International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 40, No. 5, October, 2023, pp. 2029-2039
Journal homepage: http://iieta.org/journals/ts

Real-Time Detection and Identification of Suspects in Forensic Imagery Using Advanced YOLOv8 Object Recognition Models

Serkan Karakuş[1*], Mustafa Kaya[2], Seda Arslan Tuncer[3]

[1] Siemens R&D Center, Kartal, Istanbul 34100, Turkey
[2] Department of the Digital Forensic Engineering, Fırat University, Elazığ 23100, Turkey
[3] Department of the Software Engineering, Fırat University, Elazığ 23100, Turkey

Corresponding Author Email: serkan.karakus@siemens.com

https://doi.org/10.18280/ts.400521

Received: 17 April 2023
Revised: 29 July 2023
Accepted: 9 September 2023
Available online: 30 October 2023

Keywords:
digital forensic, face detection and recognition, real-time object detection, YOLOv8

ABSTRACT

Rapid advancements in artificial intelligence, machine learning, deep learning, coupled with easy access to high-capacity processing hardware, expansive organized datasets, and the evolution of artificial intelligence algorithms, have extensively influenced numerous fields. Digital Forensics is one such discipline where the application of artificial intelligence has been significantly amplified in recent years. The analysis of extensive image and video files derived from forensic evidence presents challenges in terms of time efficiency and accuracy. To surmount these challenges, artificial intelligence models can be employed to perform identification and classification processes on these data, thus expediting the resolution of forensic cases with enhanced precision. In the current study, state-of-the-art pre-trained YOLOv8 object recognition models - nano, small, medium, large, and extra-large - were utilized. These models were trained on the Wider-Face dataset with the objective of identifying suspects from images and videos sourced from digital materials in the field of digital forensics. The models achieved mean Average Precision (mAP) values of 97.513%, 98.569%, 98.763%, 98.775%, and 99.032% respectively. The YOLOv8 architecture demonstrated superior performance, outperforming the YOLOv5 architecture by a margin of 7.1% to 8.8%. To aid digital forensic experts in the detection and identification of suspicious individuals, a desktop application capable of real-time image analysis was developed.

1. INTRODUCTION

The contemporary advancement and proliferation of technology enable individuals to generate substantial volumes of data using portable electronic devices. With the surge in digital data production and storage capacities, coupled with increased accessibility, a single device can store tens of thousands of image and video files. However, the storage of such voluminous data presents significant challenges in the analysis of information obtained from seized devices [1, 2].

Digital forensics, a process aimed at analyzing data derived from these devices, is defined by the Digital Forensic Research Workshop (DFRWS) as the application of scientifically derived and proven methods to preserve, collect, validate, identify, analyze, interpret, document, and present digital evidence. This evidence, derived from digital sources, aids in reconstructing events deemed criminal or helps anticipate unauthorized actions that appear to undermine planned operations. An illustration of the Digital Forensics Process is depicted in Figure 1.

In a typical forensic case, electronic evidence associated with a suspect is first identified and preserved. Forensic replicas of the preserved evidence are then created with image acquisition hardware and software. Subsequently, facts related to the forensic event are unveiled and analyzed based on the data extracted from the forensic image. The findings are then compiled into a report for judicial authorities.

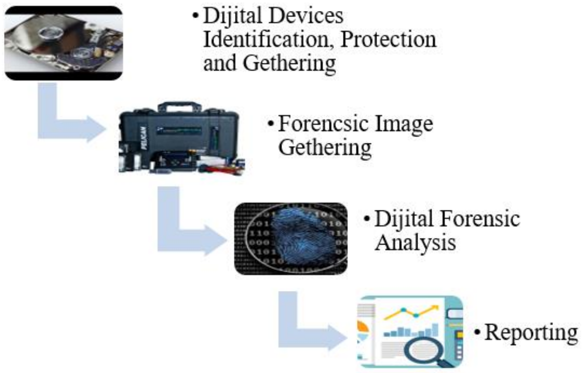

•Dijital Devices Identification, Protection and Gethering
•Forencsic Image Gethering
•Dijital Forensic Analysis
•Reporting

Figure 1. Digital forensic process

1.1 Problem statement

During the analysis phase of digital forensics, evidence derived from the forensic copy must be analyzed and linked to the forensic case. The evidence is assessed using forensic evidence analysis software, which includes the examination of forensic evidence by the suspect or other persons, as well as the inspection of deleted and hidden files visible to the analysis software in the forensic image. Forensic evidence analysis software can search for, recover, and index deleted or hidden files from a forensic image, extract data, and present it in a format accessible to an expert.

However, during the forensic analysis phase, multimedia data can only be evaluated according to metadata (file name, file index) by forensic software. Given the vast volume of multimedia data associated with forensic evidence, it is virtually impossible for a group of digital forensic experts to analyze them accurately and promptly. In a study conducted by Ferreira et al. [3], a forensic IT expert could scrutinize more than 100,000 images during the forensic image examination process, spanning 6-18 months. An examination of a database containing over 300,000 images and more than 1,100 video files yielded only 148 images of illegal content.

1.2 Motivation

Recent data suggest that at least one digital document has been obtained in every judicial case. In a report by Celebrite, a company specializing in forensic evidence analysis products, it was found that electronic files, digital photos, emails, mobile phones, digital videos, internet history, and social media data were obtained from digital materials in a forensic case at varying percentages [4]. The report also highlighted that video and digital cameras are among the fastest-growing sources of data. Respondents reported spending between 1 and 10 hours a week analyzing images and videos, often leading to psychological issues. The report further projected 8.8 million mobile users by 2024 and that each forensic case seizes between 2 and 10 devices.

Given the escalating volumes of data, the limitations of forensic analysis software, and the high volumes of data in diverse formats, it is impossible for experts to examine them in detail. This work proposes an application interface equipped with artificial intelligence-supported facial recognition models to assist digital forensics experts in analyzing and reporting thousands of data, including video and photo files derived from digital evidence.

In recent years, single-stage object recognition algorithms such as object YOLO have produced real-time and high-performance artificial intelligence models. This study demonstrates the successful application of YOLOv8 in detecting faces from images and videos in the forensic evidence analysis phase. Furthermore, the features of a face can be easily revealed with the weights of the pre-trained VGGFace2 architecture, which was previously trained on a large face dataset. When compared to the robustness of classical face recognition algorithms (HOG, LBPH, EigenFace, FisherFace, ViolaJones, etc.), it is evident that the use of artificial intelligence models that extract facial features will enhance their success.

1.3 Contribution

This study merges image processing methods in forensic informatics with artificial intelligence. With the proposed user interface operating in the background, digital forensic experts can generate reports by analyzing suspicious individuals in thousands of image and video files derived from electronic evidence, accomplishing this task with high success rates and real-time speed. The face detection model file trained on state-of-the-art object detection architectures significantly aids this process.

2. RELATED WORK

The interdisciplinary application of artificial intelligence (AI) has been recognized for its efficacy, particularly in addressing analytical challenges in digital forensics. Combining AI methods with content analysis of multimedia data obtained from forensic evidence has proven to be a promising approach. This section reviews a selection of studies that have harnessed AI methodologies to solve current problems in digital forensics.

Riadi et al. [5] investigated the digital forensics of the Signal Messenger application on Android devices, particularly during the rise of cybercrimes in the COVID-19 era. Using tools like Belkasoft, Magnet AXIOM, and MOBILedit Forensic Express within the Digital Forensics Research Workshop (DFRWS) framework, they sought to uncover digital evidence. Their research pinpointed several types of evidence, including chats, media, and account data. Among the tools, Belkasoft Evidence Center demonstrated superior accuracy at 78.69%. The findings offer valuable insights for future forensic research on the Signal Messenger application. In a related study, Korkmaz and Boyacı [6] proposed a hybrid speaker recognition model using long short-term memory (LSTM) networks. Their model demonstrated the potential to be applied to audio files obtained in digital forensics for content analysis.

Artificial intelligence has also been employed to analyze social media content. Abebaw et al. [7] used multi-channel convolutional neural networks (CNNs) to extract features of hate speech from social media, with SVM used for classification. This methodology facilitated anomaly detection from social media data. Channabasava and Raghavendra [8] developed a consensus-based ensemble model for social media link prediction using an array of features. Through methods like cross-correlation and PCA, they achieved an accuracy of up to 97%. Incorporating logistic regression, decision trees, and deep-neuro algorithms, their model surpassed other methods with a link-prediction accuracy of 98%.

Digital image analysis in forensics has also received attention, with Piva [9] focusing on procedures such as copy-move forgery, resampling detection, image enhancement, and compression. CNNs have been extensively used in image forensics for feature extraction and classification over the past five years. For instance, CNN architectures have been successfully utilized for steganography [10], watermarking [11], SCI-camera information detection [12], and copy-move forgery [13].

An emerging area of interest in recent years is the application of object detection and recognition tasks to images of forensic evidence. One such innovative approach, proposed by Javed and Jalil [14], is a byte-level object identification method for the forensic examination of digital images. This method deciphers the byte code of each pixel in an image and identifies objects based on their unique byte code.

Face recognition tasks have also been broadly explored in computer science and digital forensics. Zafeiriou et al. [15] provided a comprehensive discussion on deep learning-based face recognition technologies, datasets, deep learning architectures, and performance, along with future projections. Viola and Jones [16] examined face detection and recognition tasks, exploring the challenges, algorithmic use, and success rates. They also discussed strategies to reduce the False Positive Rate (FPR) and to develop real-time applications.

Historically, Bledsoe's seminal work [17] was a pivotal contribution to the development of face recognition technology. Bledsoe proposed the "model method," in which a mathematical model of a person's face would be constructed and compared with other faces. Following this, several classical face recognition applications were developed, including EigenFace [18], FisherFace [19], BayesianFace [20], MetaFace [21], LaplancianFace [22], and Support Vector Machine (SVM) [23].

The advent of deep learning algorithms and advanced graphics cards, particularly after 2010, has significantly impacted face recognition technology. Krizhevsky et al.'s success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with the AlexNet network was a breakthrough [24]. This network used deep neural networks and was trained on graphics cards with parallel processing capability.

Subsequent studies have utilized CNN architectures for face detection tasks. For example, the DeepFace model by Taigman et al. [25] achieved a performance comparable to human image detection, with a success rate of 97.53%. The DeepFace model was created by training a 9-layer CNN model on four million images. Sun et al. [26] demonstrated that the DeepID model, trained with a CNN architecture, could perform face recognition in 10,000 classes.

More recent models such as FaceNet [27], VGGFace [28], VGGFace2 [29], and ArcFace [30] have utilized CNN architectures. These models are trained on face data and have weight files that can easily extract unique features of face data. In the current study, artificial intelligence-supported models for next-generation object recognition and classification tasks are used. These models address the limitations of classical algorithms, such as low sensitivity, scalability issues, multiple face detection, low sensitivity to facial changes, and the inability to detect deep features.

In conclusion, the application of AI in digital forensics presents a promising avenue for future research. The current study attempts to further this line of inquiry by proposingan AI-based model for forensic facial recognition that leverages the strengths of deep learning algorithms while mitigating their limitations. This model, which will be discussed in detail in the subsequent sections, employs advanced CNN architectures and is trained on extensive, diverse datasets. It aims to significantly improve the accuracy, scalability, and robustness of facial recognition in forensic contexts.

The focus on facial recognition is motivated by its immense potential in various forensic scenarios, including identity verification, criminal investigations, surveillance, and even historical research. By building upon the existing body of knowledge and employing innovative AI techniques, this study aims to make a meaningful contribution to the field of digital forensics and shape the next generation of AI-assisted forensic tools.

3. MATERIALS AND METHODS

This study will present a software that will be made available to digital forensic experts that generates a report by detecting and identifying one or more face images given as input on image and video files obtained from forensic evidence with the help of an object recognition model file trained with the YOLO (You Only Look Once) architecture.

3.1 Materials

3.1.1 Dataset

In this study, the Wider Face dataset was used to train the face detection model. The Wider-Face dataset is publicly available on the github platform [31]. The dataset of 32,203 images contains 393,703 labeled face images. The dataset was also classified according to 61 event classes. Figure 2 shows sample images from the Wider-Face dataset.

Figure 2. Wider face dataset sample

3.1.2 Hardware

In order to increase the success rate of artificial intelligence models, training should be done with large volumes of data. During the training of these models, hardware such as the CPU (Central Processing Unit) cannot provide sufficient performance. Therefore, it is necessary to train on hardware capable of parallel processing, such as a GPU (Graphic Processing Unit). In addition, in order to update the weights by forwarding the images in the data set once to the neural network, these images must be loaded on RAM and then transferred to the GPU. Therefore, face detection model training was performed on a workstation with an i7 Intel xxx GH processor, Nvidia 2070 SUPER (8 GB), and 32 GB DRR6 RAM.

3.1.3 Programming language and framework

Currently, the Python programming language is widely used by researchers for object recognition models and training. The YOLO architecture uses the Pytorch (Deep Learning Framework) written in the Python programming language. In order to make the face detection and face recognition models easy to be use by the end user, an interface design was made with the PyQt5 framework.

3.2 Object detection framework and YOLO

Object detection is a computer vision task that involves identifying and localizing objects of interest within an image or a video sequence. The goal is to draw bounding boxes around the objects in the image and label them with their corresponding classes. Object recognition algorithms can be defined as one-stage or two-stage. The reason for making this distinction is the identification of possible locations where an object may be and whether these possible locations can be identified at the same time. In the literature, the most successful implementation of single-stage algorithms is YOLO, while the most successful implementations of two-stage algorithms are seen in R-CNN (Regional Convolutional Neural Network) architectures.

Two-Stage Object Recognition Algorithms: In 2012, with the use of CNN architectures such as AlexNet in the ILSVRC competition, object recognition tasks started to be implemented with deep learning architectures.

The R-CNN architecture was proposed by Girshick et al. [32] and consists of three stages.

- The model file makes multiple (about 2000) proposals for region of objects to be recognized on the input image.

- A fully connected neural network layer extracts the features of each recommendation domain.
- The last stage consists of classifying the features from the previous layer with the SVM algorithm.

In the R-CNN architecture, the fixed size of the inputs caused the two-stage method recognition process to be slow. In the R-CNN architecture, the classification of the propasal regions separately increased the computational cost. Not suitable for real-time object recognition applications. Fast R-CNN architecture was proposed by Girshick [33] to solve the speed problem. Fast R-CNN reduces computational costs as it learns to detect and classify the spatial positions of objects together. Fast R-CNN architecture is 100 times faster than R-CNN architecture. The Faster R-CNN architecture was proposed by Ren et al. [34]. Previous R-CNN architectures could not achieve successful results in real-time applications due to computational cost and speed issues. As a solution, Faster R-CNN implemented Regional Proposal Network (RPN) networks, which share convolution layers with the feature extraction network and bring marginal computational costs reduction for computing object proposals region. However, all R-CNN architectures with high accuracy in performing object recognition tasks are too slow to be used in real-time object detection applications. Single-stage object recognition algorithms have been developed to solve this problem.

Single-Stage Object Recognition Algorithms: The fact that R-CNN architectures are slow in object recognition applications may be the reason for the development of single-stage object recognition algorithms. In 2016, the YOLO architecture was introduced by Redmon et al. [35] for real-time object detection. The YOLO algorithm has shown that it can solve the problem of extracting and classifying proposal regions in an R-CNN architecture with a single network. In the following years, Yolov2 [36], Yolov3 [37], Yolov4 [38], Yolov5 [39], Yolov6 [40], YoloR, YoloX, Yolov7 [41], and Yolov8 [42] versions were introduced. The parameter-mAP comparison of the Yolov8 algorithm between other YOLO models is shown in Figure 3.

Ultralytics YOLOv8 is the latest version of the YOLO object detection and image segmentation model. As a cutting-edge, state-of-the-art (SOTA) model, YOLOv8 builds on the success of previous versions, introducing new features and improvements for enhanced performance, flexibility, and efficiency. YOLOv8 is designed with a strong focus on speed, size, and accuracy, making it a compelling choice for various vision AI tasks. It outperforms previous versions by incorporating innovations like a new backbone network, a new anchor-free split head, and new loss functions. These improvements enable YOLOv8 to deliver superior results while maintaining a compact size and exceptional speed. Additionally, YOLOv8 supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage YOLOv8's capabilities across diverse applications and domains.

3.3 Digital forensics analysis methods and process

In this study, in the first phase, data set collection, labeling, model optimization, and model training were performed with the YOLOv8 architecture. As a result, a face detection pytoch model was created.

In the second stage, the image file containing the face is given as input to the model file, and the region where the face is detected is found as output and, bounding box information is obtained.

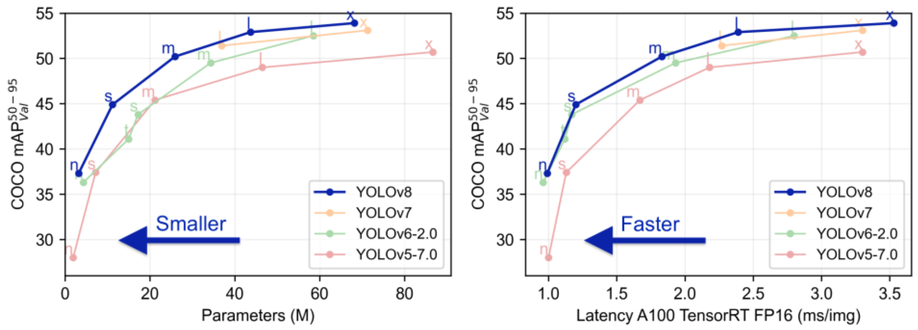


Figure 3. YOLOv5-v6-v7-v8 models comparison [42]

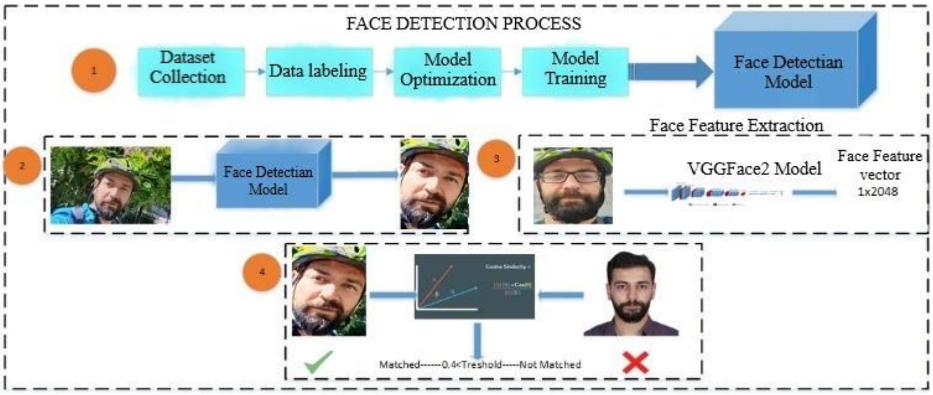


Figure 4. Face detection and recognition process

In the third stage, a 1*2048 2D feature map is extracted from the pre-trained VGGFace2 model to compare the face image entered by the user with the searched image.

The extracted feature map is compared with the feature map of the suspects in the database using the cosine similarity method, and the distance vector is calculated.

If the estimated distance vector is below the threshold value for similarity, it is determined that the person is matched, otherwise not matched considering that there is little or no similarity. The detections are recorded in a database in Excel format so that the digital forensics expert can later search for which person(s) were detected in which images or videos. The complete process and steps are given in Figure 4.

3.3.1 Face detection model training process

The training process of the face detection model consists of data collection and cleaning, data labelling, model optimization, and training stages.

In this study, the Wider-Face dataset, which is widely used in face recognition tasks, is used. The WIDER FACE dataset is a face detection benchmark dataset, of which images are selected from the publicly available WIDER dataset. It has 32,203 images and labels 393,703 faces with a high degree of variability in scale, pose, and occlusion as depicted in the sample images. The WIDER FACE dataset is organized based on 61 event classes. For each event class, the dataset randomly selects 40%, 10%, and 50% of the data as training, validation, and testing sets. In order to achieve better results, the entire dataset was used for training and testing.

Specifically, in this study, 32,203 images from the Wider-Face dataset were separated into 80% training data (25,862) and 20% test data (6440) to achieve high performance. The sample images from Wider-Face dataset are presented in the Figure 5.

When object recognition algorithms perform training with multiple images, it is expected that the image data and the data belonging to the object in the same image data should be in a certain format. As a result of labeling operations in the literature, both the information belonging to the image data and the information belonging to the object in the image can be stored in data-containing structures such as CSV, JSON, XML, and TXT. Data labeling for object recognition tasks involves classifying the location of the object in a rectangular area. Thus, the object is located in the image by drawing a rectangle with two points (left-top and right-bottom) in the coordinate system, and according to the location of the object, the ratio of the center point to the whole image and the distance to the width and height, as shown in Figure 6, are saved in a ".txt" file with the same name as the image with class information.

Training is performed with Yolov8 architecture on the Wider-Face dataset with n (nano), s (small), m (medium), l (large), and x (extra-large) pre-trained networks. The naming of the pre-trained model files is proportional to the depth and width of the mesh.

Figure 5. Sample face images from the Wider-Face dataset [31]

3.3.2 Face detection and alignment

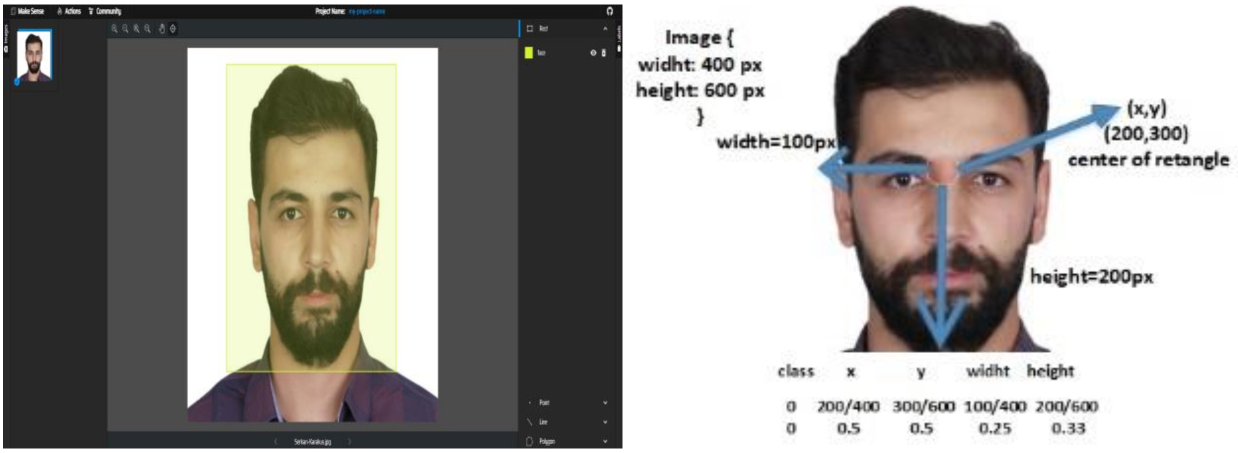
The face detection model file gives an image file as input and the coordinates of the rectangles outlining the boundaries of the detected face as output. However, if the face images detected by the model file are not aligned (tilted to the right or tilted to the left), the subsequent face recognition process cannot achieve high success. The detected face images need to be aligned before the next face recognition stage. Figure 7 shows the alignment process, and Eq. (1) shows mathematically how much to rotate.

After face detection, the face image should be rotated by the angle between the eye midpoints.

$$\cos(\alpha) = (b^2 + c^2 - a^2)/(2bc) \qquad (1)$$

3.3.3 Face feature extraction

Once the face has been identified and aligned, a feature map of each face needs to be created in order to compare it with the suspects in the database. The pre-trained VGGFace deep neural network using Se-Net architecture via deep neural network can extract a feature map of the face in 1x2048 size as output from the images given as input to the model file. This feature map will be used for matching input faces and suspect faces. The feature extraction process with the Pre-Trained VGGFace2 model is shown in Figure 8.



class	x	y	widht	height
0	200/400	300/600	100/400	200/600
0	0.5	0.5	0.25	0.33

Figure 6. Labelling and Yolo image label format

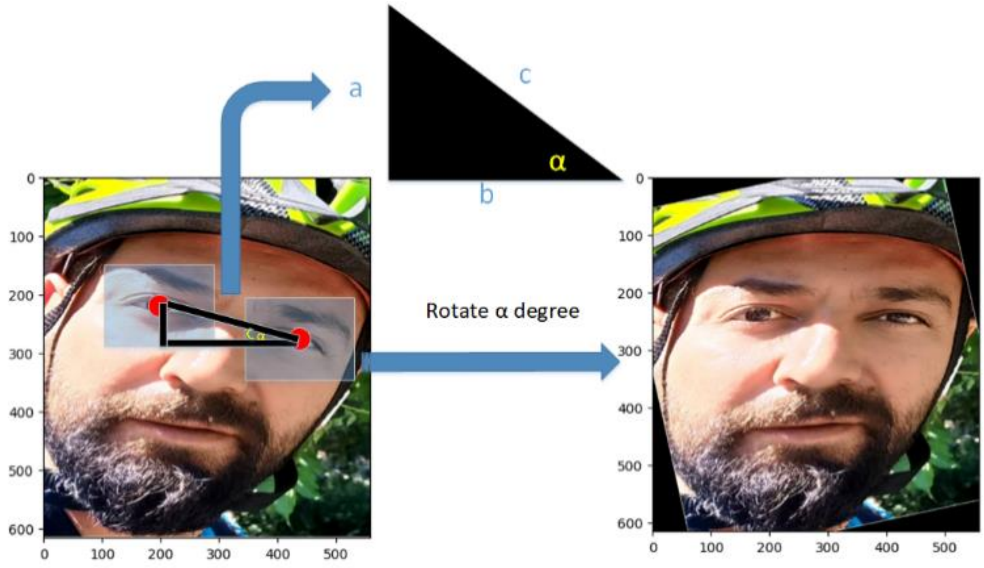


Figure 7. Face rotates and alignment

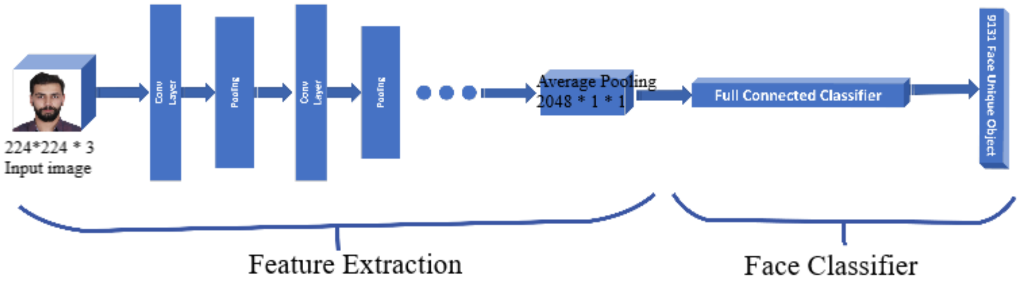


Figure 8. VGGFace2 face feature extraction process

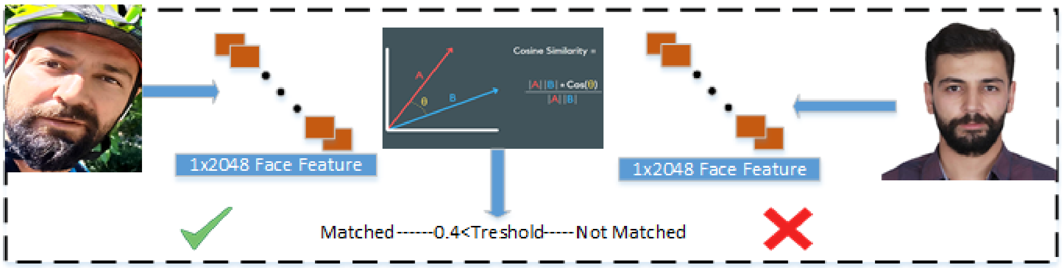


Figure 9. Face identification process

The VGGFace2 model is trained on Resnet architecture and can classify 9131 unique faces with high performance. The model architecture consists of two layers: feature extraction and face classifier. The feature extraction layer passes the input face image through the convolution and pooling layers and extracts a 1*2048 feature map in the average pooling layer. The face classifier layer classifies these extracted features through fully connected multilayer perceptrons. In this study, the pre-trained VGGFace2 model is excluded from the face classifier layer, and the features of the face images are extracted with the feature extraction layer.

3.3.4 Face identification

The digital forensics analysis application performs face detection on the frames of the image and video files given as input. The detected face is subjected to the alignment process, and the resulting image should be compared with the VGGFace pre-trained model in 1×2048 feature map vector space.

The facial features to be compared will produce a value between 0 and 1 by calculating the distance according to cosine similarity. The cosine similarly threshold value for the faces to be matched was determined to be 0.4 as a result of the tests. Figure 9 shows the face identification process, and Eq. (2) shows the mathematical expression of the cosine similarity value.

$$\cos(\theta) = \frac{A.B}{||A||\,||B||} = \frac{\sum_i^n A_i B_i}{\sqrt{\sum_i^n A_i^2}\,\sqrt{\sum_i^n B_i^2}} \qquad (2)$$

3.4 Digital forensics analysis tool

In the field of digital forensics, thousands of images and video files are obtained in each forensic case. The method required for analyzing these image and video files has been explained in detail. However, the expert digital forensic personnel who examine these files must be able to use them through a software interface. When we look at the literature, there is no software for face detection and recognition in the field of digital forensics. Existing software can extract images and videos from forensic evidence but cannot perform an analysis. The theoretical studies need to be put into use in the field and applied by expert personnel. With the method we have presented, forensic informatics experts can add photos of suspicious people through an interface. The suspect in the added image is trying to be detected in the images and videos on all the forensic evidence. It can analyze thousands of images and video files at high speed and accuracy. In the next version of the software, it will be developed to detect suspects in real-time streaming images.

4. EXPERIMENTAL RESULT

In this study, training was performed on the YOLOv8 framework with nano, small, medium, large, and extra-large pre-trained model files with the Wider-Face dataset. The inference values of the pre-trained YOLOv8 models obtained by Ultralytics on the Nvidia A100 GPU card are given in Table 1. The performance results of the pre-trained Yolov8 models (Nano-N, Small-S, Medium-M, Large-L, Xlarge-X) are compared on different GPU cards with precision, recall, mAP, inference time and fps parameters in the Table 2.

4.1 Pre-trained Yolov8 nano

The YOLOv8 pre-trained model can process real-time images on the Nvidia A100 GPU with 0.99 ms (1010 fps) inference time. As a result of the training on the Wider-Face dataset, 95.599% presicion, 91.245% recall, and 97.513% mAP (meanAvaragePresicion) values were obtained. The results are shown in Figure 10.

The model file obtained can process 25 ms (40 fps) image with 2 ms of preprocessing, 20 ms of inference, 3 ms of postprocessing on the hardware (Nvidia Super 2070).

4.2 Pre-trained Yolov8 small

YOLOv8 pre-trained model can process real-time images on the Nvidia A100 GPU with 1.20 ms (833,33 fps) inference time. As a result of the training on the Wider-Face dataset, 95.852% precision, 93.864% recall, and 98.569% mAP values were obtained. The results are shown in Figure 11.

The resulting model file can process the image for 27 ms (37.03 fps) with 2 ms of preprocessing, 22 ms of inference, and 3 ms of postprocessing on the training hardware (Nvidia Super 2070).

Table 1. YOLOv8 pre-trained model inference result [42]

Model	Size	mAP$^{val50-95}$	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Table 2. Yolov8 model performans different GPU card comparison

Model	Precision	Recall	mAP	Inference (A100/Super 2070)	Fps (A100/Super 2070)
N	95.59	91.24	97.51	0.99/25	1010/40
S	95.85	93.86	98.56	1.20/27	833/37
M	96.098	94.623	98.763	1.83/69.9	546/14
L	96.037	94.923	98.775	2.39/112.9	418/8.85
X	96.546	95.432	99.032	3.53/139.1	283/7.18

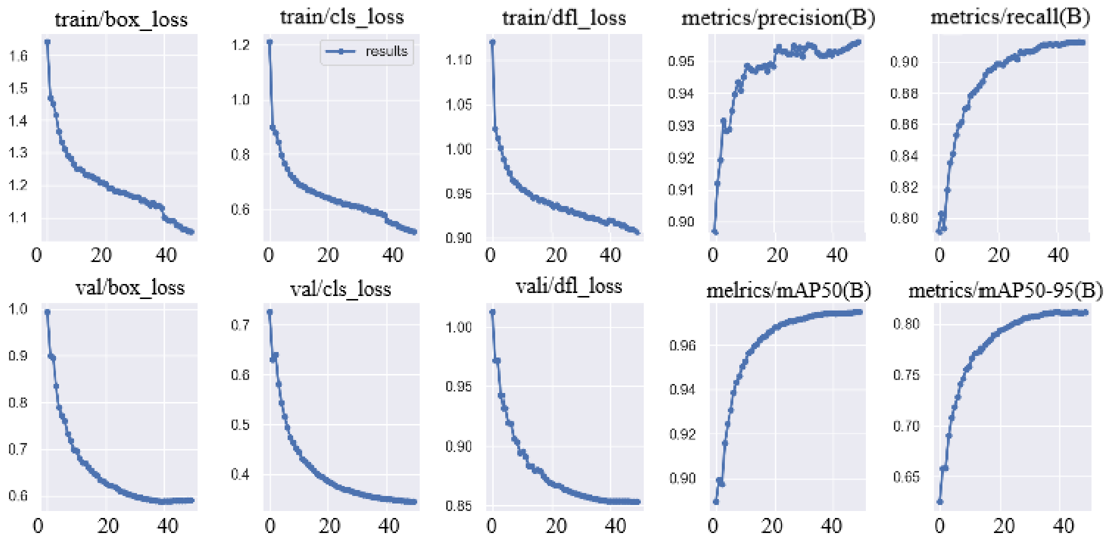


Figure 10. YOLOv8 small model training result

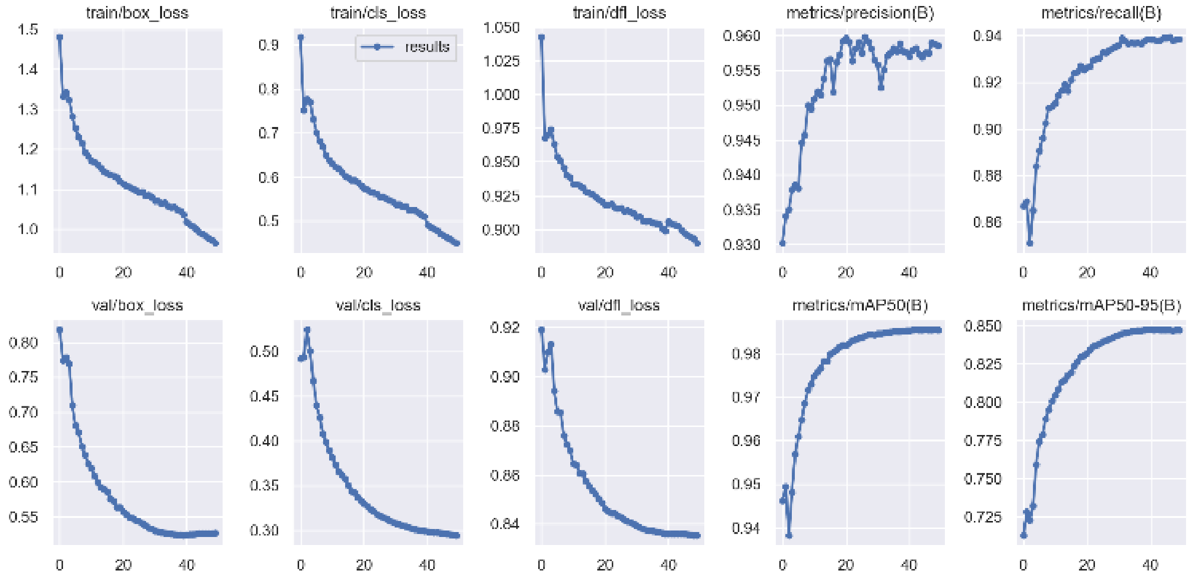


Figure 11. YOLOv8 small model training result

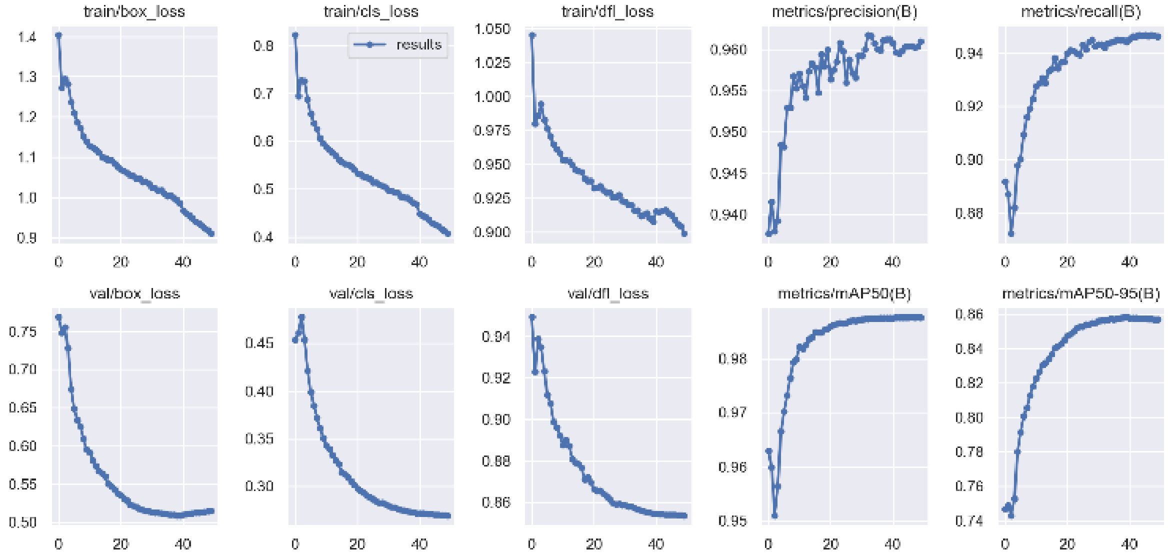


Figure 12. YOLOv8 medium model training result

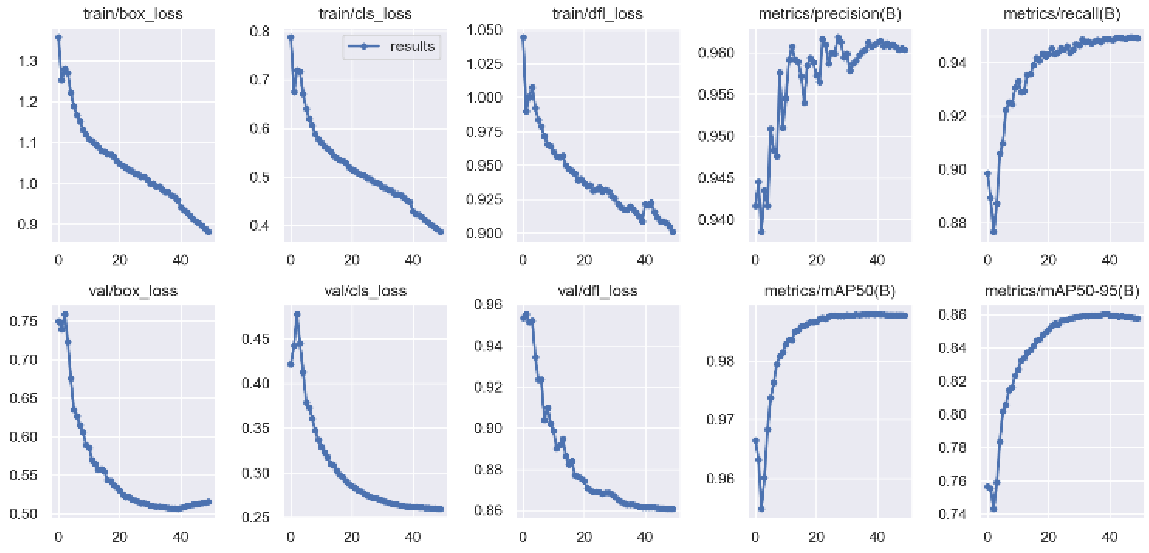


Figure 13. YOLOv8 large model training result

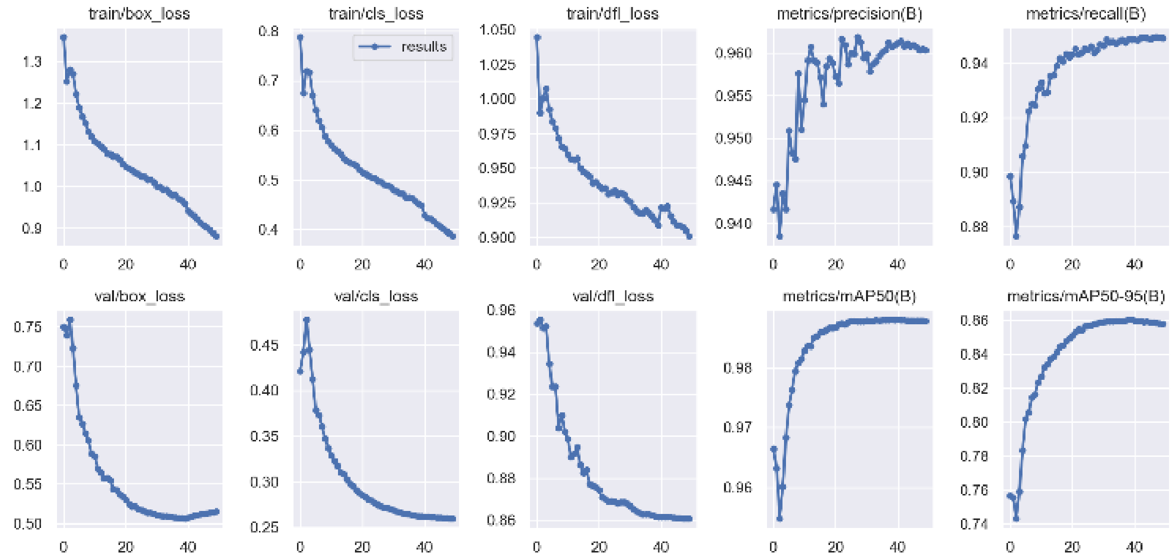


Figure 14. YOLOv8 large model training result

4.3 Pre-trained Yolov8 medium

YOLOv8's pre-trained model can process real-time images on Nvidia A100 GPU with 1.83 ms (546,44 fps) inference time. According to the results obtained as a result of model training, 96.098% precision, 94.623% recall, and 98.763% mAP were achieved. The results are shown in Figure 12.

The resulting model file can process the image for 69.9 ms (14.30 fps) with 2 ms of preprocessing, 59.9 ms of inference, 8 ms of postprocessing on the hardware (Nvidia Super 2070).

4.4 Pre-trained Yolov8 large

YOLOv8's pre-trained model can process real-time images on the Nvidia A100 GPU with 2,39 ms (418,41 fps) inference time. According to the results obtained as a result of the model training, it achieved 96.037% precision, 94.923% recall, and 98.775% mAP. The results are shown in Figure 13.

The resulting model file can process the image for 112.9 ms (8.85 fps) with 2 ms of preprocessing, 103.9 ms of inference, and 7 ms of postprocessing on the hardware (Nvidia Super 2070).

4.5 Pre-trained Yolov8 extralarge

YOLOv8's pre-trained model can process real-time images on the Nvidia A100 GPU with 3,53 ms (283,28 fps) of inference time. According to the results obtained as a result of the model training, it achieved 96.546% precision, 95.432% recall, and 99.032% mAP. The results are shown in Figure 14.

The resulting model file can process the image for 139.1 ms (7.18 fps) with 2 ms of preprocessing, 130.1 ms of inference, and 7 ms of postprocessing on the hardware (Nvidia Super 2070).

4.6 Yolov8 model performance discussion

Yolov8 object recognition models can process real-time images, depending on the hardware used. Especially with hardware such as the Nvidia A100 in the Google Colab environment, low inference and high fps values can be achieved. Especially the nano model can process real-time images at 40 fps on an average personal computer. However, considering that the Xlarge model achieved over 99.03% MAP, a high-performance face detection model was obtained with a computer with sufficient hardware specifications. If it is considered that the face detection model will be deployed on a development board such as a Nvidia Jatson, it is considered that it would be appropriate to use the nano model. In future studies, if high-performance detection and recognition in real-time images is desired, it is thought that the use of Large and Xlarge models will increase the performance in such scenarios.

5. DIGITAL FORENSIC ANALYSIS TOOL

In this study, a Digital Forensic Evidence Analysis Software (Face Detection and Recognition) was created to be used by digital forensic experts. The software can perform high-performance and real-time analysis by taking images and videos obtained from digital evidence and the image of the suspect. The interface of the analysis software is given in Figure 15.

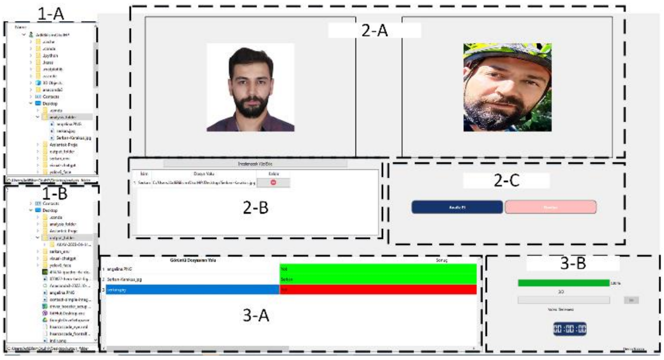


Figure 15. Digital forensics image/video analysis software

In Figure 15, in field 1-A, it is necessary to select the folder containing the videos and images to be analyzed.

The source and destination path can be defined respectively in section 1-A and 1-B.

The image of suspect analyzed, and the result image founded can be seen in section 2- A on both sides.

Forensic informatics experts can analyze trough the app interface multiple suspects from the image and video files in the folder choosen from Section 1-A, and display their path in real time in Section 3-A. In Section 1-B, they will be able to save the report they will obtain as a result of the analysis and control it later. Considering the contribution of the software to the field of forensic informatics, it is seen that it will achieve high success in terms of time and accuracy in terms of the analysis that digital forensics experts can do. In future studies, not only face detection but also crime object detection in images and videos can be easily done by changing the model file used.

6. CONCLUSIONS

In this study, model files trained on Yolov8 architecture with Wider-Face dataset on image and video files obtained from digital evidence were created. YOLOv8 model files (n, s, m, l, x) can successfully detect faces in images and videos at 97.513%, 98.569%, 98.763%, 98.775%, and 99.032%. The detected face images can be feature extracted with pre-trained feature extraction models (VGGFace2), and these feature maps can be compared with the cosine similralty algorithm.

It is seen that theoretical applications related to face recognition have been made in previous studies. However, there is no software package that can detect and analyze objects recognition models in the field of digital forensics. Although it is seen that some companies producing forensic analysis software have studies in this field, there is no product used by digital forensic experts. In future studies, it will be seen that it is possible to obtain higher mAP values and lower inference times in a training with the optimized Yolov8 architecture on larger data sets. It is planned to be able to perform face detection and recognition not only from images and video files existing in a folder but also from the camera in real time. Our software, which will be integrated into widely used forensic evidence analysis software, can automatically analyze the input folders received from forensics.

Image analysis software. Not only can it be integrated into other software, but with an additional module to be added to our software, a copy of forensic evidence can be taken, and this copy can be imported into the software so that the file system can be read.

REFERENCES

[1] Axenopoulos, A., Eiselein, V., Penta, A., Koblents, E., La Mattina, E., Daras, P. (2019). A framework for large-scale analysis of video\" in the Wild\" to assist digital forensic examination. IEEE Security & Privacy, 17(1): 23-33. https://doi.org/10.1109/MSEC.2018.2875851

[2] Qu, S. (2019). An approach based on object detection for image forensics. In 2019 1st International Conference on Industrial Artificial Intelligence (IAI), pp. 1-6. https://doi.org/10.1109/ICIAI.2019.8850791

[3] Ferreira, W.D., Ferreira, C.B., da Cruz Júnior, G., Soares, F. (2020). A review of digital image forensics. Computers & Electrical Engineering, 85: 106685. https://doi.org/10.1016/j.compeleceng.2020.106685

[4] Cellebrite. (2020). The state of digital evidence 2020. Erişim Tarihi: 4 Nisan 2023, https://www.cellebrite.com/en/resources/white-papers/the-state-of-digital-evidence-2020/.

[5] Riadi, I., Herman, Siregar, N.H. (2022). Mobile forensic analysis of signal messenger application on android using Digital Forensic Research Workshop (DFRWS) framework. Ingénierie des Systèmes d'Information, 27(6): 903-913. https://doi.org/10.18280/isi.270606

[6] Korkmaz, Y., Boyacı, A. (2023). Hybrid voice activity detection system based on LSTM and auditory speech features. Biomedical Signal Processing and Control, 80: 104408. https://doi.org/10.1016/j.bspc.2022.104408

[7] Abebaw, Z., Rauber, A., Atnafu, S. (2022). Design and implementation of a multichannel convolutional neural network for hate speech detection in social networks. Revue d'Intelligence Artificielle, 36(2): 175-183. https://doi.org/10.18280/ria.360201

[8] Channabasava, U., Raghavendra, B.K. (2022). Ensemble assisted multi-feature learnt social media link prediction model using machine learning techniques. Revue d'Intelligence Artificielle, 36(3): 439-444. https://doi.org/10.18280/ria.360311

[9] Piva, A. (2013). An overview on image forensics. International Scholarly Research Notices, Article ID 496701. https://doi.org/10.1155/2013/496701

[10] Zeng, J., Tan, S., Li, B., Huang, J. (2017). Large-scale JPEG image steganalysis using hybrid deep-learning framework. IEEE Transactions on Information Forensics and Security, 13(5): 1200-1214. https://doi.org/10.1109/TIFS.2017.2779446

[11] Kandi, H., Mishra, D., Gorthi, S.R.S. (2017). Exploring the learning capabilities of convolutional neural networks for robust image watermarking. Computers & Security, 65: 247-268. https://doi.org/10.1016/j.cose.2016.11.016

[12] Yao, H., Qiao, T., Xu, M., Zheng, N. (2018). Robust multi-classifier for camera model identification based on convolution neural network. IEEE Access, 6: 24973-24982. https://doi.org/10.1109/ACCESS.2018.2832066

[13] Ouyang, J., Liu, Y., Liao, M. (2017). Copy-move forgery detection based on deep learning. In 2017 10th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI), pp. 1-5. https://doi.org/10.1109/CISP-BMEI.2017.8301940

[14] Javed, A.R., Jalil, Z. (2020). Byte-level object identification for forensic investigation of digital images. In 2020 International Conference on Cyber Warfare and Security (ICCWS), pp. 1-4. https://doi.org/10.1109/ICCWS48432.2020.9292387

[15] Zafeiriou, S., Zhang, C., Zhang, Z. (2015). A survey on face detection in the wild: Past, present and future. Computer Vision and Image Understanding, 138: 1-24, https://doi.org/10.1016/j.cviu.2015.03.015

[16] Viola, P., Jones, M.J. (2004). Robust real-time face detection. International Journal of Computer Vision, 57: 137-154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[17] Bledsoe, W.W. (1966). The model method in facial recognition. Panoramic Research Inc. Palo Alto CA Rep. PRl, 15(47): 2.

[18] Turk, M., Pentland, A. (1991). Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1): 71-86. https://doi.org/10.1162/jocn.1991.3.1.71

[19] Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J. (1997). Eigenfaces vs. fisherfaces: Recognition using class

specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7): 711-720. https://doi.org/10.1109/34.598228

[20] Moghaddam, B., Jebara, T., Pentland, A. (2000). Bayesian face recognition. Pattern Recognition, 33(11): 1771-1782. https://doi.org/10.1016/s0031-3203(99)00179-x

[21] Yang, M., Zhang, L., Yang, J., Zhang, D. (2010). Metaface learning for sparse representation based face recognition. In 2010 IEEE International Conference on Image Processing, pp. 1601-1604. https://doi.org/10.1109/ICIP.2010.5652363

[22] He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J. (2005). Face recognition using laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(3): 328-340. https://doi.org/10.1109/TPAMI.2005.55

[23] Guo, G., Li, S.Z., Chan, K. (2000). Face recognition by support vector machines. In Proceedings fourth IEEE International Conference on Automatic Face and Gesture Recognition (cat. no. PR00580), pp. 196-201. https://doi.org/10.1109/AFGR.2000.840634

[24] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25: 1097-1105. https://doi.org/10.1145/3065386

[25] Taigman, Y., Yang, M., Ranzato, M.A., Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701-1708. https://doi.org/10.1109/CVPR.2014.220

[26] Sun, Y., Wang, X., Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1891-1898. https://doi.org/10.1109/CVPR.2014.244

[27] Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815-823. https://doi.org/10.1109/CVPR.2015.7298682

[28] Parkhi, O.M., Vedaldi, A., Zisserman, A. (2015) Deep face recognition. Proceedings of the British Machine Vision Conference (BMVC). https://doi.org/10.5244/c.29.41

[29] Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A. (2018). VGGFace2: A dataset for recognising faces across pose and age. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). https://doi.org/10.1109/fg.2018.00020

[30] Deng, J., Guo, J., Xue, N., Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690-4699. https://doi.org/10.1109/CVPR.2019.00482

[31] Yang, S., Luo, P., Loy, C.C., Tang, X. (2016). Wider face: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5525-5533.

[32] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81

[33] Girshick, R. (2015). Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[34] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 91-99. https://doi.org/10.1109/ICCV.2015.169

[35] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[36] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263-7271. https://doi.org/10.1109/CVPR.2017.690

[37] Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental improvement. Computer Science, arXiv: 1804.02767. http://arxiv.org/abs/1804.02767

[38] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.7. https://doi.org/10.48550/arXiv.2004.10934

[39] Jocher, G., Chaurasia, A., Stoken, A., et al. (2022). Ultralytics/yolov5: v7.0 - YOLOv5 SOTA realtime instance segmentation. Zenodo. https://doi.org/10.5281/zenodo.7347926

[40] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976. https://doi.org/10.48550/arXiv.2209.02976

[41] Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696. https://doi.org/10.48550/arXiv.2207.02696

[42] Ultralytics. (2023). Ultralytics. GitHub. https://github.com/ultralytics/ultralytics.