Real-time Detection of Face Mask Wearing Based on YOLOv5

Zecheng Wang
Xiamen University

Lei Peng
Xiamen University

Wensheng Pan
Xiamen University

Kaifan Wang
Xiamen University

Houjin Chen
Xiamen University

Abstract

After the outbreak of COVID-19, the public health protection needs higher requirements. At present, people must wear masks when entering or leaving public places and taking public transports. The wearing inspection of masks has become an essential operation for epidemic prevention and control. It is of great significance to use machines to identify whether people wear masks.

In order to solve the problem existing in manual inspection of personnel's mask wearing, this paper proposes a method of mask wearing detection using an improved YOLOV5 model. This method mainly adjusts the input size and initial candidate box of the original YOLOV5 model, and improves the convolution layer and loss function, so that YOLOV5 model is more suitable for the detection and recognition of mask wearing. It has the function of real-time detection of pictures, videos and cameras, and can detect the wearing of face masks in real time. The detection speed is expected to reach 100FPS under the GPU environment, and the average accuracy is expected to reach more than 80% to meet the requirements of real-time detection. Our team will first study the principle of YOLOv5 algorithm, and then improve YOLOv5 to output a high detection probability and meet the requirements of realtime detection of face masks. Then, the mask training data sets collected on the network will be trained, and some parameters will be fine tuned to make the model achieve the best effect.

Finally, the improved YOLOV5 model is tested by using test data sets. The test results will show that the model has high detection speed and recognition rate. However, its detection and recognition effect will be limited by such factors as training data sets, light, camera, face occlusion and distance.

1. Introduction

In recent years, image recognition has become a hot topic in the field of image research. In medicine, images of various parts of a patient's body can help doctors determine the patient's condition. In life, license plate detection in parking lots, security check systems in high-speed railway stations, face recognition payment, etc. Images have become indispensable in our daily life. With the development of deep learning and the great research results and technical breakthroughs of convolutional neural networks in face detection, the society has an increasing demand for the degree of accuracy of face detection, especially when the epidemic has not yet dissipated, it is important to use machines to identify whether people are wearing masks or not.

This project will use the YOLOv5 algorithm [3] for face mask detection, which was created in 2015 and has been updated for five versions since then. In the training phase of the model, the input side of the Mosaic data enhancement, adaptive anchor frame calculation, and adaptive image scaling improvements are added, and its benchmark network incorporates the Focus structure and CSP structure, and the object detection network tends to insert some layers between the BackBone and the final Head output layer, while the FPNPAN structure is added in Yolov5. YOLOv5 improves the loss function GIOU_Loss [10] for training, and DIOU_nms for prediction frame filtering. YOLOv5 uses the Pytorch framework, which is very userfriendly and can easily train its own dataset, and the training speed is very fast, and it can directly process individual images, or batch images, so using the YOLOv5 algorithm to achieve the detection of The use of the YOLOv5 algorithm for face detection with or without a mask is fully feasible and will yield good experimental results.

The difficulty of this project is that there are few publicly available datasets on the Internet, and we need to collect and produce some of the data by ourselves, so it is a problem to make the model achieve good results with a small training set. At the same time, this project not only detects whether the mask is worn or not, but also determines whether the mask is worn correctly, which requires detecting the nose, mouth and chin, i.e. this is a triple classification task, so this requires our model accuracy to be further improved.

In this project, we improved the algorithm of YOLOV5

to be able to detect whether a mask is worn and whether it is worn correctly, and implemented a graphical interface to be able to detect masks for people in pictures, videos, and cameras.

The rest of the paper is organized as follows, in Section 2 we briefly review the work related to this project, in Section 3 we give our plan for implementing the mask detection task, in Section 4 we present some of the methods used in the experiments, in Section 5 we give the experimental results, and finally in Section 6 we conclude the paper.

2. Related work

At present, mainstream object detection algorithms are mainly based on deep learning model, which can be roughly divided into two categories: (1) One-Stage object detection algorithm. Typical algorithms include YOLO, SSD [5] and CornerNet [4]. (2) Two-stage object detection algorithm. Fast R-CNN [2], Faster R-CNN [9], etc.

One-Stage object detection algorithm can directly generate the category probability and position coordinate value of objects at a stage. Compared with the Two-Stage object detection algorithm, the Region Proposal stage is not required, and the overall process is relatively simple. In Testing, input images will generate output through CNN network, decode (post-processing) and generate corresponding detection box; During Training, it is necessary to encode the Ground Truth into the corresponding format of CNN output in order to calculate the corresponding loss.

The two-stage object detection algorithm is regarded as Two One-Stage detection. The first Stage preliminarily detects the location of the object, and the second Stage further refines the results of the first Stage to carry out One-Stage detection for each candidate region. Overall process as shown in the figure below, at the time of Testing input images after convolution neural network to produce the first phase of the output, the output decoding process to generate candidate area, and then obtain corresponding to the feature of candidate said (ROIs), and then produces the output of the second phase to ROIs further elaboration, decoding (post-process) to generate the final result, Decode and generate corresponding detection box; During Training, it is necessary to encode the Ground Truth into the corresponding format of CNN output in order to calculate the corresponding loss.

Seeking continuous improvements in speed and enhancing the models' computation complexity in CNNs, the You Only Look Once (YOLO) algorithm was first released in 2015 as YOLOv1 [6]. YOLO is the best object detection algorithm because it solves the trade off between speed and accuracy. It is significantly faster as it processes 45 frames per second which enables the detection of real-time objects with high accuracy. Also, YOLO can learn the general representation of objects and apply them in detection. The

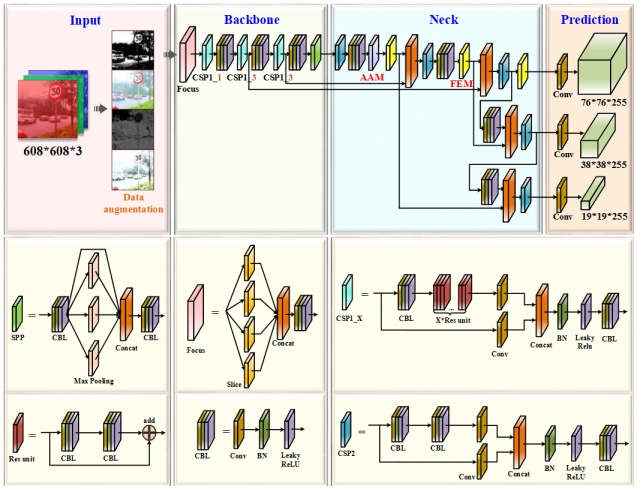

Figure 1. Model structure of YOLOv5

updated version YOLOv2 [7] appeared in 2016 with better speed, accuracy and localization. In 2018, after the release of YOLOv3 [8], YOLO was first used in mask detection because it achieved state-of-the art performance exceeding the recurrent CNN (R-CNN) family. In 2020, YOLOv4 [1] was released twice faster than YOLOv3 and used in real-time mask detection application and device installed in Politeknik Negeri Batam [11] with no recorded accuracy. Finally, YOLOv5 was released in [3] introducing the concept of adaptive anchor boxes and mosaic data augmentation for better results. Although, YOLOv5 acceleration and smaller models were claimed to outperform YOLOV4, the model accuracy was only 81%.

In order to implement a more efficient and high accuracy method to detect mask Based on Yolov5 this paper proposes a method of mask wearing detection using an improved YOLOV5 model. This method mainly adjusts the input size and initial candidate box of the original YOLOV5 model, and improves the convolution layer and loss function, so that YOLOV5 model is more suitable for the detection and recognition of mask wearing. It has the function of real-time detection of pictures, videos and cameras, and can detect the wearing of face masks in real time. The detection speed is expected to reach 100FPS under the GPU environment, and the average accuracy is expected to reach more than 80% to meet the requirements of real-time detection.

3. Method

3.1. YOLOV5

YOLOv5 was introduced in 2020 one month later to the release of YOLOv4. YOLOV5 is a novel repository written completely in Python programming language instead of C as in previous versions, which gives this version the edge of being easier to develop and more flexible to install and in-

tegrate on IoT devices given its weight and speed. Also, YOLOv5 successfully integrated the latest improvements in YOLOv4 and even enhanced them. YOLOv5 architecture, similar to YOLOv4, is divided as: input, backbone for feature extraction, neck for feature aggregation to mix and combine features, and finally the head for making detection including localization and classification.The model structure of yolov5 is shown in Fig. 1.Note that Fig. 1 is quoted from the Improved YOLOv5 network for real-time multi-scale traffic sign detection [12].The main blocks of YOLOv5 are summarized as follows:

Input: First, the input of the network is usually an image with a size of 608 × 608 pixels. Yolov5 adopts the same Mosaic data enhancement method as Yolov4. The method of random scaling, random clipping and random layout is very effective for small object detection. In the Yolo algorithm, for different data sets, there will be anchor frames with the initial set length and width. In the network training, the network outputs the prediction box on the basis of the initial anchor box, and then compares it with the real box groundtruth to calculate the gap, and then reversely updates and iterates the network parameters. In Yolov3 and Yolov4, when training different data sets, the calculation of the initial anchor box value is run through a separate program. However, Yolov5 embeds this function in the code. Each training, the optimal anchor box value in different training sets is adaptively calculated.

Backbone: Focus Structure and Cross Stage Partial Network (CSP) which divide the input feature maps to a portion that is forwarded through the deep layers of the network and another portion that is sent directly to next layer without processing. This helps fine grained features to be propagated efficiently throughout the network.

Neck: For feature extraction, YOLOv5 used both Spatial Pyramid Pooling block (SPP) and Path Aggregation Network (PANet).

Head: One stage detector – as the case in YOLOv3 – using Generalized IoU (GIoU) loss is used. Intersection over Union (IoU) is a concept used in object detection. It is the overlap rate of the generated candidate bound and the original ground truth bound, that is, the ratio of their intersection and union. The best case is full overlap, that is, a ratio of 1. GIoU is an enhancement of IoU. GIOU introduces the minimum external frame based on IOU characteristics to solve the problem that loss equals zero when the detection frame and the real frame do not overlap.It measures the intersection between the grounding truth and predicted bounding boxes and also measure how much the predicted is getting closer or nearer the truth ones even if their intersection is still zero.

YOLOv5 has successfully innovated and implemented the adaptive anchor boxes techniques. For all previous versions, the 5 best fit anchor boxes were picked from the COCO dataset and used them as default starting point. In contrast, in YOLOv5 there is no default anchor boxes. The network automatically learns the best fit anchor boxes depending on the used dataset.

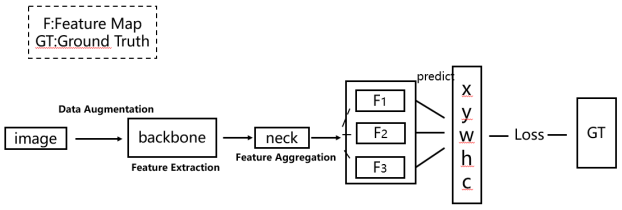

Figure 2. The proposed framework

3.2. Improvement

3.2.1 Triple Classification Task

The real demand is that we not only need to detect whether people wear masks, but also need to detect whether people wear masks correctly. Because the mask is not properly worn, it will play a very limited role. Only when the mask can cover the mouth, nose and lower part of the face can we think that the mask is correctly worn.Therefore, we divide all the detected objects in the input images into three categories: with_mask,without_mask and mask_weared_incorrect.

In this project, we improved the algorithm of YOLOV5 to be able to detect whether a mask is worn and whether it is worn correctly, and implemented a graphical interface to be able to detect masks for people in pictures, videos, and cameras.

3.2.2 Visual Interface

Qt is a set of framework for graphical interface programming encapsulated in C++ language. Qt focuses on but is not limited to the development of graphical interface, but also supports system call, network programming, database programming, 2D/3D graphics processing, audio and video processing, etc. In order to facilitate the detection, we use Python for Qt to achieve a simple interface. It includes two functions: picture detection and video detection. In the image detection interface, we need to upload local images for detection. In the video detection interface, we can upload local video detection or open the camera for real-time detection.

4. A Faster Face Mask Detector Framework

In this paper, we present a framework to detect whether or not a face mask is worn. The motivation of the proposed framework is to be accurate and fast. This contrasts with existing works that are either accurate but not real-time. or

did not provide high accuracy for the sake of speed. The proposed framework is shown in Fig. 2.

4.1. Train

The format of our training and testing data set is in accordance with VOC data set. During the training, stochastic gradient descent method is adopted to train the network model. The initial learning rate is set as 0.01 and a linear learning rate decliner is set. weight_dacay is 0.0005, we trained our model in an RTX 3060 GPU with 16 GB memory. We used an image with a size of 640 × 640 pixels, a batch size of 64, and trained the model with 300 epochs.The dataset consists of 853 images. Of these, 697 images were used for training and 156 were used for validating. The training set and the val set are independent of each other.There are two strategies we use for training

A.Data Augmentation

In order to increase the robustness of the network structure and reduce the possibility of overfitting, some data enhancement methods such as flipping, clipping and translation are adopted in this project. Flip: Flip the picture horizontally and vertically (180 degree rotation followed by horizontal flip). Crop: Randomly sample a section from the original image, then we resize this section to the original image size. Panning: Shifting only involves moving the image along the X or Y direction (or both). Moving the image is very effective, it can make the target appear anywhere in the image to break the model's memory of the position. And other image processing operations

B.Label Smoothing

Overfitting and probabilistic calibration are two problems in deep learning model training. There are many regularization techniques in deep learning that can solve the overfitting problem. Weight attenuation, early stop mechanisms, and dropout are all the most common methods. Platt scaling and sequentially preserving regression can be used for model calibration. Label smoothing is a regularization technique that perturbs target variables and reduces the certainty of the model's prediction. It is considered a regularization technique because it limits the maximum probability of the softmax function so that the maximum probability is not much greater than other tags (overconfidence). We have a multi-class classification problem. In such problems, the target variable is usually a one-hot vector, where the position of the correct category is 1 and the other positions are 0. This is a different task than binary classification because in binary classification there are only two possible classes, but in multi-label classification there can be more than one correct class in a single data point. Therefore, the multi-label classification problem needs to detect every object existing in the image. Label smoothing changes the target vector by a small amount of epsilon. Thus, instead of asking our model to predict 1 for the correct category, we

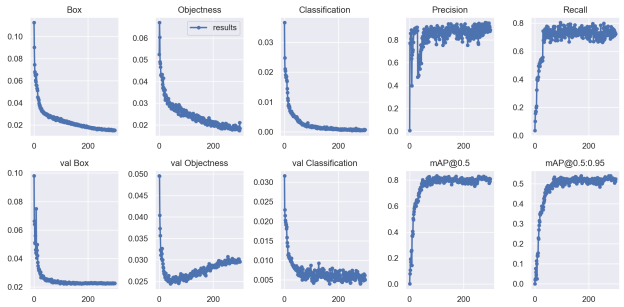

Figure 3. Verification results of each Epoch of YOLOv5 in training on Val set.

ask it to predict 1-ε for the correct category, and to predict all other categories to be ε. The cross entropy loss(eq.1) function with label smoothing is translated into the following formula(eq.2).

$$Loss = -\Sigma_{i=1}^{k} p_i \log q_i \tag{1}$$

$$Loss_i = \begin{cases} (1-\varepsilon) * Loss, if(i = y) \\ \varepsilon * Loss, if(i \neq y) \end{cases} \tag{2}$$

4.2. Validation

The test was carried out on sheet 3060Ti. The data format of the test set referred to that of VOC, and the test set data and training set were independent from each other. The val set consists of 156 images.For verification, set the iou threshold to 0.6,confidence score threshold to 0.001, input image size to 640x640, batch_size to 16 and turn off data enhancement and label smoothing.The test results are shown in Fig. 3 and Fig. 4.In general, the average accuracy of the test and the actual test results are relatively good. After training with 300 epoch, its PR-curve is shown in Fig. 5. The average accuracy of the model reaches 0.809 in the verification set, and the accuracy of the model is very high after the introduction of non-standard mask wearing without affecting the reasoning speed. The reason why the accuracy rate of non-standard mask wearing is low is that compared with the two kinds of data of wearing and not wearing masks, the amount of non-standard mask wearing data is small and can be considered as the tail-class, whose loss contribution is suppressed by the head-class. The data distribution of examples is shown in Fig. 6.

4.3. Detection

We built a real-time monitoring platform based on this model, which can receive pictures, videos and even cameras for real-time detection of mask standard wearing. In the detection, the subjects were tested in three states: wearing masks, non-standard wearing masks and not wearing masks. The experimental results are shown in Fig. 7.

Figure 4. The val results on a batch

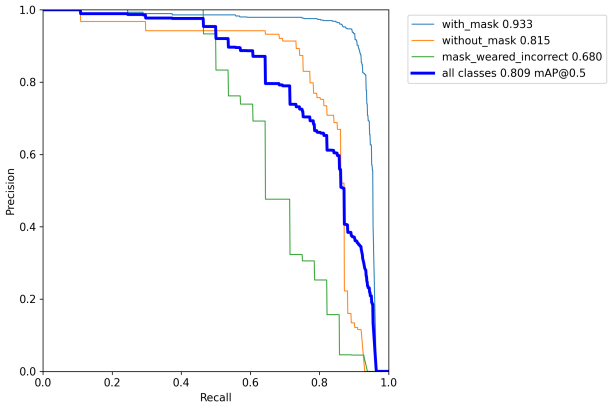

Figure 5. PR-Curve after training 300 epochs

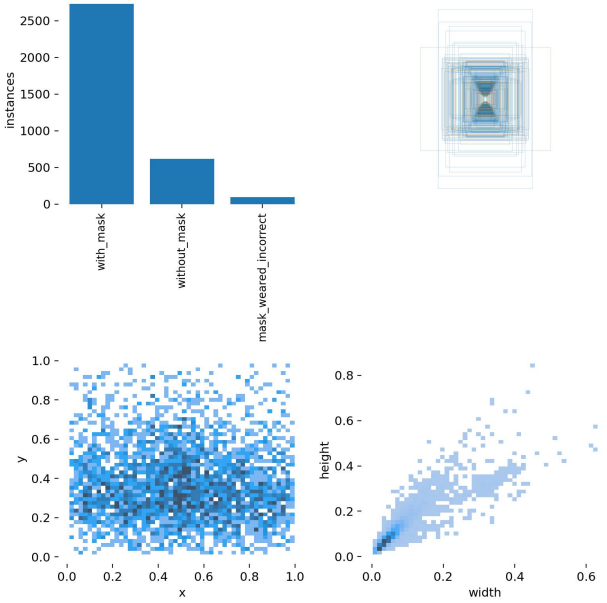

Figure 6. The number of instances of each class

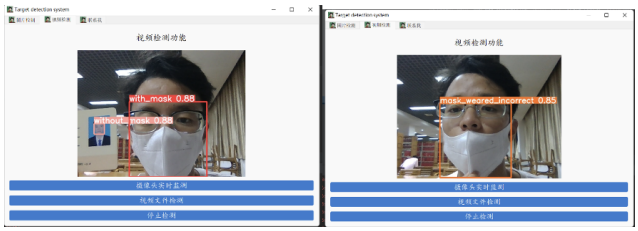

Figure 7. The detection example with QT user-interface

5. Experimental Results and Discussion

We used the original YOLOv5 for the experiment.Then we evaluated the performance of the entire framework with the improved YOLOv5. All code is written in Python. The implementation and experiments were based on the Pytorch deep learning framework.The network was trained and tested on our local machine with an RTX 3060 GPU with 16 GB memory. We used an image with a size of 640 × 640 pixels, a batch size of 64, and trained the model with 300 epochs.

5.1. Performance Metrics

The model performance is evaluated by the widely used metrics: the mean Average Precision (mAP), precision, and recall. The mAP metric is used for comparing how close the bounding box to the detected box and returns a score. Higher scores mean more accurate models.The mAP is computed depends on the precision and recall. The precision measures how accurate is the model in detecting objects. The recall is mainly concerned by number of missed objects by the algorithm.The precision and recall are defined as

$$Percision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \quad (3)$$

$$Recall = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FN)} \quad (4)$$

where TP is the number of objects detected correctly, TN is the number of correctly not detected samples. FP is the number of the objects the model failed to detect and FN is the number of the wrongly not detected samples. We then define the average precision (AP) to illustrate the relationship between precision and recall in one value representing the average of all precisions. After getting AP for each class, we can then get the mAP which is the mean APs for all classes. All mAP results are performed at IoU greater than 0.5.

5.2. Results and Discussion

In this paper, we propose a real-time framework and a qt interface for accurate mask detection. We also have an

external camera that can take photos in real time and detect whether masks are worn or whether masks are worn correctly. We can also add albums for real time detection through the qt interface. The framework first adds noise to the images and then applies a series of image enhancement techniques to increase the size of the data set and prevent overfitting. Finally, negative samples were added to the data set to enhance the training process and reduce false positives. Our framework uses YOLOv5 as the detection core. With only 45 minutes of training time and nearly 10 milliseconds of image inference using FMD, the accuracy achieved was 0.933 and the average accuracy was 0.809. Our future work will include balancing data sets by over-sampling techniques, further modifying YOLOv5 by adding momentum, using random weighted averages, applying dropout and implementing cutmix enhancement techniques instead of mixing.

6. Conclusion

In this work, we propose an improved mask wearing detection method based on the YOLOV5 model. First, we adjust the input size and initial candidate frame of the YOLOV5 model according to the task of detecting and identifying mask wear, and optimize the loss function of the convolutional layer to improve the detection speed and accuracy of the model for the mask detection task. At the same time, the model can distinguish not only whether people are wearing masks or not, but also whether they are wearing masks correctly. In addition, we implement a graphical interface in which we provide three types of detection methods: picture, video and live camera, making it more convenient for users to use.

We train the improved model with the mask training dataset collected on the web and make adjustments to the parameters based on the results to finally achieve the best performance of the model. Finally, we test the improved model using a test dataset, and the model finally achieves 0.933 accuracy. The training results and analysis show that the improved YOLOV5 algorithm is stronger than the original YOLOV5 algorithm for the mask detection task. However, there are still some shortcomings in our work, as both our test and training datasets are relatively small, the accuracy of the model can still have some deviations. In future work, we will continue to collect production images to expand our dataset and continue to adjust the parameters of the model based on the training and testing results.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[3] Glenn Jocher. YOLOv5 by Ultralytics, 5 2020.

[4] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[7] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[10] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[11] Susanto Susanto, Febri Alwan Putra, Riska Analia, and Ika Karlina Laila Nur Suciningtyas. The face mask detection for preventing the spread of covid-19 at politeknik negeri batam. In *2020 3rd International Conference on Applied Engineering (ICAE)*, pages 1–5. IEEE, 2020.

[12] Junfan Wang, Yi Chen, Zhekang Dong, and Mingyu Gao. Improved yolov5 network for real-time multi-scale traffic sign detection. *Neural Computing and Applications*, pages 1–13, 2022.