Physical Distancing and Mask Wearing Behavior Dataset Generator from CCTV Footages Using YOLOv8

Roland P. Abao(✉), Maria Regina Justina E. Estuar, and Patricia Angela R. Abu

Department of Information System and Computer Science, Ateneo de Manila University, Quezon City, Philippines
roland.p.abao@obf.ateneo.edu, {restuar,pabu}@ateneo.edu

Abstract. Computer simulations using agent-based approach aimed at modeling human behavior require a robust dataset derived from actual observation to serve as ground truth. This paper details an approach for developing a movement behavior dataset generator from CCTV footages with respect to two health-related behaviors: face mask wearing and physical distancing, while addressing the privacy concerns of confidential CCTV data. A two-stage YOLOv8-based cascaded approach was implemented for object tracking and detection. The first stage involves tracking of individuals in the video feed to determine physical distancing behavior using the pre-trained YOLOv8 xLarge model paired with BotSORT multi-object tracker and OpenCV Perspective-n-Point pose estimation. The second stage involves determining the mask wearing behavior of the tracked individuals using the best-performing model among the five YOLOv8 models (nano, small, medium, large, and xLarge), each trained for 50 epochs on a custom CCTV dataset. Results show that the custom-trained xLarge model performed the best on the mask detection task with the following metric scores: mAP50 = 0.94; mAP50-95 = 0.63; and F1 = 0.872. The faces of all the tracked individuals are blurred-out in the resulting video frames to preserve the privacy of the CCTV data. Finally, the developed system is able to generate the corresponding mask-distancing behavior dataset and annotated output videos from the input CCTV raw footages.

Keywords: YOLOv8 · dataset generator · CCTV data · physical distancing · mask wearing

1 Background

On the onset of the COVID-19 pandemic, physical distancing and mask wearing were among the non-pharmaceutical health interventions recommended by

Ateneo Social Computing Laboratory and DOST-Engineering Research and Development for Technology (ERDT).

R. Thomson et al. (Eds.): SBP-BRiMS 2023, LNCS 14161, pp. 295–304, 2023.
https://doi.org/10.1007/978-3-031-43129-6_29

public health authorities to slow down the spread of the disease among the population. Today, COVID-19 is no longer considered a public health emergency of international concern [10]. However, for low-middle income countries (LMIC), limited health capacity still necessitates observation of mask wearing and social distancing. Living with the virus entails, first and foremost, having a substantial portion of the population vaccinated, and keeping up with the voluntary preventive health behaviors such as mask wearing and physical distancing specially in enclosed indoor spaces [4].

There is a need to capture public health behavior to develop a model that will be able to predict movement based on compliance or non-compliance to public health standards. Computer simulations aimed at modeling complex human behavior, as well as other domains of interest, require a robust dataset for the development and validation of a simulation model. Specifically, data-driven agent-based models (ABM) [7] requires having actual ground truth data for modeling. Observation and capturing of real world data is used to model and infer behavioral patterns of the agents and other latent processes based on the actual observed data. The reliability of ABMs can only be as good as the assumptions and calibrations performed in developing the model [2].

In the context of modeling physical distancing and mask wearing behavior, a good source of data are the actual behavior captured from closed circuit television (CCTV) cameras installed in informal public places. In this environment, individuals may be influenced by each other to follow or not the preventive health behaviors. Processing of CCTV data to extract relevant information may be accomplished by deep learning-based object detection algorithms such as the various variants of either two-stage detectors (e.g. R-CNN, fast R-CNN, and faster R-CNN) or single stage detectors (e.g. SSD and YOLO) [11]. Among the different object detection algorithms, the YOLO (You Only Look Once) framework has been used more often by researchers for its exceptional balance of fast inference speed and high-accuracy detection, enabling a real-time and accurate identification of objects in the video frames [5,8,11]. Since its inception in 2015, the original YOLO model had undergone numerous modifications, each of which built upon the prior versions to address previous flaws and improve detection performance. As of the time of writing, YOLOv8 [3] released last January of 2023 is the state-of-the-art version of YOLO, which has the highest detection accuracy as compared to its predecessors.

Processing of CCTV data to extract relevant information in relation to the mask wearing and physical distancing behavior, poses some challenges in relation to object detection. First, the object detection model should have the ability to detect the mask wearing status of each person in the video frame, as well as compute for the distance of one person to another. Existing pre-trained object detection models, including the latest YOLOv8 model, are able to detect multiple persons in the frame already with good accuracy. However, modification (fine-tuning) of the model is still required to detect specific use cases, such as the mask and distancing behavior of the person. Secondly, CCTV cameras, which are almost always placed on a fixed elevated location, produce video frames at

a high angle or bird's-eye point-of-view (POV). Existing object detection models specifically designed to detect persons and faces in the frame, however, are usually trained on a datasets (e.g. PASCAL VOC and COCO datasets) mostly composed of images with over-the-shoulder or at the eye-level POV. Detection models trained from eye-level POV datasets may not necessarily perform very well on the actual captured CCTV feeds with high-angle POV because of the POV difference in the training process and the actual use case. Thirdly, CCTV data contains confidential information, prominently the facial features of the person in the frame. Data privacy measures should be carefully implemented to preserve the identity privacy of the individuals present in the resulting video feeds. Accounting for these challenges in extracting behavioral information from confidential CCTV data, this paper describes and presents current findings in the development a movement behavior dataset generator -with respect to physical distancing and mask wearing- from CCTV feeds, while preserving the identity privacy of the individuals in the resulting video frames. Specifically, the study addresses the following research questions:

1. How can the real-world physical distance be computed from the detected individuals in the CCTV footages?
2. How do the different YOLOv8 models (nano, small, medium, large, and xLarge) perform in the face mask detection task of CCTV footage?
3. When processing confidential CCTV data, how can the privacy of the individuals be preserved on the resulting output video frames?

Additionally, the following are the major contributions of the study:

- Five variants of face mask detection models based on YOLOv8 fine-tuned for high-angle point-of-view video frames such as in CCTV footages.
- A behavior dataset generator system employing a two-stage YOLOv8-based cascaded approach of processing CCTV data, whilst preserving the identity privacy of the individuals in the resulting video frames.

2 Methodology

This section describes the procedure in the development of a dataset generator for physical distancing and mask wearing behavior captured from CCTV footages. The first step involves collecting CCTV footages as presented in Sect. 2.1. A two-stage YOLOv8-based cascaded approach for object tracking and detection was then implemented in the study. The first stage, as detailed in Sect. 2.2, involves tracking the position of the individuals in the frame to determine their physical distancing behavior using a pre-trained YOLOv8 model. The second stage of the cascade involves using a custom-trained YOLOv8-based model for detecting the mask wearing behavior of the individuals, as detailed in Sect. 2.3. Finally, Sect. 2.4 presents the method for integrating the two stages -object tracking and mask detection- to generate a behavior dataset, as well as the method to preserve the privacy identity of individuals in the resulting CCTV frames.

2.1 Collecting Data from CCTV Footages

The trial data was obtained from CCTV footages located in informal public places within a university setting. All CCTV cameras were accompanied with signage informing the public of surveillance recording using CCTV camera. A total of 100 recorded footages were obtained from October 1 to 30, 2022. A non-disclosure agreement was executed by the concerned parties prior to receipt of the confidential CCTV data. The selected informal locations included study areas, activity areas, and walkways where individuals have a considerable leeway to practice wearing of face mask and observe physical distance from each other.

2.2 First Stage: Object Tracking of All Individuals in the Frame

In this stage, the study utilized the pre-trained YOLOv8 xLarge model, which has the highest mean average precision score ($mAP_{50-95} = 53.9$) among all the pre-trained YOLOv8 models [3], for detecting the individual person (class = 0) in the CCTV frames. For object tracking, the default Bot-SORT multi-object tracker algorithm [1] built-into the YOLO package was used. The object tracker produces a bounding box and assigns a unique ID for each tracked individual in the frame. The X and Y coordinates of the middle-bottom part of the bounding box, representing the point of the ground where the person is currently standing (or sitting), was used as the anchor point of the individual in the camera frame, which should be converted into the real-world XYZ coordinates (where Z = 0). Translating between the camera frame coordinates and the real-world coordinates was made possible by identifying 6 known 3-dimensional (3d) fixed points in the real-world and pinpointing its corresponding 2-dimensional (2d) points on the on the camera frame. The identified 3d to 2d translation points were applied to OpenCV's perspective-n-point pose estimation algorithm [6] to get the vector rotation and translation matrix. The dot product of the camera matrix properties and the concatenated rotation & translation matrices would result to the image projection matrix and its inverse projection, which can then be used to compute for the desired real-world coordinates given the input camera frame coordinates.

2.3 Second Stage: Mask Detection

For the mask detection stage, a custom mask dataset to fine-tune the pre-trained YOLOv8 models was first created consisting of a total of 500 random snapshots from the gathered CCTV data. All faces in the snapshots were manually annotated with either noMask (class = 0) or withMask (class = 1) using LabelImg, an open-source image annotation tool. The custom dataset was split into a 80%-10%-10% distribution for the train, validation, and test set respectively. The custom mask dataset was then used to produce five YOLOv8-based mask detection models, namely: mask_YOLOv8n (nano), mask_YOLOv8s (small), mask_YOLOv8m (medium), mask_YOLOv8l (large) and mask_YOLOv8x (extra large). The model fine-tuning was implemented with the following parameters:

Algorithm 1: Pseudo-code of the two-stage YOLOv8-based cascaded approach for object tracking and detection in generating a behavior dataset from CCTV data

Data: $CCTV\_footage,\ 3d\_to\_2d\_translation\_pts$
Result: $output\_csv\_file$ /* Behavior dataset CSV file */
$output\_video\_file$ /* Annotated output video file */

$time \leftarrow 0$
$df \leftarrow pandas.DataFrame()$
`/* First model: object tracking of individuals in the frames */`
$model \leftarrow YOLO(yolov8x)$
$frame\_results \leftarrow model.track(CCTV\_footage,\ tracker = botsort)$
for $result$ ***in*** $frame\_results$ **do**
  $realXY \leftarrow \{\}$
  `/* Second model: mask detection of all faces in the frame */`
  $mask\_model \leftarrow YOLO(best\_mask\_model)$
  $mask\_results \leftarrow mask\_model.predict(result.orig\_img)$
  `/* Pixelate all detected faces to preserve identity privacy */`
  $resulting\_frame \leftarrow result.orig\_img.copy()$
  for box_face ***in*** $mask\_results.boxes$ **do**
    $x1, y1, x2, y2 \leftarrow box\_face.xyxy$
    $resulting\_frame[y1 : y2, x1 : x2] \leftarrow pixelate\_face(box\_face.xyxy)$
  end
  `/* Identifying the mask wearing status of the individuals */`
  for box ***in*** $result.boxes$ **do**
    $mask\_stat,\ mask\_conf,\ max\_area \leftarrow null, null, min\_integer$
    for box_face ***in*** $mask\_results.boxes$ **do**
      if $area\_intersection(box.xyxy,\ box\_face.xyxy) > max\_area$ **then**
        $mask\_stat,\ mask\_conf \leftarrow box\_face.cls,\ box\_face.conf$
        $max\_area \leftarrow area\_intersection(box.xyxy,\ box\_face.xyxy)$
      end
    end
    $id \leftarrow box.id$
    $realXY[id] \leftarrow get\_world\_XY(box.xyxy,\ 3d\_to\_2d\_translation\_pts)$
    append $[time, id, box.xyxy, mask\_stat, mask\_conf, realXY[id]]$ **to** df
  end
  `/* Computing for the Euclidean distance of the individuals */`
  for row ***in*** $df[df.time == time].iterrows()$ **do**
    $closest\_dist \leftarrow max\_integer$
    for j ***in*** $realXY$ ***where*** $row.id \neq j$ **do**
      $d \leftarrow math.dist(realXY[row.id],\ realXY[j])$
      if $d \leq closest\_dist$ **then**
        $closest\_dist \leftarrow d$
      end
    end
    annotate $resulting\_frame$ **with** $row.id, row.mask\_stat, closest\_dist$
  end
  $time \leftarrow time + 1$
  write $resulting\_frame$ **to** $output\_video\_file$
end
save df **as** $output\_csv\_file$

epoch of 50, batch size of 8, image size of 640 pixels, SGD as the optimizer, and an initial & final learning rate of 0.01. After the training process, the best-performing mask detection model based on the F1, mAP50, and mAP50-95 metric scores was used as the final model for mask detection in the study.

2.4 Combining the Two Stages and Generating the Behavior Dataset

Generating the desired behavior dataset was achieved by cascading the two YOLOv8-based models: the pre-trained YOLOv8 xLarge object (person) tracking model and the custom-trained YOLOv8-based mask detection model. Both models are able to produce a bounding box enclosing the tracked individuals and detected faces in the CCTV frames respectively. For each bounding box enclosing a person, its corresponding mask wearing behavior is determined by the bounding box of the face with the largest area of intersection with itself (see LINES 14–21 of ALGORITHM 1 for the pseudo-code implementation). For determining the physical distancing behavior, the euclidean distance for each individual relative to its nearest neighbor in the frame was computed with the help of the DIST() function from the python math library (see LINES 26–35 of ALGORITHM 1 for the pseudo-code implementation). All the detected faces in the input frames using the best-performing mask model were pixelated (blurred-out) to a point where the facial features are no longer distinguishable, as denoted in LINES 9–13 of ALGORITHM 1 to preserve the identity privacy of individuals in the resulting video frames. The blurring-out process was carried out by dividing the cropped image of the face into multiple blocks and filling each block with a color value based on the mean of all the pixel colors within the block. Finally, the output behavior dataset for the CCTV footage was generated as a CSV file with the following column information: ***time*** indicating the time stamp of the frame; ***id*** referring to the unique ID of the tracked individual; ***box.xyxy*** conveying the points of the bounding box enclosing the tracked individual in the frame; ***mask_stat*** & ***mask_conf*** indicating the mask wearing behavior and the classification confidence score; and ***real_XY*** referring to the real-world x & y coordinates of the individual.

3 Results and Discussion

The generated custom mask dataset consisting of a total of 500 random snapshots from the CCTV data was split into 80% (train set) - 10% (val set) - and 10% (test set) distribution. The 400 images in the train set have a total of 2,034 instances (39% unmasked & 61% masked) of labeled mask wearing behavior. The validation and test sets have 248 (39% unmasked & 61% masked) and 286 (44% unmasked & 56% masked) instances respectively. Results showed that there are more instances of faces labeled as masked than unmasked across the train-val-test distribution, reflecting the general mask behavior of the test university constituents at the time period the CCTV data was gathered. Although there is

a slight imbalance of the unmasked-masked class on our custom dataset, a small imbalance of this scale is still acceptable not to cause any possible problems in the learning process of the models [9].

After using the custom mask dataset to fine-tune the five scaled versions of YOLOv8 models (nano, small, medium, large, and xLarge), the training process produced the resulting graphs shown in Fig. 1 consisting of the training losses and performance metrics for the five models over 50 epochs. From the loss graphs (train cls_loss & val cls_loss), there exists a small spike in the loss values at around the 33rd epoch, prominently on the nano and xLarge models. This gives a hint that the models may already have achieved its best version prior to epoch 33. Nevertheless, the general downward trend of the training and validation losses indicate that the models are not over-fitting our custom dataset, which is a good indication that the models may also perform well even for the other unseen dataset. For the precision, recall, and mean average precision scores (mAP50 & mAP50-95), the graphs show a steady increase in values during the early epochs and eventually saturating at a certain point, indicating that the models may not necessarily benefit from training beyond the 50th epoch anymore. The nano version achieved its best model at the 50th epoch, the small model on the 45th epoch, medium at the 31st epoch, large at the 25th epoch, and xLarge at 28th epoch.

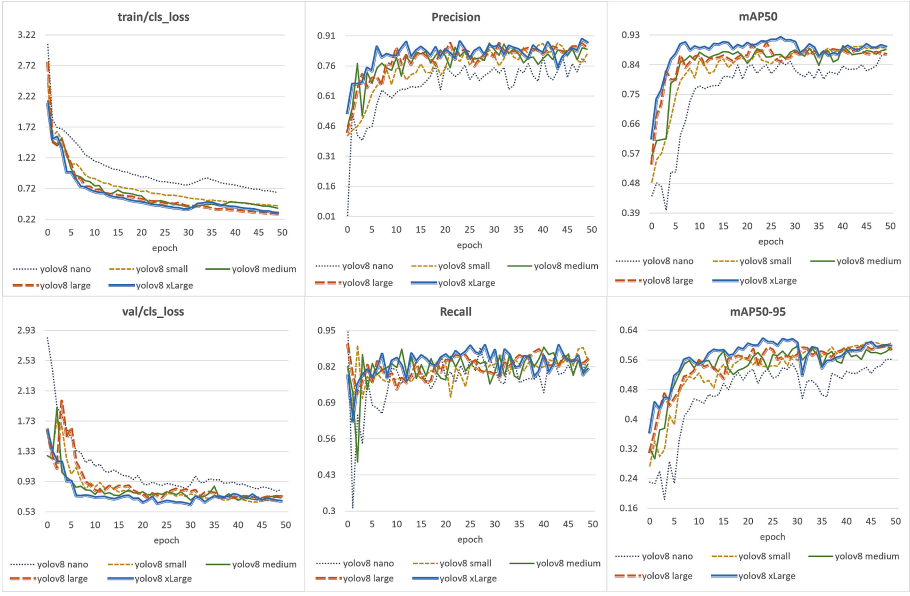


Fig. 1. Training results of the different YOLOv8 models fine-tuned on the custom mask dataset for 50 epoch.

Table 1. Summary of the developed fine-tuned YOLOv8-based mask detection models.

	Params (M)	Speed* (ms/img)	Precision	Recall	F1	mAP50	mAP50-95
nano	3.006	15.5	0.808	0.811	0.809	0.864	0.548
small	11.126	18.1	0.866	0.850	0.858	0.925	0.617
medium	25.841	19.5	0.876	0.849	0.862	0.918	0.625
large	43.608	25.3	0.878	0.808	0.842	0.898	0.609
xLarge	68.125	30.2	0.904	0.843	**0.872**	**0.940**	**0.630**

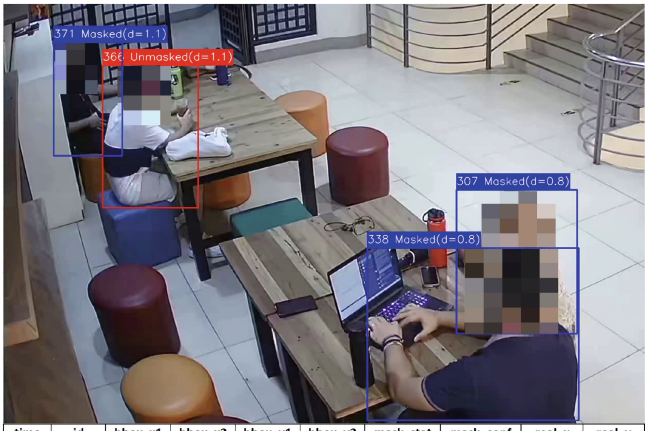
*(Speed = preprocess + inference + postprocess) using NVIDIA 4060 mobile GPU

For the detailed training (fine-tuning) results, Table 1 shows the summary properties of the five fine-tuned YOLOv8 models for mask detection. Results showed that the xLarge is the best-performing model with F1 = 0.872, mAP50 = 0.940 and mAP50-95 = 0.630. On the other hand, results showed that the nano model is the least performing model with F1 = 0.809, mAP50 = 0.864 and mAP50-95 = 0.548. However, when considering the average speed of the models, the nano variant, with an average speed of 15.5 ms/img, is the fastest among the five models. As expected the xLarge variant, with an average speed of 30.2 ms/img, performs the slowest. These results suggest that the nano model should be used if faster detection time is at most priority. If however, the accuracy rate (in terms of detection & classification) is more desired over than speed, the xLarge model is more suited for such kind of use case. The small (F1 = 0.858 & speed = 18.1), medium (F1 = 0.862 & speed = 19.5), and large (F1 = 0.842 & speed = 25.3) models are also good options when considering a balance between speed and accuracy.

After implementing the two-stage YOLOv8-based cascaded object tracking and detection pseudo-code shown in Algorithm 1 into a Python code, the study is able to process the raw CCTV footages producing the desired behavior dataset and output video files. Figure 2 shows an example snapshot of the resulting video frame generated by the system, as well as its corresponding entry in the output behavior dataset CSV file. The tracked individuals in the video frames were annotated with a bounding box (colored as red if unmasked and colored as blue if masked) enclosing the person. The unique ID, mask behavior, and the physical distance value (in meters) of the person to its nearest neighbor were also included in the annotated frame. Additionally, all faces in the resulting frames were adequately blurred-out (pixelated) enough to be unrecognizable, preserving the identity privacy of the individuals. The system was also able to extract other relevant information from the image frame and generate those data in the corresponding resulting behavior dataset.

Though the developed system is able to achieve its goal to generate the mask and distancing behavior of the individuals from the input CCTV frames, the study still comes with some limitations. Since the mask detection model was trained (fine-tuned) on a custom dataset taken from the CCTV footages of a test university, the model may only work best on the conditions akin to the

extrinsic appearance of the individuals in training dataset. For example, almost all individuals in the training dataset were only using common medical and non-medical face masks when wearing one, thus other clothing (e.g. hijab and other similar clothing) and some accessories (e.g. head cap, costumes covering the head, etc.) may affect drastically the detection performance of the models of the study. The way how the individuals are captured in the CCTV camera image frame also poses another limitation on the developed system. Some image frames are not able to capture the whole body of the individuals in the frame (see the bounding box of individuals in Fig. 2 as an example). The system can only assume the real world position of the individual base from the resulting bounding box generated by the object (person) detection model, resulting in an inaccurate 2d to 3d coordinate translation result when only a portion of the individual's body is detected in the image frame.



time	id	bbox_x1	bbox_x2	bbox_y1	bbox_y2	mask_stat	mask_conf	real_x	real_y
25:38	307	877	1114	622	907	1	0.855	2.170	5.204
25:38	338	704	1115	735	1074	1	0.807	1.529	5.641
25:38	366	191	379	378	654	0	0.847	0.817	2.285
25:38	371	93	234	334	557	1	0.836	0.653	1.201

Fig. 2. Example snapshot of the resulting video frame and its corresponding entry in the generated behavior dataset.

4 Conclusion

A detailed implementation of a two-stage YOLOv8-based cascaded approach was used in generating a movement behavior dataset with respect to two behaviors: physical distancing and mask wearing. The first stage involved tracking of individuals in the video feed to determine their physical distancing behavior based on the euclidean distance. The second stage involves determining the mask wearing behavior of the tracked individuals using the custom-trained YOLOv8 model. From the five different YOLOv8 models (nano, small, medium, large,

and xLarge), the xLarge mask detection variant has the highest performance scores while the nano variant has the lowest scores. In terms of speed, however, the nano variant is the fastest while the xLarge variant is the slowest among the five models. The medium variant presents a good balance between accuracy and speed. Finally, a blurring-out method was employed to make the faces in the resulting video frames unrecognizable, thus preserving the identity privacy of the individuals. The source code used in the study and the fine-tuned models are available at https://github.com/rpabao/dataset-generator-YOLOv8.

Future works of the study include validating the actual accuracy of the fine-tuned mask detection and distancing models on real world data (and other external dataset), to establish the soundness of the behavior dataset generator developed in the study. Another possible route would be to use other deep learning algorithms (e.g. R-CNNs, SSD, and other versions of YOLO) for object detection and compare its performance to the fine-tuned YOLOv8-based models in this study, to get the best performing model out of all the object detection algorithms available.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. An, L., et al.: Challenges, tasks, and opportunities in modeling agent-based complex systems. Ecol. Model. **457**, 109685 (2021)
3. Jocher, G., Chaurasia, A., Qiu, J.: Yolo by ultralytics (2023). https://github.com/ultralytics/ultralytics. Accessed 26 Mar 2023
4. Kasai, T.: Adapting to life with covid-19 and staying safe (2021). https://www.who.int/westernpacific/news-room/commentaries/detail-hq/adapting-to-life-with-covid-19-and-staying-safe. Accessed May 2023
5. Nazir, A., Wani, M.A.: You only look once-object detection models: a review. In: 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1088–1095. IEEE (2023)
6. OpenCV: Perspective-n-point (pnp) pose computation (nd). https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html. Accessed 26 Mar 2023
7. Ravaioli, G., Domingos, T., Teixeira, R.F.M.: A framework for data-driven agent-based modelling of agricultural land use. Land **12**(4) (2023). https://doi.org/10.3390/land12040756
8. Terven, J., Cordova-Esparza, D.: A comprehensive review of yolo: from yolov1 to yolov8 and beyond. arXiv preprint arXiv:2304.00501 (2023)
9. Weiss, G.M.: Foundations of imbalanced learning. In: Imbalanced Learning: Foundations, Algorithms, and Applications, pp. 13–41 (2013)
10. World Health Organization: Statement on the fifteenth meeting of the ihr (2005) emergency committee on the covid-19 pandemic (2023). https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-coronavirus-disease-(covid-19)-pandemic. Accessed 10 May 2023
11. Xiao, Y., et al.: A review of object detection based on deep learning. Multimedia Tools Appl. **79**, 23729–23791 (2020). https://doi.org/10.1007/s11042-020-08976-6