000 annotated object bounding boxes and segmentations. Detailed performance metrics of the Mask R-CNN detector evaluated on OPOS are presented in [25]. Average precision (AP) scores for each person activity class is presented in Table I. The popular AP50 metric is used and measures the precision of the detector on different classes when an IoU over 0.5 between ground truth bounding boxes (manual annotations) and detected bounding boxes is considered a true detection. As is evident in Table I, AP50 scores for the major OPOS person classes is 89% or higher.

B. Face mask detection

The face mask detector was trained and validated on the OPOS-FM dataset. OPOS-FM contains 2,015 images of faces with face masks and 4,024 images of faces without face masks. 5,133 images were used for training while 906 images were withheld for validation. The validation subset contains 306 faces with face masks and 600 faces without face masks. When tested on the validation subset, the detector correctly classified 296 out of the 306 with face masks and correctly classified 574 out of the 600 without face masks, achieving an overall accuracy of 96%.

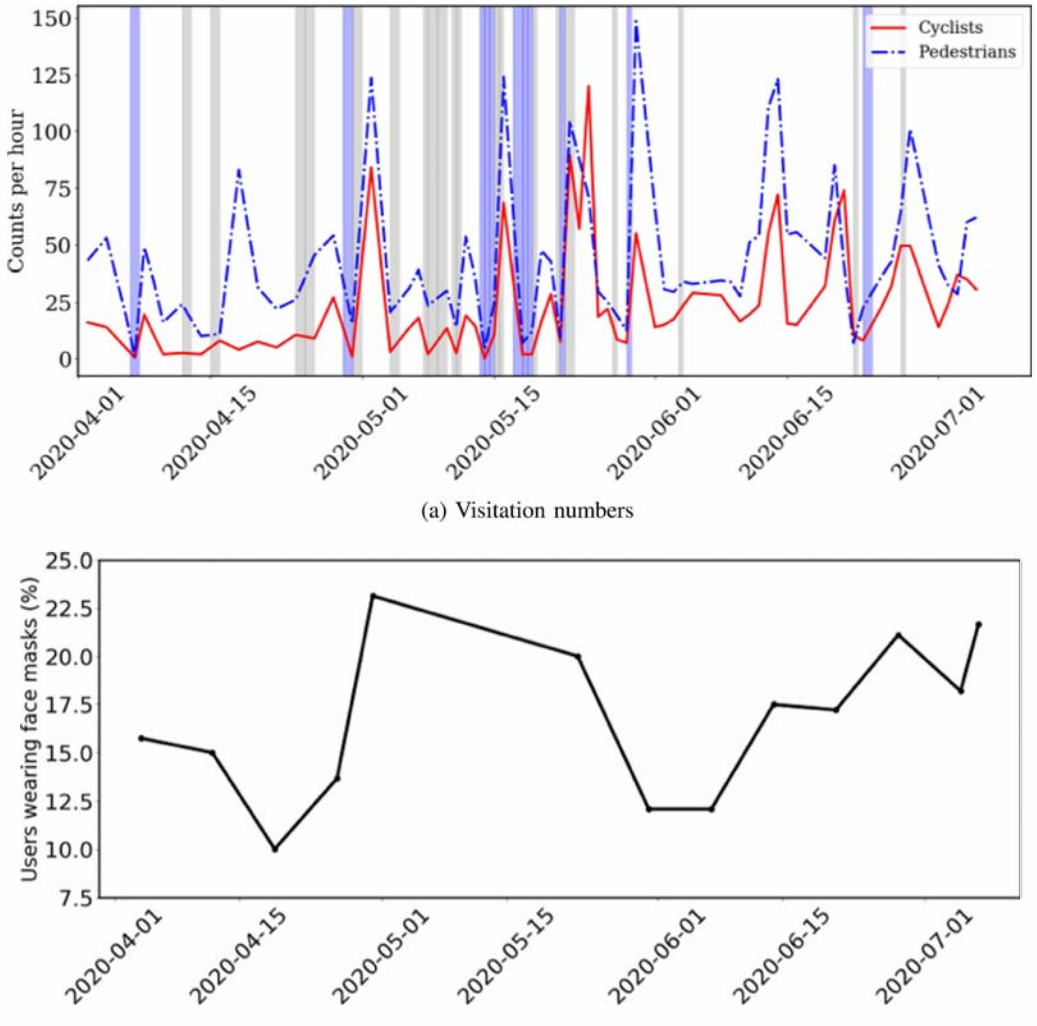
TABLE I: Performance of Mask R-CNN person detector evaluated on OPOS testing dataset

Person class	Description	AP50 (%)
Pedestrian	Person standing, walking, or running	96.36
Cyclist	Person riding bicycle	96.50
Scooter	Person riding scooter	89.39
Skater	Person skateboarding	89.52
Sitter	Person sitting on bench or ground	89.14
People–Other	Truncated image	74.08

C. Implementation

The framework is implemented on several surveillance cameras at the Detroit RiverWalk, which span approximately 5 kilometers along the Detroit River (Detroit, Michigan). The RiverWalk is a public space that provides pedestrians direct access to rails-to-trails greenways, plazas, pavilions, playgrounds, and open green spaces. The RiverWalk is used by approximately three million visitors annually who come to walk, run, bike, and enjoy the scenery [26]. During the COVID-19 pandemic, the public spaces of the RiverWalk remained open to the public but social distancing and face mask usage was required. The framework proposed in this paper was tested on cameras at the RiverWalk in April 2020 and later fully implemented operating since May 2020. Video footage must first be downloaded from a camera server maintained by a contractor to the park manager. Current bandwidth restrictions on the surveillance camera network limit video processing to 12 hours of footage per day; this daily quota is distributed across three cameras during peak morning (11:00- 13:00) and evening (17:00-19:00) times. Additional hardware is currently being installed to give the framework 24-7 access to real-time camera streams thereby removing current restrictions.

Using data collected between April 1, 2020 and July 8,2020, Fig. 5 shows cyclist and pedestrian visitation numbers (reported on an hourly average) as well face mask usage rates. Most of the spikes in Fig. 5a occur during sunny days while low activity level is usually associated with gloomy or rainy weather conditions. Face mask usage rates (Fig. 5b) hit an



(a) Visitation numbers

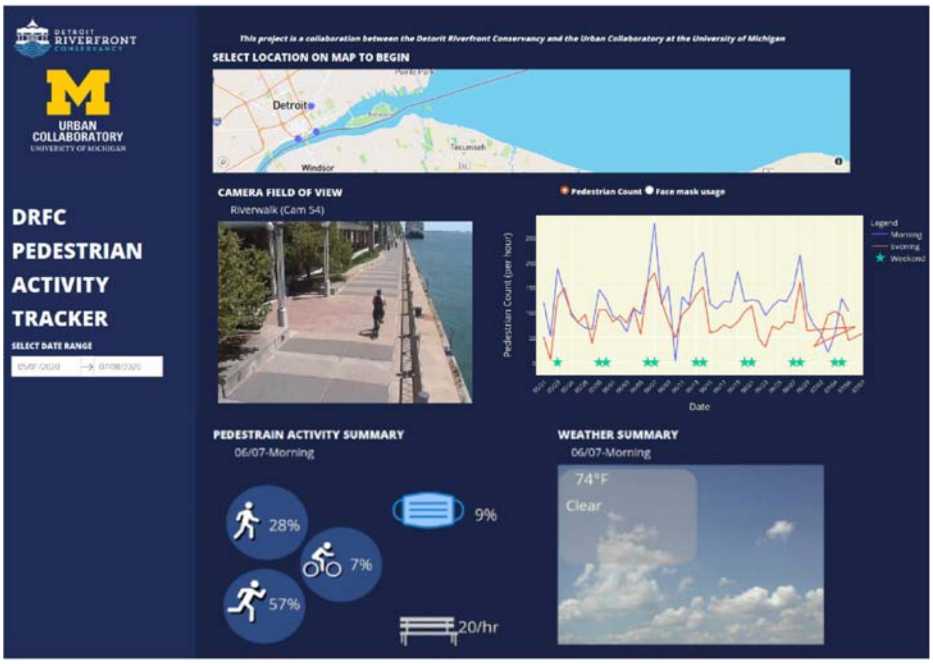
(b) Face mask usage

Fig. 5: Long-time monitoring of (a) patron counts and (b) face mask usage at the Detroit RiverWalk during morning hours (from April 1 to July 8, 2020). The white background represents sunny weather, gray shade represents overcast, and light blue shade represents raining weather.

initial peak in early May, decline in June, and then rise as new COVID-19 cases begin to surge again. All data is stored on AWS S3 and is visualized on an interactive website for park managers and stakeholders. A snapshot of the website is shown in Fig. 6. The website allows users to select a camera from the map (top), choose a date range (left side bar), and see a breakdown of visitation rates and face mask usage trends. If a user hovers the mouse over a specific day they will also see a weather summary as well as a breakdown of visitor activity, such as percentage of cyclists, joggers, or bench users. The website is being actively used by park managers to monitor hygienic practices of patrons and inform effectiveness of tactics used to increase face mask usage. For example, park managers have used the data from the website to identify locations where increased signage is needed to remind patrons to wear face masks.

IV. Conclusion

CNN's can perform person detection and face mask classification with high accuracy. This paper presents the OPOS-FM dataset, a manually curated gallery of faces specifically designed to train CNN-based face mask classifiers for real world applications, along with a computer vision framework which integrates multiple modules to achieve person and face mask detection and tracking. The framework is designed for use in urban environments and leverages AWS cloud infrastructure to equip communities and stakeholders with actionable data. An example of the framework is demonstrated on surveillance cameras along the Detroit riverfront where the data is being used by park managers to make informed decisions regarding COVID-19. With an average precision of



This project is a collaboration between the Detorit Riverfront Conservancy and the Urban Collaboratory at the University of Michigan

Fig. 6: Screenshot of the interactive web application for visualizing data captured from cameras along the Detroit RiverWalk.

89% or higher for person classes and a face mask classification accuracy of 96%, the framework can accurately track number of individuals and face masks present in a camera's field of view. Furthermore, AWS and Dash provide an easy to use environment for development of scalable web applications. The Dash platform is thoroughly documented and users can leverage a plethora of templates for rapid development. However, the tracking capabilities of the framework are currently limited to one camera. The framework can not track people as they move between different fields of view. Additionally, if a person leaves a camera's field of view and returns a few minutes later, the tracker will create a new ID for them and the person will be counted twice. In the future, to address these challenges, a re-identification module will be added to the framework. By doing so, full person trajectories can be mapped across multiple cameras through large public open spaces, and individuals can be re-identified when they return to a camera's field of view. While the framework is currently being used primarily for tracking face mask usage rates, the person tracking and proposed data visualization components can be leveraged in applications in other urban planning studies that extend beyond COVID-19.

V. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1831347. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The authors also wish to acknowledge the generous support provided by the Detroit Riverfront Conservancy especially that offered by Mark Wallace, President & CEO, William Smith, Chief Financial Officer, Rachel Frierson, Director of Programming, and Mac McCracken, Director of Operations & Security. Additional support provided by the University of Michigan Institute of Data Science (MIDAS) also gratefully acknowledged.

References

[1] World Health Organization, "WHO director-general's opening remarks at the media briefing on covid-19-11 march 2020," *Geneva, Switzerland*, 2020. [Online]. Available: https://www.who.int/dg/speeches /detail/who-director-general-s-opening -remarks-at-the-media-briefing-on-covid - 19—11-march-2020

[2] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533 – 534, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1473309920301201

[3] P. Folegatti, "Safety and immunogenicity of the chadox1 ncov-19 vaccine against sars-cov-2: a preliminary report of a phase 1/2, single-blind, randomised controlled trial," *The Lancet*, 2020.

[4] CDC, "Cdc newsroom july 14, 2020 press release," 2020. [Online]. Available: https://www.cdc.gov/media/releases/2020/p0714-americans-to-wear-masks.html

[5] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector," 2020, arXiv:2005.03950.

[6] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2682–2690.

[7] P. Sun, R. Hou, and J. Lynch, "Measuring the utilization of public open spaces by deep learning: a benchmark study at the detroit riverfront," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020, pp. 2228–2237.

[8] T. J. Leeper, *aws.s3: AWS S3 Client Package*, 2020, r package version 0.3.21.

[9] Amazon Web Services. [Online]. Available: https://aws.amazon.com/elasticbeanstalk/

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, 2001, pp. I–I.

[11] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 555–562.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[13] 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*, 2016, pp. 779–788.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[19] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, "Spatially supervised recurrent convolutional neural networks for visual object tracking," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[20] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[21] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," vol. 9910, 10 2016, pp. 868–884.

[22] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[23] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, arXiv:1605.07146.

[24] S. Hossain, "Visualization of Bioinformatics Data with Dash Bio," in *Proceedings of the 18th Python in Science Conference*, C. Calloway, D. Lippa, D. Niederhut, and D. Shupe, Eds., 2019, pp. 126–133.

[25] P. Sun, G. Draughon, and J. Lynch, "'autonomous approach to measuring social distancing and hygienic practices during the covid-19 pandemic in public spaces'," unpublished.

[26] Detroit Riverfront Conservancy, "The Riverfront." [Online]. Available: https://detroitriverfront.org/riverfront