DOI: https://doi.org/10.15276/aait.06.2023.15
UDC 004.93

A survey on deep learning based face detection

Tran The Vinh Tran[1)]
ORCID: https://orcid.org/0000-0002-4241-1065; vinhtt@ut.edu.vn. Scopus ID: 288641
Tien Thi Khanh Nguyen[1)]
ORCID: https://orcid.org/0000-0001-5379-7226; tienntk@ut.edu.vn
Tran Kim Thanh[1)]
ORCID: https://orcid.org/0000-0002-4241-1065; tkthanh2011@gmail.com
[1)] Ho Chi Minh City University of Transport, Ho Chi Minh City. Vietnam

ABSTRACT

The article has focused on surveying face detection models based on deep learning, specifically examining different one-stage models in order to determine how to choose the appropriate face detection model as well as propose a direction to enhance our face detection model to match the actual requirements of computer vision application systems related to the face. The face detection models that were conducted survey include single shot detector, multi-task cascaded convolution neural networks, RetinaNet, YuNet on the Wider Face dataset. Tasks during the survey are structural investigation of chosen models, conducting experimental surveys to evaluate the accuracy and performance of these models. To evaluate and provide criteria for choosing face detection suitable for the requirements, two indicators are used, average precision to evaluate accuracy and frames-per-second to evaluate performance. Experiential results were analyzed and used for making conclusions and suggestions for future work. For our real-time applications on face-related camera systems, such as driver monitoring system, supermarket security system (shoplifting warning, disorderly warning), attendance system, often require fast processing, but still ensures accuracy. The models currently applied in our system such as Yolos, Single Shot Detector, MobileNetv1 guarantee real-time processing, but most of these models have difficulty in detecting small faces in the frame and cases containing contexts, which are easily mistaken for a human face. Meanwhile, the RetinaNet_ResNet50 model brings the highest accuracy, especially to ensure the detection of small faces in the frame, but the processing time is larger. Therefore, through this survey, we propose an enhancement direction of the face detection model based on the RetinaNet structure with the goal of ensuring accuracy and reducing processing time.

Keywords: Face detection; one-stage detector; two-stage detector; deep learning; single shot detector; multi-task cascaded convolutional neural networks; RetinaNet, YuNet

***For citation*: Tran The Vinh, Nguyen Thi Khanh Tien, Tran Kim Thanh "A survey on deep learning based face detection". *Applied Aspects of Information Technology*. 2023; Vol. 6 No. 2: 201–217. DOI: https://doi.org/10.15276/aait.06.2023.15**

INTRODUCTION

When building computer vision applications related to faces [1, 2], [3, 4], [5, 6], [7] (such as face recognition, gender classification, face landmark detection, etc.), face detection is one of the first and most important tasks. Face detection (FD) [8, 9] makes it possible for the system to detect the presence and position of a face in an image or in a video stream. The input of almost any face detection algorithm is an image. The output is an image area containing a rectangular face that can be represented by 4 points (or 2 points and length and width) along with the probability that the face is in that image area.

The accuracy and processing speed of face detection is very important, directly affecting the functionality of the entire application system today [10] in areas including: Security, Marketing, Healthcare, Entertainment, Law Enforcement, Surveillance, Photography, Games, and Video Conferencing, etc. Optimizing AI models in the face detection problem is always an urgent task and should be prioritized when there is a huge requirement for robustness and real time through the camera system. The face detection problem often encounters challenges that reduce the performance of the Face detection system, which currently does not have an AI model that can completely solve, such as occlusion, light, skin color, facial poses, facial expressions, accessories (glasses, masks...), face ratio [8, 9]. The FD system can detect partially obscured faces, but it is difficult for the system to confirm whether the face is completely or partially obscured in the frame. The light, skin color is a detrimental factor for the system especially under constantly changing lighting conditions, so to enhance the detection ability when conducting training FD models often add manipulation augmentation during data generation. The performance of the FD model is also often reduced when the face is tilted, turned to the side,

wearing sunglasses, and wearing a mask. In the case of multi-detection cases, faces with too small proportions in photos and videos are often easily missed when detected.

LITERATURE REVIEW

Classical face detection algorithms [11, 12], [13] such as Haar Cascades (2001) or DLib-HOG (2005) may not be able to detect faces in some frames, this can lead to the application not working as intended or cause complexity in the system. In addition, FD processing time is large, which increases the time and memory of the entire application system, especially in the real-time applications. Therefore, it is necessary to switch to face detection deep learning tools that provide high accuracy (so that no face is undetected) at very high speed and can also be used in low-power microprocessors such as CNN-based methods [11], [14, 15], [16]. CNN-based methods have significantly increased FD performance compared with traditional methods, and are being divided into two main approaches: one-stage detector [17], two-stage detector [18], as shown in Fig. 1 and Fig. 2.

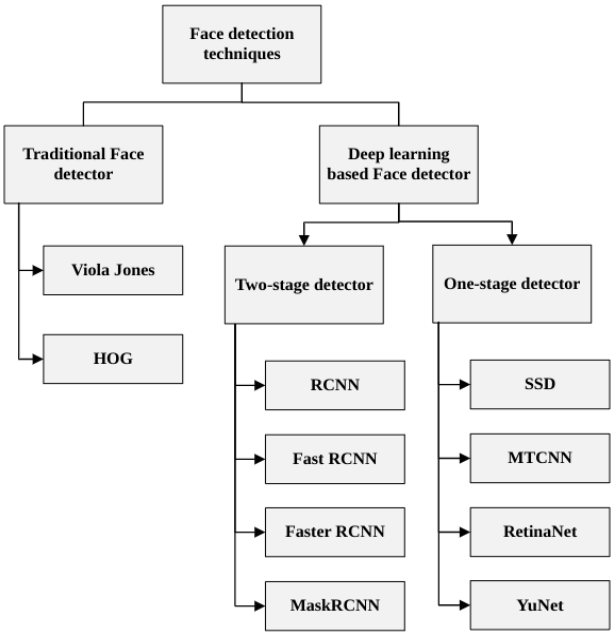

Fig. 1. **Face detection techniques**
Source: **compiled by the authors**

One-stage detector with some typical models such as: SSD [19], Yolo [20], RetinaNet [21, 22], [23]. Called one-stage because in the design of the model, there is absolutely no extraction of feature regions (areas that can contain objects) like RPN [24] or Faster-RCNN [25]. Single-stage detector models treat object detection as a regression problem (with four offset coordinates e.g., x, y, w, h) and also rely on predefined boxes called anchors to do so.

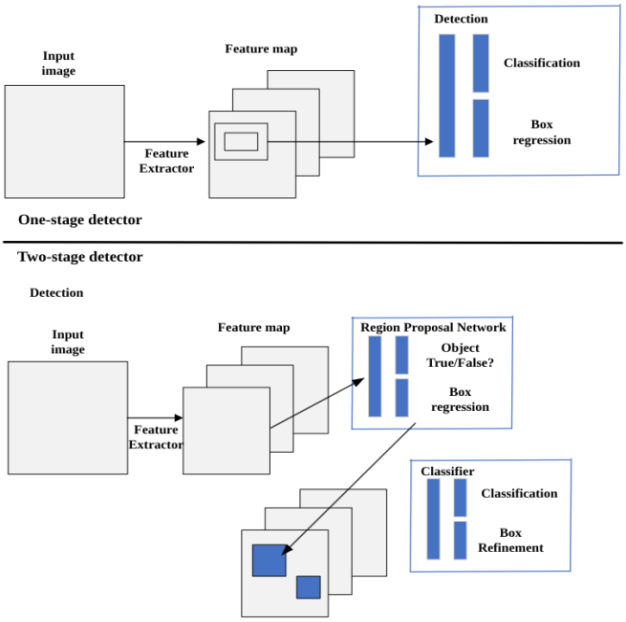

Fig. 2. **Overall schema of one-stage detector and two-stage detector**
Source: **compiled by the authors**

Models of this type often have a faster prediction speed. However, the "accuracy" of the model is often lower than that of two-stage object detection. Of course, some one-stage models still prove to be a bit superior to two-stage models such as RetinaNet with the network design according to Feature Pyramid Network (FPN) [26] and Focal Loss [21].

Some face recognition model architectures designed with one-stage detectors have achieved very remarkable results (remarkable results on common benchmarks for face detection). For example, SFD [27] adopts VGG-16 but is built more optimally (tiling and assigning the anchors more tightly) for small face detection. SSH network [28] has been proposed that removes fully connected layers from the base network VGG-16, and provides context information to the detection module by merging several convolution layers. For the masked face situation, a FAN model [29] has been proposed using a feature pyramid integrated with the attention mechanism to improve accuracy.

The reason for calling two-stage is because of the way the model handles to extract possible object areas from the image. Two-stage detectors divide the detection task into two stages: extract Region of Interest (RoIs), then classify and regress the RoIs. Some typical two-stage detector model architectures [30] are R-CNN, Fast-RCNN, Faster-RCNN, Mask-RCNN and others. With Faster-RCNN, in stage-1, the image will be given a sub-network called RPN (Region Proposal Network) with the task of extracting regions on the image that are likely to

contain objects based on anchors. After obtaining the feature regions from the RPN, the Faster-RCNN model will continue to classify objects and determine the location by dividing into 2 branches at the end of the model (Object classification & Bounding box regression). Two-stage detector achieves the better performance but has low time efficiency, for example, SSFD+ [31] focus on achieving comparable performance and simplifying the network architecture for detecting multiscale faces while it spends 582 ms on detecting a picture.

PURPOSE AND TASKS OF WORK

The purpose of the article is to determine the appropriate face dectection model selection, as well as determine the direction of face detection model enhancement aimed at increasing efficiency for building automatic face-related application systems.

With the purpose set out above, the following tasks were raised and implemented:

– structural investigation of different Deep FD models currently being applied in many face-related systems;

– conduct experimental survey to evaluate the accuracy and performance of different FD models;

– make conclusions and suggestions on the selection of suitable models; suggest directions for face detection model enhancement.

OVERVIEW DIFFERENT DEEP FACE DETECTION MODELS

Some of the face detection models currently being applied to automated systems are listed as:

– Single Shot Detector;

– Multi-Task Cascaded Convolutional Neural Networks (MTCNN);

– RetinaNet;

– YuNet.

Single Shot Detector. Single Shot Detector [19] is a single-stage object detection algorithm. Unlike two-stage models, Single Shot Detectors do not need an initial object proposal generation step. This often makes it faster and more efficient than two-stage approaches such as Faster R-CNN [25]. The Single Shot Detector has two components: the backbone model and the Single Shot Detector's head. The backbone model is usually a pre-trained image classifier network as a feature extractor. Pre-trained Resnet on ImageNet is commonly used, but leaves out the last fully connected classifier layer. For Resnet34, the backbone leads to 256 feature maps (7x7) for the input image. The Single Shot Detector's head is just one or more convolutional layers added to this backbone, and the model outputs are bounding boxes and classes of objects in the spatial location of the final layers' activations.

In Fig. 3, the Single Shot Detector model is built with VGG on a pre-trained Resnet model that is converted to a fully convolutional neural network. Then some extra convolutional layers are attached to help with handling larger objects. So, in principle Single Shot Detector's Architecture can be used with any deep network base model. These convolutional layers are added to generate feature maps of sizes 19x19, 10x10, 5x5, 3x3, 1x1, along with the feature map (38x38) generated by VGG's (conv4_3) will be the feature maps to use for predicting bounding boxes. This algorithm accepts a significant reduction in detection performance of small objects to achieve processing speed. In the Single Shot Detector's architecture, single-layer object detection models will need fewer features that are easy to learn. Smaller networks with fewer parameters reduce the number of computations, thus reducing processing time significantly.

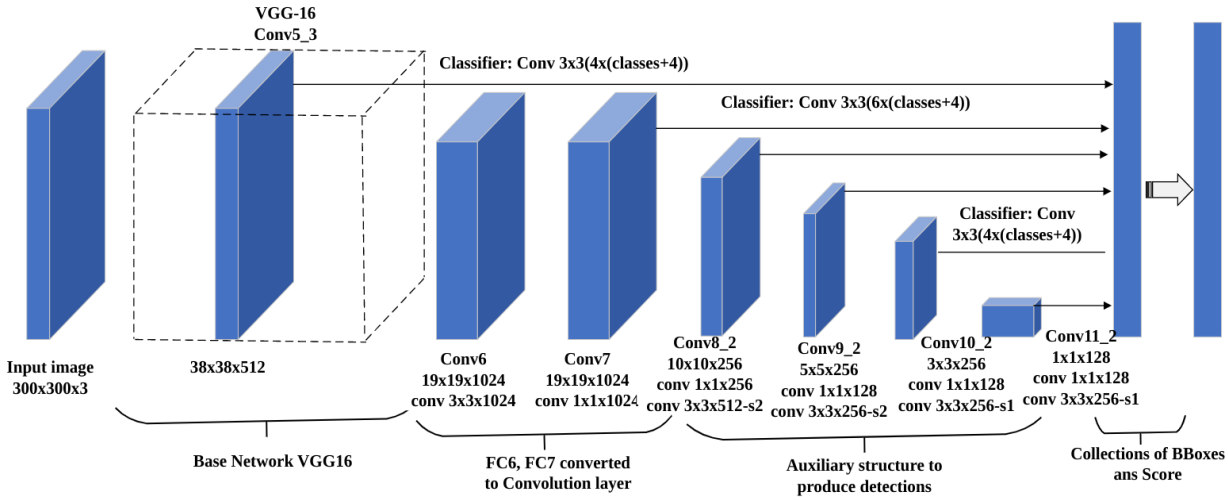

Fig. 3. **Overall architecture of Single Shot Detector**

Source: **compiled by the authors**

Instead of using a sliding window, the Single Shot Detector divides the image by a mesh for the purpose of detecting objects in that region of the image. Detected objects are just a prediction of the class and position of an object in that area. If there are no objects, the area is the background layer and the position is ignored. Each grid cell can export the position and shape of the object it contains. Each grid cell in Single Shot Detector can be assigned with anchor boxes/prior boxes. These anchor boxes are predefined and contain information about the size and shape in a grid cell. The Single Shot Detector uses a matching phase during training, to match the appropriate anchor box to the bounding boxes of each underlying truth object in an image. The anchor box with the highest overlap with an object will predict the class of that object and its position. This property is also used for predicting objects after the network has been trained.

In practice, each anchor box is specified by its aspect ratio and zoom level.

For each default box on each cell the network output the following:

– a probability vector of length *c*, where c is the number of classes plus the background class that indicates no object.

– a vector with 4 elements (*x*, *y*, *width*, *height*) representing the offset required moves the default box position to the real object.

The process of training a Single Shot Detector minimizes classification and regression losses through multi-box loss function, which is calculated using the following formula (1):

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)), \quad (1)$$

where:

$L_{conf}(x,c)$ – class score;

$L_{loc}(x,l,g)$ – bounding box offset;

x – *default* box;

c – is class;

l – bbox coordinate;

g – GT coordinate;

N – number of bounding boxes, that have IoU(l,g)>= 0,5

The single shot detector's loss balances the classification objective and the localization objective.

Multi-task cascaded convolution neural networks (MTCNN). MTCNN was published in 2016 by Zhang et al. [32], is one of the most commonly applied facial recognition tools today. MTCNN is a neural network that detects faces and facial landmarks on images using a cascading structure with three stages P-Net, R-Net and O-Net.

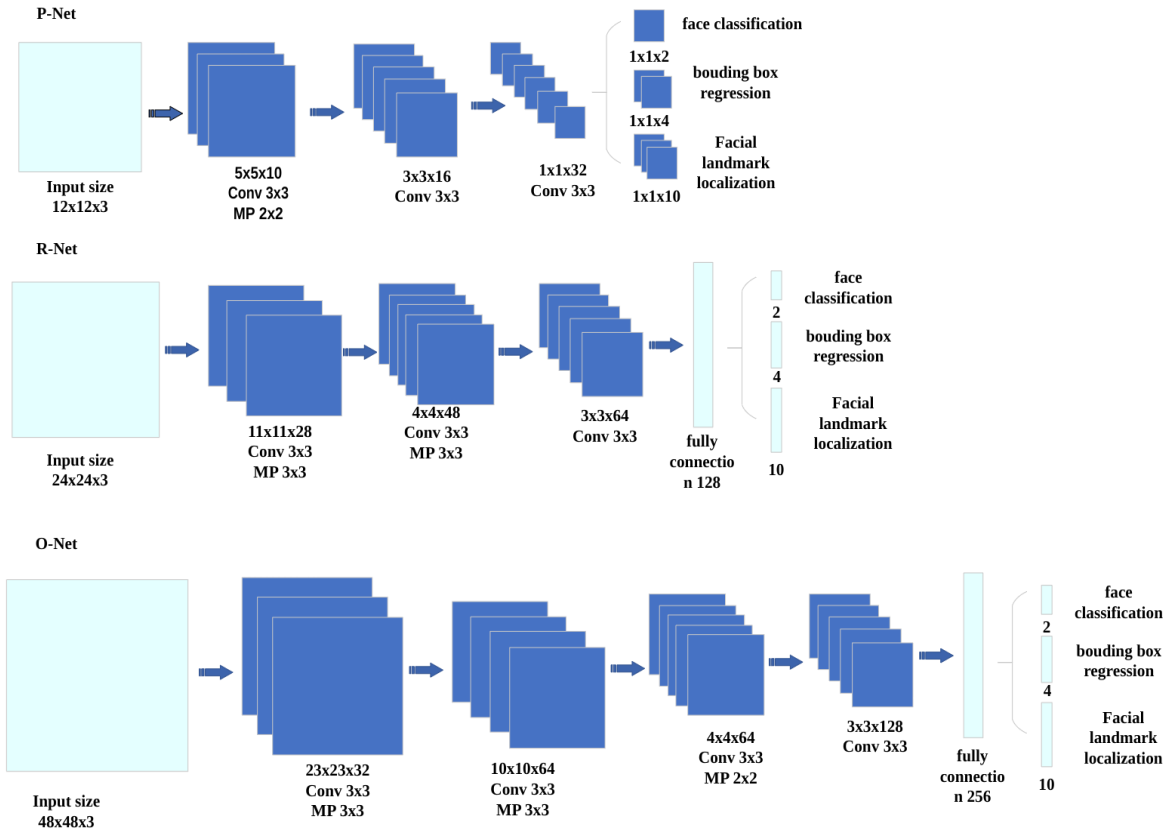

Fig. 4. **Multi-task cascaded convolution neural networks architecture: P-Net, R-Net, O-Net**

Source: **compiled by the authors**

Model structure, as shown in Fig. 4, is mainly based on 3 separate CNN models (P-Net, R-Net and O-Net). P-Net (Proposal network) searches for faces in frames of size 12 × 12. The task of this network is quickly generating candidate windows. R-Net (Refined network) has a deeper structure than P-Net. All candidates from the previous P-Net network are included in the R-Net, for the purpose of filtering and selecting high-precision candidates.

Finally, O-Net (Output network) returns the bounding box (face area) and key point landmarks on the face. In the MTCNN model structure, in addition to using convolution networks to solve image problems; the model also uses image pyramids, contour regression, non-maximum suppression and other technologies.

RetinaNet. RetinaNet [21, 22], [23] is one of the best one-stage object detection models that has proven to work well with dense and small-scale objects. This detection model has been formed by making two improvements over existing single stage object detection models – Feature Pyramid Networks (FPN) and Focal Loss. For this reason, it has become a popular object detection model to be used with aerial and satellite imagery. RetinaNet was introduced by Facebook AI Research to tackle the dense detection problem. It was needed to fill in for the imbalances and inconsistencies of the single-shot object detectors like YOLO and SSD while dealing with extreme foreground-background classes. In the face detection task, the RetinaNet model can generate an accurate rectangle face bounding box together with a 5-points facial landmark. It supports two backbone kernels: Resnet and Mobilenet. The RetinaNet model with the Resnet backbone is more accurate but relatively slow, the Mobilenet version is fast and really small.

Architecturally, RetinaNet is a composite network consisting of Backbone Network, Subnetwork for object classification, Subnetwork for object regression, illustrated as shown in the Fig. 5.

The backbone network includes 2 major components of a RetinaNet model, which are Bottom-up pathway, Top-down pathway with lateral connection. Bottom-up pathway (e.g. Resnet or Mobilenet) is used for feature extraction. It calculates the feature maps at different scales, irrespective of the input image size. The top-down path samples spatially feature maps from higher pyramid levels, and the lateral connections merge top-down and bottom-up layers with the same spatial size. Higher-level feature maps tend to have a small resolution despite being semantically stronger. Therefore, it is more suitable for detecting larger objects; in contrast, grid cells from lower-level feature maps have a higher resolution and thus detect smaller objects better. So, with a combination of the top-down pathway and its lateral connections with bottom up the pathway, which do not require much extra computation, every level of the resulting feature maps can be both semantically and spatially strong. Hence this architecture is scale-invariant and can provide better performance both in terms of speed and accuracy.

Classification subnetwork predicts the probability of an object being present at each spatial location for each anchor box and object class. A fully convolutional network (FCN) is attached to each FPN level for object classification. In Fig 5, this subnetwork combines convolutional layers of size 3x3 with 256 filters, followed by convolutional layers of size 3x3 with KxA filters. Thus, the output feature map will have dimensions WxHxKxA, where W, H are proportional to the width and height of the input feature map, K – the number of feature classes, and A – the number of anchor boxes. At last, the Sigmoid layer is used for object classification, but not the Softmax layer.

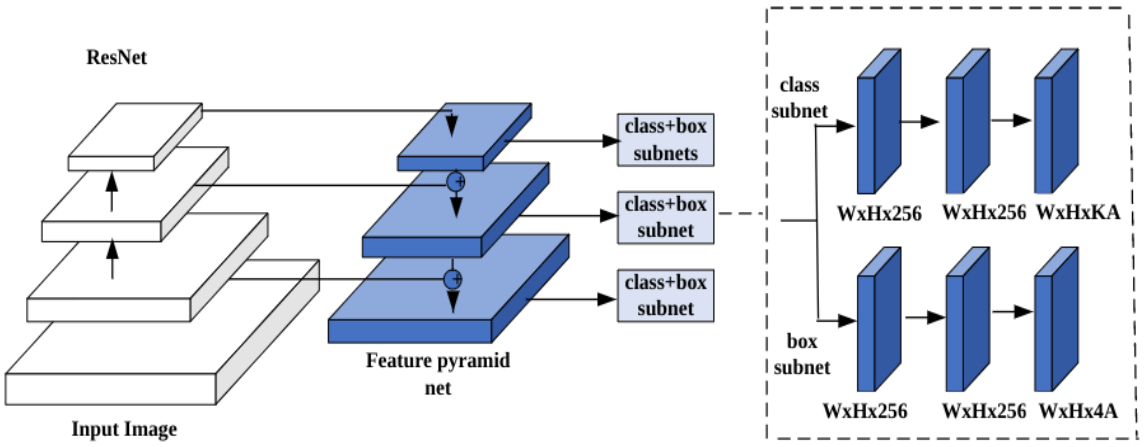

Fig. 5. **RetinaNet model architecture**
Source: **compiled by the authors**

Regression subnetwork regresses the offset for the bounding boxes from the anchor boxes for each ground-truth object. The regression subnetwork is attached to each feature map of the FPN in parallel with the classification subnet. The architecture of the regression subnet is essentially the same as that of the classification subnet; the difference is in the final convolutional layer.

This final convolution layer has size 3x3 with 4 filters so the size of the output feature map is WxHx4xA. The purpose of using the 4 filters in the final convolution layer is to localize the layer features, the regression subnetwork generates 4 numbers for each anchor that predicts relative deviations (in terms of center coordinates, width and height) between the anchor box and the ground truth box. So, the output feature map of the regression subnet has 4xA channels.

Focal loss (FL) is an enhancement over Cross Entropy Loss (CE) and is introduced to handle class imbalance when using single-stage object detection models. The existing single-stage models often suffer from foreground-background classes imbalance due to dense sampling of anchor boxes (possible object positions). So, FL is just an extension of the cross-entropy loss function, reduces the loss contribution from easy examples and increases the importance of correcting misclassified examples, that would down-weight easy examples and focus training on hard negatives.

Focal loss can be defined as (2)

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t), \qquad (2)$$

where p_t is the probability of ground truth in the softmax output distribution; α, γ are hyperparameters used to balance the loss in that α is balanced variant of the FL, a $\gamma \geq 0$ is focusing parameter

$$p_t = \begin{cases} p, when\ \gamma = 1 \\ 1-p, when\ \gamma = 0 \end{cases}.$$

FL is divided into two main parts:

– standard cross entropy $-log(p_t)$;

– modulating factor $(1-p_t)^{\gamma}$.

Adding a factor $(1-p_t)^{\gamma}$to the normal CE factor reduced the the relative loss for well-classified examples ($p_t \geq 0.5$), putting more focus on hard, misclassified examples.

YuNet (Oct 2021). OpenCV face detection equipped with face detectors such as Haar cascades and HOG detectors, which worked well for frontal faces, is no longer successful. The OpenCV version (4.5.4 Oct 2021) added model face recognition named YuNet to solve this problem. YuNet is a FD model based on CNN, developed by Chengrui Wang and Yuantao Feng [33]. It is a very light and fast model. With a model size less than MB, it can be loaded on almost any device. In the YuNet model, MobileNet is used as the backbone network, containing a total of 85000 parameters. It scores a respectable score on the WIDER Face dataset's validation set for such a lightweight model.

YuNet weight is a light-weight, fast and accurate face detection model, which achieves 0.834 (AP_easy), 0.824 (AP_medium), 0.708 (AP_hard) on the WIDER Face validation set. This model can detect faces of pixels between around 10x10 to 300x300 due to the training scheme. This ONNX model has fixed input shape, but OpenCV DNN infers on the exact shape of the input image [link for detail: *https://github.com/opencv/opencv_zoo/tree/master/models/face_detection_yunet*]. Currently, there is not much public information about the internal structure of the YuNet model. In this survey, we use YuNet to conduct experiments and compare the performance between the FD models.

FACE DATABASES AND EVALUATION PROTOCOLS

Benchmarking datasets for face detection. Currently, there are many public datasets with extremely large sizes for model training, as well as diverse test datasets. Important factors to consider when comparing datasets include: number of images and faces, range of face sizes, amount of metadata assigned to each face, and range of quality conditions the number of faces/images represented. Two of the widely used datasets, which are scales to evaluate and compare the effectiveness of Face Detection models, are Wider Face [34]. Face Detection Dataset and Benchmark (FDDB) [35].

Wider face dataset [34] is a face detection benchmark dataset, of which images are selected from the publicly available WIDER dataset, defines three levels of difficulty: Easy, Medium, and Hard. The dataset consisted of 32,203 images and labeled 393,703 faces with a high degree of variation in proportions, posture and occlusion as depicted in the sample images, divided 40 % by training set, 10 % for validation set, and 50 % for testing set.

Face detection dataset and benchmark (FDDB) [35] is a collection of labeled faces from Faces in the Wild dataset from – University of Massachusetts, Amherst, introduced – 2010. It contains a total of 5171 face annotations, where images are also of various resolutions, e.g. 363x450 and 229x410. The dataset incorporates a range of challenges, including difficult pose angles, out-of-

focus faces and low resolution. Both grayscale and color images are included.

Metrics used for evaluating face detection models.

The metrics used in Face Detection [36, 37] are the same as any other object detection problem. The popular metrics used are IoU, Precision, Recall, PR Curve, ROC Curve, AP, mAP.

Intersection over Union (IoU) [37] is a metric that quantifies the degree of overlap between two regions to evaluate the accuracy of a prediction. The measured IoU result will usually be in the range *(0,1)* with each detection having its own value. To determine whether the prediction is false or correct, we will need to rely on a given threshold, if the IoU is greater than or equal to the threshold, we will define the bounding box that will contain the object to be searched and vice versa. With the help of the IoU threshold value, we can decide whether a prediction is True Positive (TP), False Positive (FP), False Negative (FN). For the True Positive (TP) case, the predictive model is Face (Positive) and in fact it is Face. For False Positive, the predictive model is a Face (Positive) but in reality that bounding box does not contain any Face to be determined. In the case of False Negative, the Bounding box is determined to not contain Face but in fact that is false.

Precision. Accuracy measures the proportion of positives that are predicted to be correct. Precision will calculate the percentage of True Positives in the total number of detected times (total predictions) according to the following formula (3):

$$P = TP / (TP + FP) = TP / Total\ Prediction. \quad (3)$$

Recall is a parameter that represents the ratio of correct predictions to the total number of ground truths (formula 4).

$$R = TP / (TP + FN) = TP / Total\ Ground\ Truths, \quad (4)$$

Precision-Recall Curve (PR-Curse) is a graph with Precision rate on the y-axis and Recall rate on the x-axis (Fig. 6). It shows the precision as a recall function for all the different threshold values.

Receiver operating characteristic (**ROC Curve**) [38] is a graph that represents the performance of a model as a function of its threshold (similar to the precision-recall curve). It basically shows the Recall with False Positive Rate (FPR) for different threshold values, as shown in Fig. 7.

Average Precision (AP) [39] is not the mean of precision, it's the area under the PR Curse. The area under the curve used to summarize the performance of a model becomes a general measure of prediction accuracy.

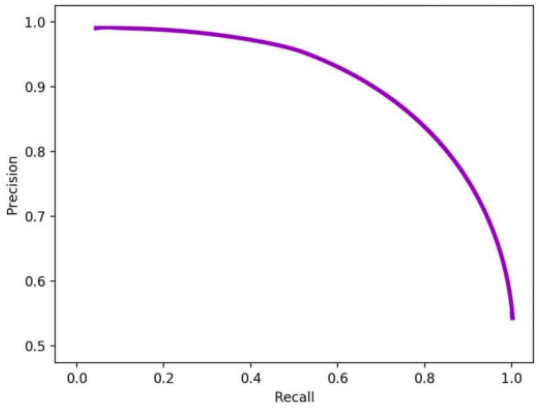

Fig. 6. **Precision-Recall Curve**
Source: **compiled by the [35]**

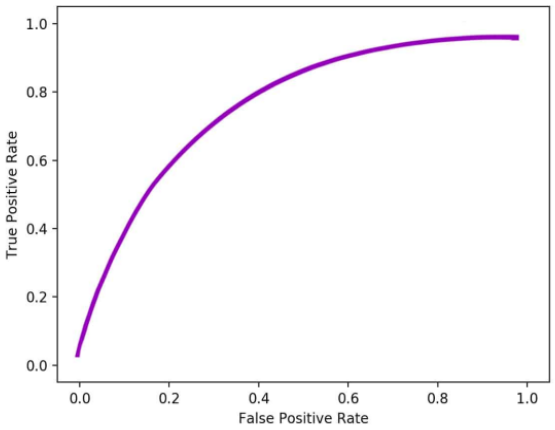

Fig. 7. **ROC-Curse**
Source: **compiled by the [35]**

Mean Average Precision (mAP) is the average of AP over all classes detected in multilayer object detection, calculated by the below formula (5):

$$mAP = 1/n * sum(AP), \quad (5)$$

where n is the number of classes; AP id accuracy calculated separately for each class.

AP, mAP are the current benchmark metrics used to evaluate the robustness of object detection models, encapsulates the tradeoff between precision and recall and maximizes the effect of both metrics. Object detection systems make predictions in terms of a bounding box and a class label. The relationship between precision – recall helps mAP assess the accuracy of the classification task, while for each bounding box, IoU is used to measure an overlap between the predicted bounding box and the ground truth bounding box. In addition, the precision and recall values change when the IoU threshold changes (the threshold to predict which class a bounding box is), therefore, to evaluate the accuracy of the model, it is necessary to calculate Precision and Recall at a

specified IoU value. For the task of the accuracy evaluation of these FD models, we chose to survey the AP values at the threshold IoU=0.5 (AP@0.5).

EXPERIMENTAL PERFORMANCE

We have conducted an experimental test of the face detection task for FD models namely SSD, MTCNN, RetinaFace_ResNet5, and YuNet with the WIDER FACE dataset with 3 levels – easy, medium, and hard.

System Configuration

– Processor – AMD® Ryzen 5 5600h with radeon graphics × 12.

– GPU – NVIDIA Corporation TU117M [GeForce GTX 1650 Mobile / Max-Q].

– RAM – 16,0 GB

– OS Ubuntu 22.04

The obtained results are processed to compare the processing speed of Face Detectors (Table 1; Fig. 8), and compare the AP accuracy at IoU=0.5 (AP@0.5), as shown in Table 1, Fig. 9.

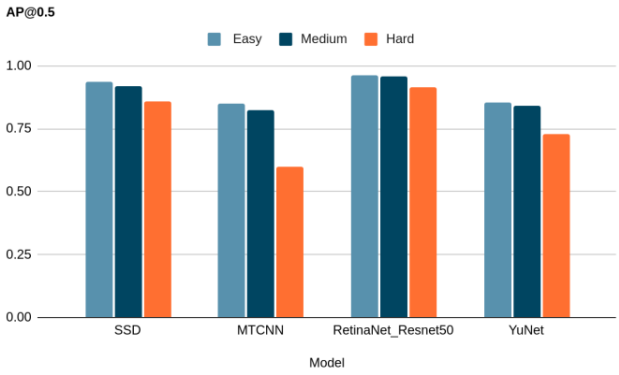

Fig. 8. **Model accuracy diagram, based on AP@0.5 with WIDER Face (Easy, Medium, Hard)**
Source: **compiled by the authors**

In Fig. 10 is shown an example for face detection using SSD, MTCNN, RetinaNet_Resnet50, YuNet models with an image from the supermarket camera in the WIDER Face dataset. Observing the results obtained, we can see that: RetinaNet_Resnet50 model recognizes all faces in the frame; Yunet model is very good, but one small face is missing in the frame; the other two models are hardly effective at detecting small and large faces.

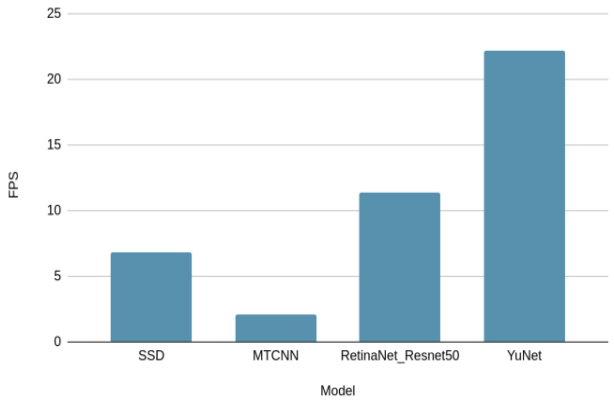

Fig. 9. **Processing speed diagram, based on FPS**
Source: **compiled by the authors**

CONCLUSION

Face detection is one of the first tasks in today's face-related automated systems. The systems have been developed and applied to life and high industries such as security surveillance by camera, time attendance system, camera attendance, driver monitoring system in autonomous, medical assistance system via camera, ... When applied in practice, especially real-time systems, the problems from the results of the FD model are the False Positive outputs, in addition, the small object detection results also bring low accuracy, so now FD model enhancement is always an urgent task and should be prioritized. In the survey of FD models, we analyzed the architecture of different one-stage deep learning models and conducted experiments, evaluations, and comparisons of accuracy and performance of the FD models to make a suitable choice to solve the Face detection task.

Table 1. **Comparison of the models, based on FPS and AP@0.5 with Multi-task cascaded convolution neural networks. Wider Face (Easy, Medium, Hard)**

Model	FPS (image 320x320)	AP@0.5 WIDER Face Easy	AP@0.5 WIDER Face Medium	AP@0.5 WIDER Face Hard
SSD	6.81	0.935	0.921	0.858
MTCNN	2.11	0.848	0.825	0.598
RetinaNet_Resnet50	11.35	0.963	0.956	0.914
YuNet	22.22	0.856	0.842	0.727

Source: **compiled by the authors**

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

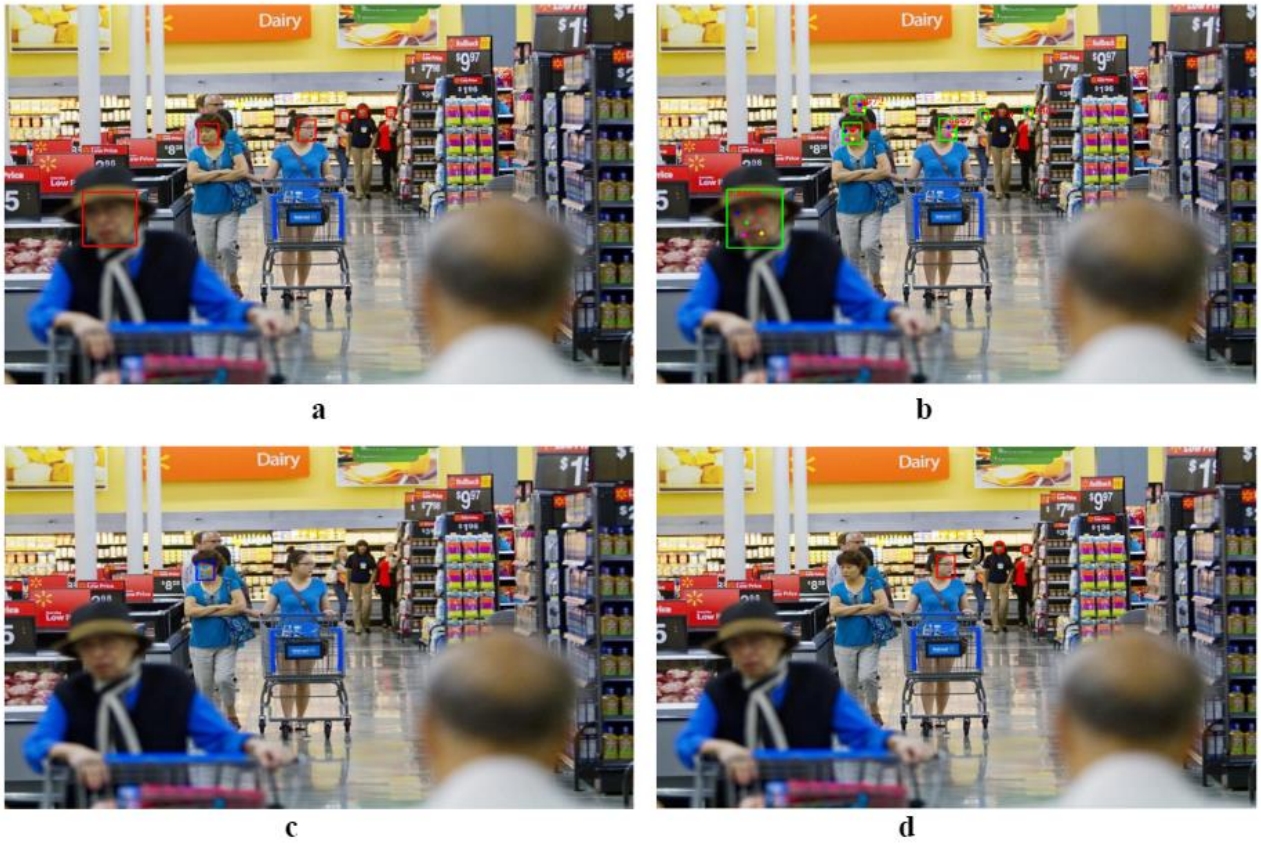

Fig. 10. **An example for face detection using:**
a – Retinanet_Resnet50 model; b – YuNet model;
c – Multi-task cascaded convolution neural networks model; d - SSD model
Source: **compiled by the authors**

Usually choosing the most suitable model will depend on the requirements of the particular application. It is possible to rely on 3 conditions to determine model selection: detection accuracy, detection speed, balance between accuracy and speed. If the system doesn't need to factor in processing speed in real time inference but focuses on best-in-class detection accuracy and doesn't want to miss any faces, then RetinaNet-Resnet50 is the best choice right now.

However, now applications in real-time conditions in addition to accuracy requirements must also ensure processing speed requirements, so the choice needs to balance between accuracy and speed. With this criterion, YuNet is a model that responds well. When it comes to practical application in our applications, one problem affecting the accuracy of the system that the two selected models are still unable to guarantee is the problem of detecting small faces in the frame. Therefore, through this survey, we propose a direction to improve FD face recognition model based on RetinaNet structure with the goal of ensuring accuracy and reducing processing time.

REFERENCES

1. Chai, J., Zeng, H., Li, A. & Ngai Eric W.T. “Deep learning in computer vision: A critical review of emerging techniques and application scenarios”. *Machine Learning with Applications*. 2021; 6: 100134. DOI: https://doi.org/10.1016/j.mlwa.2021.100134.

2. Tian, Y., Pan G. & Alouini M.-S. “Applying deep-learning-based computer vision to wireless communications: methodologies, opportunities, and challenges.” *IEEE Open Journal of the Communications Society*. 2020; 2: 132–143, https://www.scopus.com/authid/detail.uri?authorId=35194164800. DOI: https://doi.org/10.1109/OJCOMS.2020.3042630.

3. Parkhi, O. M., Vedaldi, A. & Zisserman, A. “Deep face recognition.” *British Machine Vision Conference*. 2015. p. 41.1–41.12. DOI: https://doi.org/10.5244/C.29.41.

4. Jahan, I., Uddin, K. M. A., Murad, S. A., Miah, M. S. U., Khan, T. Z., Masud, M., Aljahdali, S. & Bairagi, A. K. “4D. A real-time driver drowsiness detector using deep learning”. *Electronics.* 2023, 12 (1): 235. DOI: https://doi.org/10.3390/electronics12010235.

5. Verma, B. & Choudhary A. “Deep learning based real-time driver emotion monitoring”. *IEEE International Conference on Vehicular Electronics and Safety*. Madrid: Spain. 2018. p. 1–6. DOI: https://doi.org/10.1109/ICVES.2018.8519595.

6. Le, Q. T., Antoschuk, S. G., Tran, T. V., Nguyen, T. K. T. & Dang, N. C. “Automated student attendance monitoring system in the classroom based on convolution neural networks”. *Applied Aspects of Information Technology*. 2020; 3 (3): 179–190. DOI: https://doi.org/10.15276/aait.03.2020.6.

7. Feng Zhao, Jing Li, Lu Zhang, Zhe Li & Sang-Gyun Na. “Multi-view face recognition using deep neural networks”. *Future Generation Computer Systems*. 2020; 111: 375–380. DOI: https://doi.org/10.1016/j.future.2020.05.002.

8. Qasim, H. S. A., Shahzad M. & Fraz M. M. “Deep learning for face detection: recent advancements”. *International Conference on Digital Futures and Transformative Technologies (ICoDT2).* 2021. p. 1–6. DOI: https://doi.org/10.1109/ICoDT252288.2021.9441476.

9. Zhou, Y., Liu D. & Huang T. S. “Survey of face detection on low-quality images.” *13th IEEE International Conference on Automatic Face & Gesture Recognition (FG).* 2018. p. 769–773, https://www.scopus.com/authid/detail.uri?authorId=56413153000.DOI:https://doi.org/10.1109/FG.2018.00121.

10. Tryhuba, A., Tryhuba, I. & Bashynsky, O. “Conceptual model of management of technologically integrated industry development projects”. *Proceedings of the 15th International Scientific and Technical Conference on Computer Sciences and Information Technology*. 2020. p. 155–158, https://www.scopus.com/authid/detail.uri?AuthorId=57205225539.DOI: https://doi.org/10.1109/CSIT49958.2020.9321903.

11. O’Mahony, N. et al. “Deep learning vs. Traditional computer vision”. *Computer Vision Conference (CVC).* 2019; 943: 128–144. DOI: https://doi.org/10.1007/978-3-030-17795-9_10.

12. Gangopadhyay, I., Chatterjee, A., & Das, I. “Face detection and expression recognition using haar cascade classifier and fisherface algorithm”. *Recent Trends in Signal and Image Processing*. 2019; 922: 1–11. DOI: https://doi.org/10.1007/978-981-13-6783-0_1.

13. Adouani, A., Ben Henia, W. M. & Lachiri, Z. “Comparison of Hear-like, HOG and LBP approaches for face detection in video sequences”. *16th International Multi-Conference on Systems, Signals & Devices (SSD)*. Istanbul: Turkey. 2019. p. 266–271. DOI: https://doi.org/0.1109/SSD.2019.8893214.

14. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.” *Computer Vision and Pattern Recognition.* 2014. DOI: https://doi.org/10.48550/arXiv.1312.6229.

15. Farfade, S. S., Saberian, M. J. & Li, L. “Multi-view face detection using deep convolutional neural networks”. *ICMR'15: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. 2015. p. 643–650. DOI: https://doi.org/10.1145/2671188.2749408.

16. Zhang, C. & Zhang, Z. "Improving multitier face detection with multi-task deep convolution neural networks". *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs. CO, USA*, 2014, p. 1036–1041. DOI: https://doi.org/10.1109/WACV.2014.6835990.

17. Zhang, H. & Cloutier, R. S. “Review on one-stage object detection based on deep learning”. *EAI Endorsed Trans. e Learn.* 2022; 7. DOI: https://doi.org/10.4108/eai.9-6-2022.174181.

18. Marcetic, D., Hrkać, T., & Ribaric, S. “Two-stage cascade model for unconstrained face detection”. *2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE).* 2016. p.1–4. DOI: https://doi.org/10.1109/SPLIM.2016.7528404.

19. Liu, W., Leibe, B., Matas, J., Sebe, N. & Welling, M. “SSD: Single shot MultiBox detector”. *Computer Vision – ECCV 2016. Lecture Notes in Computer Science.* 2016; 9905: 21–37. DOI: https://doi.org/10.1007/978-3-319-46448-0_2.

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

21. Wibowo, M. E., Ashari, A., Subiantoro, A. & Wahyono, W. “Human Face Detection and Tracking Using RetinaFace Network for Surveillance Systems”. *IECON – 47th Annual Conference of the IEEE Industrial Electronics Society.* 2021. p. 1–5. DOI: https://doi.org/10.1109/IECON48115.2021.9589577.

22. Chen, Y., Wu, Z., Peng, H., Zhou, C., Yan, Z., & Huang, Y. “Face detection algorithm based on improved Retinaface”. *China Automation Congress (CAC)*. 2021. p. 4542–4547. DOI: https://doi.org/10.1109/cac53003.2021.9728659.

23. Chen, J., Ma, H. & Li, L. “IRNet: An improved RetinaNet model for face detection”. *7th International Conference on Image, Vision and Computing (ICIVC)*. 2022. p. 129–134. DOI: https://doi.org/10.1109/ICIVC55077.2022.9886975.

24. Ren, S., He, K., Girshick, R. & Sun, J. “Faster R-CNN: Towards real-time object detection with region proposal networks”. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 2017; 39 (06): 1137–1149. DOI: https://doi.org/10.1109/TPAMI.2016.2577031.

25. Jiang, H. & Learned-Miller E. G. “Face detection with the faster R-CNN”. *12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. 2017. p. 650–657. DOI: https://doi.org/10.1109/FG.2017.82.

26. Lin, T.-Y., Dollár, P., Ross B. Girshick, K. He, Hariharan, B. & Belongie S. J. “Feature pyramid networks for object detection”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. p. 936–944. DOI: https://doi.org/10.1109/CVPR.2017.106.

27. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X. & Li, S. “SFD: Single shot scale-invariant face detector”. *IEEE International Conference on Computer Vision (ICCV)*. 2017. p. 192–201. DOI: https://doi.org/10.1109/ICCV.2017.30.

28. Najibi, M., Samangouei, P., Chellappa, R. & Davis, L. “SSH: Single stage headless face detector”. *IEEE International Conference on Computer Vision (ICCV)*. 2017. p. 4885–4894. DOI: https://doi.org/10.1109/ICCV.2017.522.

29. Fan, X., Jiang, M. & Yan, H. “A deep learning based light-weight face mask detector with residual context attention and gaussian heatmap to fight against COVID-190”. *IEEE Access*. 2021; 9: 96964–96974. DOI: https://doi.org/10.1109/ACCESS.2021.3095191.

30. Du, L., Zhang, R. & Wang, X. “Overview of two-stage object detection algorithms”. *Journal of Physics: Conference Series.* 2020; 1544 (1): 012033. DOI: https://dx.doi.org/10.1088/1742-6596/1544/1/012033.

31. Shi, L., Xu, X. & Kakadiaris, I. A. “SSFD+: A robust two-stage face detector”. *IEEE Transactions on Biometrics, Behavior, and Identity Science*. 2019; 1 (3): 181–191. DOI: https://doi.org/10.1109/TBIOM.2019.2928118.

32. Yang, Z., Ge, W. ^ Zhang, Z. “Face recognition based on MTCNN and integrated application of FaceNet and LBP method”. *2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*. 2020. p. 95–98. DOI: https://doi.org/10.1109/AIAM50918.2020.00024.

33. Feng, Y. “OpenCV zoo and benchmark. YuNet”. Available from: https://github.com/opencv/opencv_zoo/tree/master/models/face_detection_yunet. – [Accessed: February, 2023].

34. Yang, S., Luo, P., Loy, C. C. & Tang, X. “WIDER FACE: A face detection benchmark”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016. p. 5525–5533. DOI: https://doi.org/10.1109/CVPR.2016.596.

35. Jain, V, & Learned-Miller, E. G. “FDDB: A benchmark for face detection in unconstrained settings”. 2010. – Available from: http://vis-www.cs.umass.edu/fddb/fddb.pdf. **–** [Accessed: December 2010].

36. Abaza, A. A. & Harrison, M. A. F. & Bourlai, T. “Quality metrics for practical face recognition”. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR).* Tsukuba: Japan. 2012. p. 3103–3107.

37. Peng, H. & Yu, S. “A Systematic IoU-Related method: Beyond simplified regression for better localization”. *IEEE Transactions on Image Processing*. 2021; 30: 5032–5044. DOI: https://doi.org/10.1109/TIP.2021.3077144.

38. Prati, R. C., Batista, G. & Monard, M. C. “Evaluating classifiers using ROC curves”. *IEEE Latin America Transactions.* 2008; 6: 215–222. DOI: https://doi.org/10.1109/TLA.2008.4609920.

39. Shah, D. “Mean Average Precision (mAP) Explained: Everything you need to know”. 2020. **–** Available from: https://www.v7labs.com/blog/mean-average-precision#h4. – [Accessed: March 2022].

Conflicts of Interest: the authors declare no conflict of interest

Received 27.03.2023
Received after revision 16.05.2023
Accepted 02.06.2023

DOI: https://doi.org/10.15276/aait.06.2023.15
УДК 004.93

Огляд з детектування обличь на основі глибокого навчання

Тхе Вінь Чан[1)]
ORCID: https://orcid.org/0000-0002-4241-1065; vinhtt@ut.edu.vn. Scopus ID: 288641
Тхі Кхань Тієн Нгуєн[1)]
ORCID: https://orcid.org/0000-0001-5379-7226; tienntk@ut.edu.vn
Кім Тхань Чан [1)]
ORCID: https://orcid.org/0000-0002-4241-1065; tkthanh2011@gmail.com
[1)] Університет транспорту міста Хошимін, місто Хошімін. В'єтнам

АНОТАЦІЯ

У статті основна увага приділяється огляду моделей виявлення обличчя, що ґрунтуються на глибокому навчанні, зокрема огляду різних одноетапних моделей, з яких можна вибрати відповідну модель розпізнавання осіб, і в той же час пропонується напрям удосконалення моделі виявлення обличчя відповідно до фактичних вимог прикладних систем комп'ютерного зору. Моделі виявлення обличчя, які були проведені, включають SSD, MTCNN, RetinaNet, YuNet у наборі даних Wider Face. Завдання під час опитування – структурне дослідження вибраних моделей, проведення експериментальних досліджень для оцінки точності та продуктивності цих моделей. Для оцінки та надання моделі виявлення обличчя, що відповідає вимогам, використовуються два показники - AP для оцінки точності та FPS для оцінки продуктивності. Для наших додатків у режимі реального часу на системах камер, пов’язаних із обличчям, таких як система моніторингу водія, система безпеки супермаркетів (попередження про крадіжки в магазинах, попередження про порушення порядку), система відвідування, часто вимагає швидкої обробки, але все одно забезпечує точність. Моделі, які зараз застосовуються в нашій системі, як-от Yolos, RetinaNet_MobileNet, SSD, гарантують обробку в реальному часі, але більшість із цих моделей мають труднощі з виявленням маленьких облич у кадрі та випадках, що містять контексти, які легко прийняти за обличчя людини. У той же час модель RetinaNet_Resnet50 забезпечує найвищу точність, особливо для забезпечення виявлення маленьких облич у кадрі, але час обробки більший. Тому за допомогою цього опитування ми пропонуємо напрямок удосконалення моделі розпізнавання обличчя на основі структури RetinaNet з метою забезпечення точності та скорочення часу обробки.

Ключові слова: Виявлення обличчя; одноступеневий детектор; двоступеневий детектор; глибоке навчання; детектор одиночного пострілу; багатозадачні каскадні згорткові нейронні мережі MTCNN; RetinaNet, YuNet

ABOUT THE AUTHORS

Tran The Vinh Tran – Doctor of Philosophy, Senior Lecturer of the Department of Information Technology. Ho Chi Minh City University of Transport/. Ho Chi Minh City. Vietnam
ORCID: https://orcid.org/0000-0002-4241-1065; vinhtt@ut.edu.vn. Scopus ID: 288641
Research field: Data science; cyber security

Чан Тхе Вінь Чан – доктор філософії, старший викладач кафедри Інформаційних систем. Університет транспорту міста Хошимін, місто Хошімін. В'єтнам

Nguyen Thi Khanh Tien – Doctor of Philosophy, Senior Lecturer of Department of Information Technology (Ho Chi Minh City University of Transport, Ho Chi Minh City, Vietnam), Senior Lecturer of Department of Information System (Odessa Polytechnic National University, Odessa, Ukraine)
ORCID: https://orcid.org/0000-0001-5379-7226; tienntk@ut.edu.vn
Research field: Data science; artificial intelligence methods and systems

Нгуєн Тхі Кхань Тієн – доктор філософії, старший викладач кафедри Інформаційних технологій (Ho Chi Minh City University of Transport. Ho Chi Minh City. Vietnam), старший викладач кафедри Інформаційних систем. Національний університет «Одеська політехніка». Одеса, Україна

Tran Kim Thanh – Doctor of Philosophy, Senior Lecturer of Department of Information Technology, Ho Chi Minh City University of Transport. Ho Chi Minh City. Vietnam
ORCID: https://orcid.org/0000-0002-4241-1065; tkthanh2011@gmail.com
Research field: Statistical probability; data science

Чан Кім Тхань – доктор філософії, старший викладач. Університет транспорту міста Хошимін, місто Хошімін, В'єтнам