Face Mask Detection on Real-World Webcam Images

Eashan Adhikarla Brian D. Davison
Computer Science and Engineering Dept., Lehigh University
Bethlehem, Pennsylvania, USA
{eaa418|bdd3}@lehigh.edu

ABSTRACT

The COVID-19 pandemic has been one of the biggest health crises in recent memory. According to leading scientists, face masks and maintaining six feet of social distancing are the most substantial protections to limit the virus's spread. Experimental data on face mask usage in the US is limited and has not been studied in scale. Thus, an understanding of population compliance with mask recommendations may be helpful in current and future pandemic situations. Knowledge of mask usage will help researchers answer many questions about the spread in various regions. One way to understand mask usage is by monitoring human behavior through publicly available webcams. Recently, researchers have come up with abundant research on face mask detection and recognition but their experiments are performed on datasets that do not reflect real-world complexity. In this paper, we propose a new webcam-based real-world face-mask detection dataset of over 1TB of images collected across different regions of the United States, and we implement state-of-the-art object detection algorithms to understand their effectiveness in such a real-world application.

CCS CONCEPTS

• **Machine Learning**; • **Computer Vision**; • **Neural Networks**;

KEYWORDS

COVID-19, Face-Mask Detection, Real-World Dataset, Webcam Images

ACM Reference Format:
Eashan Adhikarla Brian D. Davison. 2021. Face Mask Detection on Real-World Webcam Images. In *Conference on Information Technology for Social Good (GoodIT '21), September 9–11, 2021, Roma, Italy.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3462203.3475903

1 INTRODUCTION

The COVID-19 pandemic emerged in December 2019 in Wuhan City in the Hubei province of central China. Observing the virus's growth and spread among humans, the World Health Organization declared the coronavirus (i.e., Sars-CoV-2) to be a world pandemic in March 2020. Researchers experimentally found that wearing a face mask and maintaining social distancing will help slow the spread of the coronavirus [19]. As a result, various health departments across the world started issuing guidelines for their respective countries.

It is crucial to observe face mask usage across various regions to adequately provide information to policymakers and epidemiologists who project the progress of the outbreak. It is also essential to understand the future upcoming waves of COVID-19 and any other such new viruses with the help of face mask usage activity, and to have an early warning indicator so that society can act upon it. As a result of COVID-19, the need has arisen to develop an efficient face mask detection algorithm to track face mask usage in populated areas. Numerous research publications have attempted to do so with strategies ranging from single-stage detection to multi-stage detection. COVID-19 motivates current research because it demonstrated that existing object and face detection algorithms do not perform well for face detection tasks as it becomes more difficult for the algorithms to detect faces with partial facial visibility (as we will show in Table 2). It is also a significant problem from a dataset standpoint as there are few historical datasets available for the face mask detection problem. Current dataset developments focus on adding synthetic mask images over actual face images using facial landmark annotations to infer the location of facial structures [2, 24, 27] or images that have real masked faces but are relatively easy to detect and close to the cameras. The difference in complexity is highlighted in Figure 1.

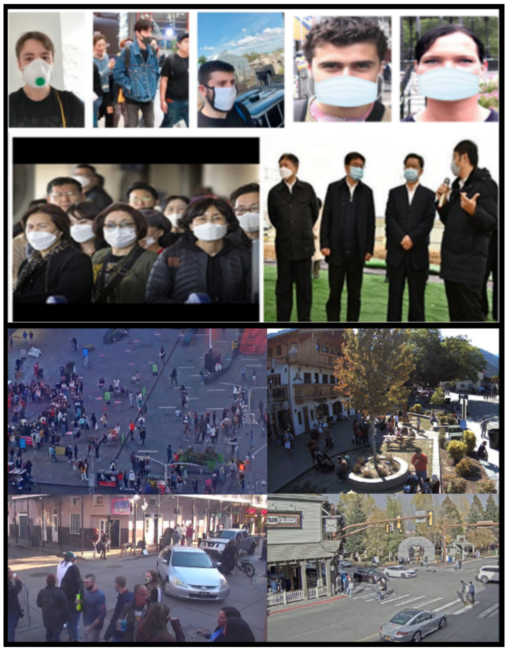

Figure 1: Comparing the face mask images that are widely being used for face detection (top half) versus webcam face-mask images (bottom half).

Moreover, existing real masked face datasets cover some specific regions of the world that could lead to models that would be problematic to deploy elsewhere. Xiong et al. [26] proposed a dataset of 360K images but pulled from a population that is heavily biased

GoodIT '21, September 9–11, 2021, Roma, Italy
2021. ACM ISBN 978-1-4503-8478-0/21/09...$15.00
https://doi.org/10.1145/3462203.3475903

toward Asians. However, current face mask detectors trained over current datasets are scarce and would still need improvements for complicated real-world tasks. This paper answers how well the current models work in real-world public webcam image datasets.

One way to determine face mask usage without further spreading the virus is to observe the publicly available webcams in bulk and examine the faces for masks. The approach is scalable, safe to execute, and provides a bigger picture of face mask usage in the United States. An example of Times Square in New York is shown in Figure 2. We gathered approximately 1016 Gigabytes of data from 74 webcams covering a variety of locations across the United States of America. Our contributions are as follows:

- The paper presents a novel dataset with much higher difficulty than existing Wider-Face hard category and COCO datasets.
- We have collected 1016 Gigabytes of data (i.e., 653,997 images) from public US webcams from June 2020 to March 2021.
- We re-implemented state-of-the-art face detection algorithms for face mask detection and demonstrate their effectiveness against real-world webcam images.
- We estimate mask usage for the United States over time using multiple state-of-the-art face mask detection algorithms.

We make available our source code and dataset with annotations at: **github.com/eashanadhikarla/wfm**.

2 RELATED WORK

Face detection is a vital and primary step for face mask detection, face recognition, masked face recognition, head-pose detection, and numerous other applications related to faces. Thus typically face mask detection algorithms are divided in two tasks; (i) Detecting the faces in a given image, and (ii) then classifying the image as a masked or no-masked face that is a binary classification task. We define an additional third class to reflect uncertainty or when the mask is not worn properly. This work is a form of image classification and feature extraction. For such tasks deep learning has proven to be an incredibly powerful tool. Deep learning algorithms help in extracting the most relevant features from the images for image classification tasks. A lot of study has been accomplished in the area of face detection and recognition [23], object detection [20], analyzing road traffic via street object detection [25].

Recently, numerous Face Detection algorithms have been specifically designed for face mask detection. Loey et al. [15] designed a transfer learning model with ResNet-50 for feature extraction and SVM for classification. Although the method achieved a high accuracy on multiple datasets, the datasets were not complex enough to generalize for all types of complicated real-world applications such as face detection on streets, public areas, etc. For example, one dataset used (LFW) is a benchmark for face recognition and has very clear faces in every image making the detection task easier, whereas our challenge is to find faces in extremely complicated scenarios as shown in Fig. 1. Nagrath et al. [17] proposed a Single-Shot Detector with MobileNetV2 for face mask detection achieving an accuracy of 92.64% on their self-prepared dataset that is a combination of multiple publicly available datasets. Rodríguez et al. [18] designed a method to detect medical masks in an operating room. The combination of two detectors, one for faces and another for medical masks, enhances the models' performance, achieving 95% accuracy in detecting faces with surgical masks. This fairly narrow method would not work in cases with faces more than 5m away from the camera. Loey et. al [14] used a YOLO-v2 with ResNet-50 model and achieved 81% average precision for face mask detection. Overall, most algorithms are using state-of-the-art face detection models for the task of face mask detection. Hence, we plan to understand the state-of-the-art models and widely used pre-trained networks for our analysis.

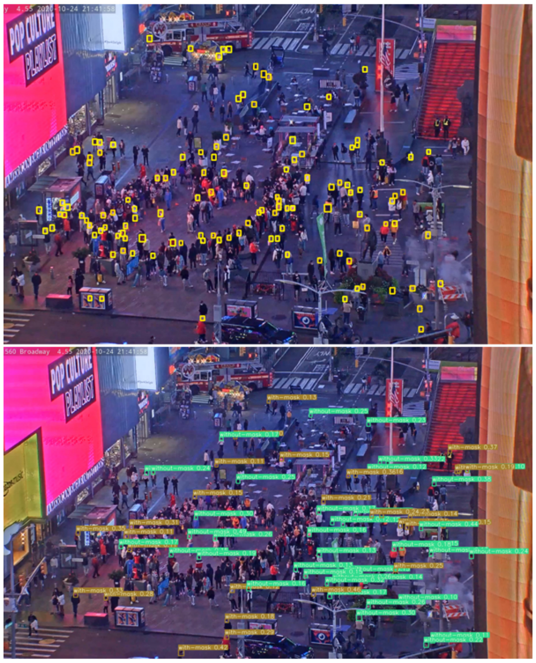

Figure 2: A real-world webcam image of New York City's Times Square reveals the complicated nature of the problem. Top: faces detected using RetinaFace [5]; Bottom: face mask detection using Mask-RCNN [7].

Convolutional Neural Networks. Object recognition is a popular task in computer vision. The purpose of any object detection model is to detect a sub-region in a given image that contains an object, create a bounding box and label the object with a class label. Deep learning algorithms such as Convolutional Neural Networks (CNNs) have shown impressive performance on visual recognition tasks and have a phenomenal ability to extract latent features from images.

Single & Multi-Stage Detectors. A recent work, Retina-Mask [10], uses a two-stage detector where the stage 1 detector extracts the sub-regions of the image that have higher probability of containing objects using the R-CNN [22] architecture. The sub-regions are then passed on to the stage 2 detector that contains an SVM preceded by a CNN to extract features and perform classification. The pipeline takes a resized input which passes through the classification network which serves as a feature extractor. Then it is passed through another classifier that contains an SVM for each class that predicts the object (and a regressor is trained in the background for corrections). The overall idea shows significant performance improvement, but the overall choice of dataset used by the authors is very small, and the evaluation performed on another dataset [4]

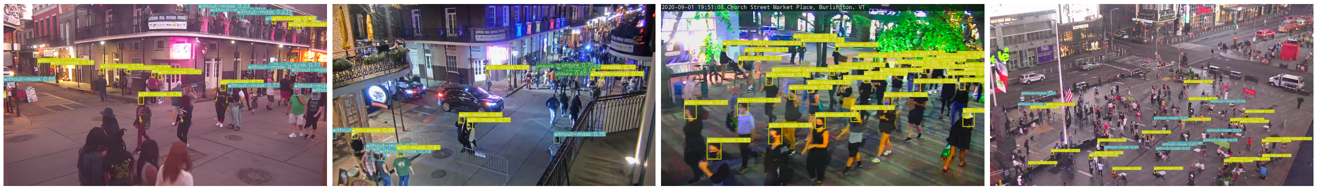
2020-09-01 19:51:08 Church Street Market Place, Burlington, VT

Figure 3: Left-to-right: (a.) Cats and Meows Karaoke Bar, New Orleans, LA, (b.) Bourbon Street, New Orleans, (c.) Church Street Market Place, Burlington, VT (d.) Times Square, New York. Labels are provided by YOLOv5x-TTA [11].

is fairly uncomplicated. Additionally, the authors do not provide performance measures from any real-world dataset.

Unified Neural Networks. A work based on convolution neural networks is the single shot detector [13] which is faster than RetinaMask but performance drops for images with relatively small objects. Another recent work, Unified Multi-scale Deep-CNN [3], uses a unified network for face detection that is trained end-to-end. Unlike a two-stage detector, the unified network uses sub-networks for proposals and detection, and uses a multi-task loss to optimize model training. The method uses less memory by replacing input upsampling by feature upsampling using deconvolution.

3 WEBCAM FACE MASK (WFM) DATASET

An idealized dataset would incorporate 1) significant real-world diversity (e.g., of location, time of day, distance to subjects, and number of subjects); 2) authentic masked images (as opposed to artificial masks over prior images); and, 3) annotations that are compatible with many existing models. Our dataset generally reflects these goals.

Our overall goal was to collect real-world public images over 2020-2021 from different areas in the United States to provide information about mask usage in the population. However, since no prior source of faces in public with masks exists, such a dataset would also have value in training and testing face and mask detection models. Figure 3 demonstrates four different locations with different densities of faces, bounding box sizes, and lighting conditions.

3.1 Finding Webcams

Finding public webcams was a challenge as we needed to select webcams where algorithms can detect faces and hence face masks (i.e., we were only interested in webcams where public gathering was seen in bulk). We identified ~100 webcams from locations across the United States. Out of those webcams we filtered out a quarter that either did not include good views of people and their face masks or were restricted by website policy. Out of the remaining 74 webcams, 12 are from YouTube and the rest were from various popular public websites such as Earthcam, Ocean City, IP24.net, Skyline Webcams, and ipcamlive.

We selected sources with image sizes ranging from 480 to 2048 pixels in both dimensions, after manual inspection of each webcam to ensure good data quality for training and analysis. Higher priority was given to images with better quality as they would offer more detailed features about the location. Hence, we use algorithms that support different image sizes; for example, YOLO-v5x is good in capturing image details using the CNN as feature extractor. Low resolution images work well for webcams with a comparatively less complex background in the images and faces that appear to be much closer to the cameras.

3.2 Dataset Characteristics

We periodically retrieved image frames from our set of live webcams. The average period between fetched images from a single webcam was between 400 and 500 seconds (i.e., 6.67-8.34 minutes). Due to the long span of image collection, many webcams shut down after a while and we added a few more later. Some images were captured in greyscale; the vast majority were in color. Image sizes ranged from 480 to 2048 pixels wide and which were subsequently resized as needed for the algorithm. The resizing was performed uniformly to preserve the ground truth bounding-box aspect ratios. In total, 653,997 images were collected, covering a wide variety of images with a range of facial bounding-box sizes (from 1% to 60% of the image width) and density of bounding boxes (from 1 to 83 in an image). These images also offer diversity in terms of illumination and capture all types of weather conditions from June 2020 to March 2021. Figure 4 shows scatterplots reflecting the distribution of x vs. y coordinates (i.e., the lower-left corner of where the face appears) and width vs. height of the bounding boxes. Annotations were recorded in multiple formats (xml, txt, normalized coordinates).

3.3 Data Labeling & Preprocessing

Since the data is collected without labels, we labeled a portion for model training and comparison. We used the COCO Annotator [1] to manually label images. First, we applied a pre-trained YOLOv5x6TTA model over the dataset to filter out any images with no faces detected (which naturally eliminates corrupted images and useless, e.g., rain-smeared images) from the manually labeled dataset. We randomly chose ~2500 images for the labeling task. The selection process is constrained to select images at least 60 minutes apart if they are from the same source to prevent similar

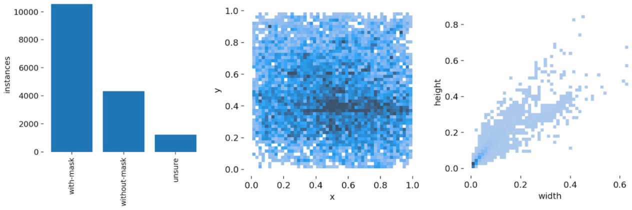

Figure 4: Characteristics of the hand-labeled portion of the dataset. Middle figure (x vs. y) and right figure (width vs. height) are normalized coordinate values between 0 and 1.

Table 1: Performance characteristics of all object detection models.

Model	Size	Model	Inference	Precision	Recall	AP	AP	AP	mAP
	(pixels)	size (mb)	time (ms)	(P)	(R)	(mask)	(no-mask)	(unsure)	@.5
YOLOv3-tiny	640	17.4	5.4	21.4	25.2	17.7	23.0	3.94	14.9
YOLOv3	640	123.4	13.8	27.5	36.6	35.2	36.6	8.44	26.8
Faster-RCNN	600	329.7	54.25	27.7	39.2	34.8	38.6	10.9	28.1
YOLOv3-SPP	640	125.5	13.2	27.9	38.7	35.7	37	10.7	27.8
RetinaNet	800	155	24.8	32.2	35.2	36.9	39.8	10.1	29
Mask-RCNN	600	176.3	237.8	33.1	36.8	39.7	42.2	11.3	31
YOLOv5x	640	175.1	36.1	36.1	33.5	37.1	40.2	10.3	29.2
YOLOv5x	1280	1055	35.7	32.6	37.7	42.8	46.7	11.7	33.8
YOLOv5x6TTA	1280	1130	39.2	37.0	41.6	46.5	47.4	11.2	35.1

images from being selected for labelling. In total, there were two annotators per image for the labeling of the first ~300 images which were found to be highly consistent. The remaining annotations were performed by a single annotator per image.

4 EXPERIMENTAL WORK

4.1 System Overview

Our overall implementation is comprised of eight object detection and four face detection models. The object detection methods Mask-RCNN [7], YOLO-v3 (Darknet-53) [20], YOLO-v3-tiny [20], YOLO-v3-SPP (Spatial Pyramid Pooling) [9], RetinaNet [12], Faster-RCNN [21], YOLO-v5x [11], and YOLO-v5x6-TTA (Test Time Augmentation) [11] are widely popular object detection models and we specifically fine-tune them for face mask detection. Object detectors in our case are trained specifically for faces as objects that wear masks and do not wear masks. On the other hand, we were also interested to understand the efficacy of face detection algorithms namely: Tinyface [8], RetinaFace [5], FaceBoxes [29], MTCNN [28]. For the object detectors, given an image, the task is to detect faces in the image and classify the detected boxes as into three labels: **with-mask**, **without-mask**, and **unsure**). Whereas, for face detection algorithms, we measure their face detection efficacy.

YOLO stands for "You Only Look Once" and uses CNNs throughout, unlike other two-stage detectors for object detection. On a given image YOLO detects multiple objects using extracted features, creates bounding boxes around them using box-regression and assigns a label using object classification. There are various versions of the YOLO architecture; we use three types of YOLO-v3 architectures as they are widely used in different applications, and they improve upon their predecessors. Generally, YOLOs are popular mainly because they are fast despite having 106 layers in the network, and they are capable of generalizing well to datasets with slight distributional shifts versus that of the training set [20]. YOLO-v3 contains a dense network, including 75 layers of convolutional-2D, 31 layers of shortcut, routes, and up-samplings, resulting in reduced localization errors and significantly improved detection and classification accuracy.

Faster-RCNN uses region proposal networks (RPNs) to generate anchor boxes that are filtered based on a top-k threshold value, and compute the box regression and classification. A RPN predicts whether an anchor will be in the background or foreground and refines the anchor accordingly. In addition to the existing branch for classification and bounding box regression in Faster-RCNN, Mask R-CNN adds a branch for predicting segmentation masks on each Region of Interest (RoI).

4.2 Intersection Over Union (IOU)

In all cases, to measure the bounding-box accuracy we use the Intersection over Union (IOU) of the predicted and ground-truth bounding boxes as an evaluation metric. The IOU metric is independent of the algorithm used for predicting the bounding boxes. IOU is computed as *area of intersection* divided by *area of union*. In order to compute the areas, we need the ground-truth bounding boxes (i.e., where the object is actually located in the image) and the predicted bounding box that the algorithm generates as output. IOU values are always greater than 0 and less than or equal to 1.

4.3 Non-Maximum Suppression and Confidence Threshold

Typically, when identifying faces in an image, we use a sliding window to compute the feature map and provide scores to each window. However, because we have too many candidate region proposals, we filter them using a **non-maximum suppression threshold** (NMS) which utilizes intersection over union as described in Section 4.2. It takes input two boxes and computes the intersection and the union of the two boxes and computes the ratio. Once we have the intersection over union values, following are the steps to evaluate over NMS threshold;

(1) We define a list K of proposed bounding boxes and sort them by confidence score in decreasing order.
(2) Take the proposal with the highest confidence score in list K, remove it from K and add it to an empty list X.
(3) For each remaining proposal i in list K, calculate IOU between i and the most recently added proposal in X. If IOU is greater than a defined threshold (0.5 in our case), then remove i from K.
(4) Now, we have removed all the proposals that are similar to the proposal in X, thus any proposals remaining in K would refer to a different object. So, repeat steps 2 - 4, with the most recently added proposal in X.

4.4 Training, Testing, and Evaluation

For training the model, the training and testing dataset is structured differently for each algorithm we used. Ultimately, all models perform a multi-class object detection task where we provide three

Table 2: Performance of the WIDER-Face pre-trained face detection models on the webcam dataset (average precision at different IOU thresholds).

Model	$AP_{.10}$	$AP_{.20}$	$AP_{.25}$	$AP_{.30}$	$AP_{.50}$	$AP_{.75}$
MTCNN	20.1%	18.4%	17.3%	16.1%	8.5%	0.3%
Faceboxes	40.3%	39.8%	39.7%	39.4%	37.1%	9.2%
Tinyface	65.5%	63.1%	60.3%	56.5%	34.2%	3.7%
Retinaface	84.1%	80.3%	75.7%	69.4%	43.8%	7.9%

different labels for training and evaluation (**with-mask**, **without-mask**, and **unsure**).

We divide the hand-labeled ~2500 image dataset with 3-fold cross validation. We resize the images to 640, 1024, 1280 pixels (height and width).[1] We used a stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and a decay rate of 5e-4 for all the algorithms except Faster-RCNN and Mask-RCNN with a decay rate of 1e-4. To augment our hand-labeled dataset, we use ~1530 labeled images from Kaggle [16, 24] for model training and use pre-trained models for weight initialization.

Table 1 shows the state-of-the-art object detection algorithms' performance by computing the average precision as the accuracy measure. We show the average precision with the IOU box-retention threshold of 0.50 for each class, i.e., with-mask, without-mask, unsure, and calculate the mAP (mean average precision) averaging all three classes. We also show the models' precision, recall for each class, model size, input size, and inference time of the model on our dataset. The AP of the unsure class is consistently low for all the architectures, making the overall-mAP lower. It can be inferred from Figure 4 as we see the distribution of the imbalanced classes. It is likely normal to have a low number of "unsure" class labels as hand-labeling is performed mainly towards detecting people with mask and without a mask. However, several conditions such as pose, lighting condition at a specific part of the image, or have obstructions such as hand on the face made it harder to label it. Hence, it was labeled "unsure".

We used various popular models ranging from a model size of 17.4MB, representing a lightweight model that can be easily installed in an embedded device (YOLOv30tiny), to 1130MB, representing a very deep model (YOLOv5x6TTA). We show how different model architectures perform on a dataset containing many highly complex images. For all models, instead of random model initialization, we use the pre-trained models as initial weights for our model for faster convergence, corresponding to a type of transfer learning. It is interesting to observe YOLOv3-tiny. Despite having only seven convolutional layers and six max-pooling layers in the backbone, with a mean-average precision of 14.9%. Furthermore, we see a ~11% improvement with YOLOv3 Darknet-53 network, which has 106 layers achieving 26.8% mean-average precision.

On the other hand, RetinaNet with a ResNet-50 backbone has comparatively similar performance to YOLOv3-SPP. We follow a similar configuration for the convolutional layers by adding batch-normalization after every convolutional layer. It helped in faster convergence of the model and improved the mAP score by 1.2%. Batch-normalization acts as a regularization technique that prevents the model from over-fitting on the training dataset. We use *upsampling* = 2, which upsamples the output feature map from the previous layer by a factor of 2 by bi-linear upsampling. Moreover, RetinaNet uses focal loss as a novel approach to address the foreground-background imbalance in an image. RetinaNet is a single-stage detector, unlike Faster-RCNN and Mask-RCNN that are two stage detectors. YOLO-v5 with Test time augmentation performs the best on our dataset with mAP=35.1% in Table 1.

4.5 Mask-wearing trends

Figure 5 shows the comparison between face mask usage trends analyzed by the different mediums of data collection. The upper (purple) line reflects data collected via surveillance, hospital data, personal interview, etc. This dataset [6] from compiles multiple resources (e.g., self-assessment surveys) with uncertain veracity (e.g., it is not certain whether a subject told the truth about his/her face mask usage). On the other hand, the lower three lines shows our face mask dataset from an objective source, i.e., our Webcam Face Mask (WFM) dataset, but utilizing different detection models. We applied our model across ten months of the dataset (the same ten months as for the self-assessment data) from June 23, 2020 - Mar 31, 2021. Due to training the model for face detection and face mask classification, the model is slightly biased towards detecting non-masked faces compared to masked faces. Overall, we do see a similar trend line for face mask usage using our webcam dataset. We consider two other impacts on the model below in Section 4.6. In general, the overall trend in all cases shows an increase in face mask usage over time.

4.6 Sensitivity Testing

We are interested in knowing whether the number of faces in an image and the size of bounding boxes in the webcam dataset will impact the face mask detection models' efficacy. We use YOLOv5x6TTA

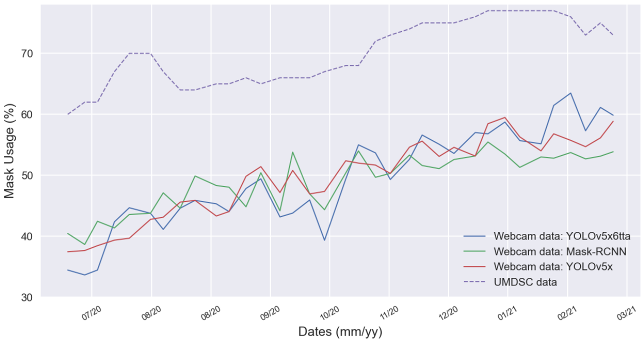

Figure 5: Comparison between face mask usage trends. While the overall rates differ with different datasets, the trends are quite similar. Purple dashed: data from self-assessment surveys [6], and (Blue, Green, Red): dataset evaluation using face mask detection using the top three models (YOLOv5x6-TTA, Mask-RCNN, YOLOv5x).

[1] Mask-RCNN, Faster-RCNN, and RetinaNet keep the aspect ratio and pad with zeros if the image is not square; the remaining models square the image as part of their preprocessing.

for hypothesis testing as it performed best in Section 4.4.

Hypothesis 1. Testing the impact of the *size* of the bounding boxes. We examined 500 images, divided into images with small (120 images) or large bounding boxes (380 images), and compared the performance on each set (0.197 and 0.377 on small and large, respectively). Using a two-proportion z-test to determine whether the performance of the two sets differs, we found that $p << .001$, allowing us to reject H_0 (which said that they were the same).

Hypothesis 2. Testing the impact of the *density* of bounding boxes. Similarly, we divided 500 images into two sets, with more than 4 people, or four or fewer people. Performance on images with $n > 4$ was .286 (averaged across 165 images) and with $n <= 4$ was .327 (averaged across 335 images). A two-proportion z-test finds that $p = .353$, i.e., we cannot reject H_0, and conclude that although we see decrease in average precision with higher density but it is not significantly impacting the performance.

5 CONCLUSION

We presented a new webcam-based dataset that reflects real-world complexity. We tested 12 different models to understand their efficacy. We also utilized three models to label the remaining data to compare predicted mask usage trends and with another source of data. The WFM dataset is valuable for potential COVID-19 related studies and offers diversity for AI-related datasets as this is the first webcam dataset with face masks that has been collected. The dataset provides a real-world challenge for developing better AI models, incorporating real-world face masks for face detection and face mask detection tasks, and is a collection of 10 months of captured images, a small portion of which has been hand-labeled.

ACKNOWLEDGMENTS

We acknowledge Dr. Khushboo Patel for contributions in data labelling of webcam images. We also thank Scott Grenestedt and Youshan Zhang for valuable advice and the members of the Community Resilience Research Group at Lehigh University for their motivation and insights. This work was supported in part by Collaborative Opportunity Research Grants from Lehigh University ("Perceptions shaping reality in disaster recovery"; "Perceptions shaping reality v2.0: Tracking perceptions and responses to the COVID-19 pandemic").

REFERENCES

[1] Justin Brooks. 2019. COCO Annotator. https://github.com/jsbroks/coco-annotator/.

[2] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi. 2021. MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health* 19 (Mar 2021), 100144. https://doi.org/10.1016/j.smhl.2020.100144

[3] Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, and Nuno Vasconcelos. 2016. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In *Computer Vision – ECCV 2016*. Springer International Publishing, Cham, 354–370.

[4] Daniell Chiang. 2020. Detect faces and determine whether people are wearing mask. https://github.com/AIZOOTech/FaceMaskDetection

[5] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. 2020. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5202–5211. https://doi.org/10.1109/CVPR42600.2020.00525

[6] Institute for Health Metrics and Evluation. 2021. *COVID-19 Projections*. Retrieved April 27, 2021 from https://covid19.healthdata.org/united-states-of-america?view=total-deaths&tab=trend

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2980–2988. https://doi.org/10.1109/ICCV.2017.322

[8] Peiyun Hu and Deva Ramanan. 2017. Finding Tiny Faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 951–959.

[9] Zhanchao Huang, Jianlin Wang, Xuesong Fu, Tao Yu, Yongqi Guo, and Rutong Wang. 2020. DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection. *Information Sciences* 522 (2020), 241–258.

[10] Mingjie Jiang, Xinqi Fan, and Hong Yan. 2020. RetinaMask: A Face Mask detector. arXiv:2005.03950 [cs.CV]

[11] Glenn Jocher. 2021. *YOLOv5*. Retrieved April 11, 2021 from https://github.com/ultralytics/yolov5/tree/v5.0

[12] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2980–2988.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*. Springer, 21–37.

[14] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. 2021. Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustainable Cities and Society* 65 (2021), 102600. https://doi.org/10.1016/j.scs.2020.102600

[15] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. 2021. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* 167 (2021), 108288. https://doi.org/10.1016/j.measurement.2020.108288

[16] MakeML. 2020. Mask Dataset. https://makeml.app/datasets/mask

[17] Preeti Nagrath, Rachna Jain, Agam Madan, Rohan Arora, Piyush Kataria, and Jude Hemanth. 2021. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society* 66 (2021), 102692. https://doi.org/10.1016/j.scs.2020.102692

[18] A. Nieto-Rodríguez, M. Mucientes, and V. M. Brea. 2015. System for Medical Mask Detection in the Operating Room Through Facial Attributes. In *Pattern Recognition and Image Analysis*, Roberto Paredes, Jaime S. Cardoso, and Xosé M. Pardo (Eds.). Springer International Publishing, Cham, 138–145.

[19] World Health Organization. 2020. *Infection prevention and control during health care when novel coronavirus (nCoV) infection is suspected: interim guidance, 25 January 2020*. World Health Organization. 5 pages.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. https://doi.org/10.1109/CVPR.2016.91

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[22] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 6 (2016), 1137–1149.

[23] F. Schroff, D. Kalenichenko, and J. Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 815–823. https://doi.org/10.1109/CVPR.2015.7298682

[24] Zhongyuan Wang, Guangcheng Wang, Baojin Huang, Zhangyang Xiong, Qi Hong, Hao Wu, Peng Yi, Kui Jiang, Nanxi Wang, Yingjiao Pei, Heling Chen, Yu Miao, Zhibing Huang, and Jinbi Liang. 2020. Masked Face Recognition Dataset and Application. *CoRR* abs/2003.09093 (2020). arXiv:2003.09093 https://arxiv.org/abs/2003.09093

[25] Y. Wei, N. Song, L. Ke, M. Chang, and S. Lyu. 2017. Street object detection / tracking for AI city traffic analysis. In *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 1–5. https://doi.org/10.1109/UIC-ATC.2017.8397669

[26] Zhangyang Xiong, Zhongyuan Wang, Changqing Du, Rong Zhu, J. Xiao, and T. Lu. 2018. An Asian Face Dataset and How Race Influences Face Recognition. In *Pacific Rim Conference on Multimedia*. 372–383.

[27] Valeriia Koriukina (xperience.ai). 2020. Using Facial Landmarks for Overlaying Faces with Masks: Learn OpenCV. https://learnopencv.com/using-facial-landmarks-for-overlaying-faces-with-masks/

[28] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters* 23, 10 (2016), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342

[29] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. 2017. Faceboxes: A CPU Real-time Face Detector with High Accuracy. In *IEEE International Joint Conference on Biometrics (IJCB)*. 1–9.