ISSN : 2210-142X

International Journal of Computing and Digital Systems (IJCDS)

Volume No. 13
Issue No. 3
September - December 2024

ENRICHED PUBLICATIONS PVT. LTD

JE-18, Gupta Colony, Khirki, Extn, Malviya
Nagar, New Delhi-110017
PHONE : - + 91-8877340707
E-Mail : info@enrichedpublications.com

ISSN : 2210-142X

International Journal of Computing and Digital Systems (IJCDS)

Aims and Scope

International Journal of Computing and Digital Systems (IJCDS) is a peer-reviewed International Journal that currently publishes 6 issues annually. IJCDS journal publishes technical papers, as well as review articles and surveys, describing recent research and development work that covers all areas of computer science, information systems, and computer / electrical engineering.

The topics covered by IJCDS are including and not limited to the following research areas:

- Reconfigurable Computing & Embedded systems
- Computer Communications and Networking
- Internet of Things & Real Time Systems
- Cyber Security
- Cloud Computing
- Smart Systems
- Information Systems and Communication Service
- Innovation/Technology Management
- Business Information Systems
- Software Engineering
- Mobile & Web Applications
- Theory of Computation
- Data Structures, Cryptology and Information Theory
- Artificial Intelligence & Robotics
- Image Processing, Computer Vision, Pattern Recognition & Graphics
- Data Mining & Big Data
- Smart Grids & Renewable Energy
- Human Computer Interaction

Responsiveness Speed

Time from submission to first decision after peer review: 4-8 Weeks
Time to immediate reject: 2-4 Weeks

Publication Speed

Time from final acceptance to print: 3-6 Months
Total time to publication: 4-8 Months
Time from submission to acceptance: 4-12 Weeks
Time from acceptance to online Publishing: 4-8 Weeks

Publishing Ethics

Plagiarism in all its forms constitutes unethical publishing behavior and is unacceptable. The journal has no tolerance on plagiarism. All submitted manuscripts must go through cross checking using turnitin as an online plagiarism checker.

ISSN : 2210-142X

International Journal of Computing and Digital Systems (IJCDS)

(Volume No. 13, Issue No. 3, September - December 2024)

Contents

ISSN 2210-142X

An Approach for Aircraft Detection using VGG19 and OCSVM

Marwa A. Hameed1 and Zainab A. Khalaf 2*
1Computer Sciences, College of Computer Science & Information Technology, Basrah University, Basrah, Iraq
2Department of Computer Science, College of Education of Pure Sciences, Basrah University, Basrah, Iraq

ABSTRACT

Aircraft detection is an essential and noteworthy area of object detection that has received significant interest from scholars,especially with the progress of deep learning techniques. Aircraft detection is now extensively employed in various civil and militarydomains. Automated aircraft detection systems play a crucial role in preventing crashes, controlling airspace, and improving aviationtraffic and safety on a civil scale. In the context of military operations, detection systems play a crucial role in quickly locating aircraftfor surveillance purposes, enabling decisive military strategies in real time. This article proposes a system that accurately detectsairplanes independent of their type, model, size, and color variations. However, the diversity of aircraft images, including variationsin size, illumination, resolution, and other visual factors, poses challenges to detection performance. As a result, an aircraft detectionsystem must be designed to distinguish airplanes clearly without affecting the aircraft's position, rotation, or visibility. The methodologyinvolves three significant steps: feature extraction, detection, and evaluation. Firstly, deep features will be extracted using a pre-trainedVGG19 model and transfer learning principle. Subsequently, the extracted feature vectors are employed in One Class Support VectorMachine (OCSVM) for detection purposes. Finally, the results are assessed using evaluation criteria to ensure the effectiveness andaccuracy of the proposed system. The experimental evaluations were conducted across three distinct datasets: Caltech-101, Militarydataset, and MTARSI dataset. Furthermore, the study compares its experimental results with those of comparable publications releasedin the past three years. The findings illustrate the efficacy of the proposed approach, achieving F1-scores of 96% on the Caltech-101dataset and 99% on both Military and MTARSI datasets.

Keywords: Aircraft Detection, Deep Learning, Object Detection, VGG19, OCSVM.

1. Introduction

At the core of computer vision lies the principle of object detection, a process that involves the identification and categorization of objects by employing rectangular bounding frames. This method serves the dual purpose of localizing and classifying the identified objects, making it an integral part of the broader field of computer vision. Object detection is closely intertwined with related tasks, including object classification, semantic segmentation, and instance segmentation, collectively contributing to the un derstanding of visual data [1], [2]. The significance of object detection extends across a diverse spectrum of practical applications, encompassing domains such as autonomous driving, robotics, and video surveillance. It plays a pivotal role in enabling machines to perceive and interact with their surroundings effectively [3].

Recently, there was an increasing scholarly focus on the utilization of object detection techniques to the detection of aircraft within the domain of computer vision. This application serves critical functions in contexts like airport control, search operations for crashed aircraft, monitoring the dynamics of potentially hostile aircraft, and finds relevance in both civil and military sectors [3], [4].

Deep learning, as an advanced methodology within computer vision, has emerged as a powerful tool for extracting precise image characteristics and analyzing data. This specific category of machine learning techniques, referred to as deep learning, introduces complexity into models, ultimately enhancing their capabilities. Deep learning has demonstrated the potential to significantly improve accuracy across various domains, encompassing classification, segmentation, as well as object detection [5].

In recent times, deep learning played a pivotal role in autonomously extracting feature representations from data, leading to significant advancements in object detection [6].

A notable contribution by Liu et al. involved the development of an aircraft detection method utilizing corner clustering and Convolutional Neural Networks (CNN). The methodology comprised two primary phases: region detection and classification. It initiated with the identification of potential regions through the application of mean-shift clustering on corners observed in binary images. Subsequently, CNNs were employed for feature extraction and classification of regions likely to contain aircraft. The aircraft's precise location was ultimately ascertained through subsequent refinement and evaluation. The study utilized optical remote sensing images from the Remote Sensing Object Detection (RSOD) dataset that yielded a classification accuracy of 98.29% [7].

In another comparative evaluation, Alganci et al. assessed the efficacy of three object detection models: Faster Region-based Convolutional Neural Networks (R-CNN), YOLO-v3, and Single Shot MultiBox Detector (SSD) specifically for aircraft detection within very high-resolution (VHR) satellite imagery. The evaluation aimed to handle the challenge of limited labeled data by leveraging the Dataset for Object deTection in Aerial Images (DOTA), which comprised satellite image patches from various sources. Among the models, YOLO achieved a precision rate of 99.6%, while SSD exhibited a precision rate of 87.9%, and Faster R-CNN attained 81.2% [8].

Shi et al. introduced a two-stage airplane detection approach called DPANet (Deconvolution operation with Position Attention mechanism). This method incorporated deep neural networks with deconvolution operations and a Position Attention mechanism to enhance aircraft recognition in aerial imagery. Evaluations were performed on the DOTA and DIstill Observations to Representations (DIOR) datasets, resulting in a notable average precision (AP) increase to 85.95% [4].

Furthermore, Wang et al. proposed the Efficient Weighted Feature Fusion and Attention Network (EWFAN) for aircraft detection. This unique deep neural network integrated a module for weighted feature fusion and spatial attention mechanisms. The experiment utilized large-scale Gaofen-3 SAR images with a resolution of 1 m to assess the efficiency of the proposed architecture, ultimately achieving a detection rate of 95.4% and a false alarm rate of 3.3% [9].

Hu et al. introduced a cascade framework inspired by the YOLOv5 architecture for aircraft detection in remote sensing images. Their dataset, sourced from various satellite platforms, included 17,506 instances of aircraft belonging to 13 different types. The proposed approach achieved a mean average precision of 83.7%. However, the accurate detection of small objects against low-resolution and complex backgrounds remains a challenge. Additionally, the accuracy of direction prediction and recall rate is deemed inadequate, indicating the need for further improvements [10].

Xiao et al. introduced an adjustable deformable network (ADN) incorporating peak feature fusion (PFF) to recognize airplanes in SAR pictures. The primary objective of the PFF approach is to maximize the utilization of the solid scattering properties of aircraft. This involves extracting the most prominent features and combining them. The Harris detector and eight-domain pixel detection of local maxima extract peak features. Subsequently, multichannel blending is utilized to improve the visibility of airplanes in different imaging situations. The ADN comprises an adaptive spatial feature fusion (ASFF) module and a deformable convolution module (DCM). The ASFF module addresses scale incon sistencies, enhancing the feature pyramid's ability to characterize features across scales and improving multi-scale aircraft identification performance. The DCM dynamically calculates and determines the 2-D differences in feature maps, enhancing the representation of aircraft with different shapes in geometric modeling. The experimental findings on the GaoFen-3 (GF3) dataset indicate that PFF-ADN is highly effective, as evidenced by its F1-score of 91.11% and average precision of 89.34% [11].

A novel framework was presented by Chen et al. for detecting aircraft within Remote Sensing Images (RSIs). They employed a region suggestion technique relying on circular intensity filtering to find probable targets at different scales. In addition, they utilized the Vector of Locally Aggregated Descriptors (VLAD) method to characterize the rotation-invariant Fourier Histogram of Oriented Gradients (HOG) feature. This approach provides a more condensed representation and improved descriptive capability while disregarding the rotation behavior of the target. The optical Remote Sensing Indices (RSIs) utilized in this research were obtained via the RSOD dataset, that consists of 446 remote sensing images, including a total of 4993 aircraft objects. The images used in this study were obtained from Google Earth and Tianditu. They had different levels of detail, with spatial resolutions ranging from 0.5 to 2 meters. The image from Google Earth was 1072 x 975 pixels, while the one from Tianditu had a size of 1116 x 659 pixels. The results demonstrated that the proposed technique could achieve faster and more accurate detection of aircraft targets in RSIs, thereby enhancing overall performance, with an average precision of 93.4%. Analysis of the failure results revealed that in intricate background scenarios, including the concourse of an airport building, might potentially result in misdirected targets. Conversely, the similarity in color between the airplane fuselage and the ground, caused by irregular lighting and aircraft paint, could result in potential missed detections [12].

Liu et al. introduced the YOLO-extract algorithm that improves the model architecture of the YOLOv5 method. The strategy eliminates the feature layer and prediction head. which have limited feature extraction capabilities, replacing them with a new feature extractor with higher abilities within the network. Furthermore, YOLO-extract leverages the concept of residual networks to integrate Coordinate Attention into the network seamlessly. The algorithm further enhances the capacity of the shallow layer to extract feature and position information by pairing mixed dilated convolution with the redesigned residual structure, optimizing the model's feature extraction ability for targets of various scales. The experiment results on the DOTA dataset indicate that YOLO-extract exhibits quicker conver gence and reduces computational workload by 45.3GFLOPs and parameter count by 10.526M, compared to the YOLOv5 algorithm. Additionally, YOLO-extract enhances the mean average precision (mAP) by 8.1% and triples the detection speed. However, the extraction of various aircraft target fea tures from remote sensing satellite images faces challenges due to weather conditions, including skylight circumstances, clouds, and fog. Additionally, the scarcity of datasets related to aircraft types poses challenges for aircraft type detection [13].

Zhang et al. introduced an innovative methodology for identifying aircraft in Synthetic Aperture Radar (SAR) images with a low signal-to-clutter-noise ratio (SCNR). This approach utilized coherent scattering enhancement and a fusion attention mechanism. They also improved the Faster R-CNN by implementing a novel pyramid network that incorporates mechanisms for local and contextual attention. The contextual attention mechanism enables the network to extract pertinent contextual information from the image, whereas the local attention mechanism adaptively emphasizes significant features by enhancing their unique attributes. By effectively integrating local and contextual attention, the network can detect aircraft. Considerable experimentation was undertaken utilizing the TerraSAR-X SAR datasets to establish benchmarks. The experimental findings indicate that under conditions of low SCNR, the suggested aircraft detection method attains an average precision of 91.7% [14].

This study presents a method specifically developed to accurately detect airplanes, regardless of model, type, or color modifications. Detection of airplanes in automated activities is a significant difficulty due to their substantial variations in scale, direction, and visual resemblance to other objects. To address these challenges, an airplane detection system must be engineered to achieve robust discrimination, independent of factors such as rotation, pose, or resolution of the airplanes.

The paper is structured into four primary sections. Section 2 provides an in-depth discussion of the key al gorithms and techniques utilized in this study. Section 3 presents the proposed system. The ensuing section, Section 4, delves into a comprehensive presentation of the experimental results. This is followed by a comparative analysis and discussion with recent related works in Section 5. Finally, Section 6 offers a concise summary of the paper's conclusion.

2. Methods

This section primarily concentrates on the presentation of the VGG19 model, an exploration of the one-class Support Vector Machines (OCSVM) algorithm and transfer learning concept.

A. VGG19 model

The VGG19 model, introduced by Simonyan and Zisserman in 2014, is a convolutional neural network consisting of 19 layers. VGG19 comprises five architectural blocks. The convolutional and pooling layers are present in the first and following blocks. Two convolutional layers and one pooling layer are present in the third and fourth blocks, respectively. Four convolutional layers are in the final block. 3 × 3 modest filters are employed [15]. It includes 16 convolution layers and three fully connected layer. VGG19 is trained using the ImageNet dataset, which consists of more than one million images categorized into 1000 distinct categories [16].

The VGG19 architecture follows a sequential pattern of convolutional layers, interspersed with max-pooling layers, to reduce spatial dimensions and increase the depth of learned features. The convolutional layers employ rectified linear unit (ReLU) activations, promoting faster convergence and better training efficiency. The final layers consist of fully connected layers with softmax activation, enabling classification into multiple classes [17] , as shown in Figure 1.

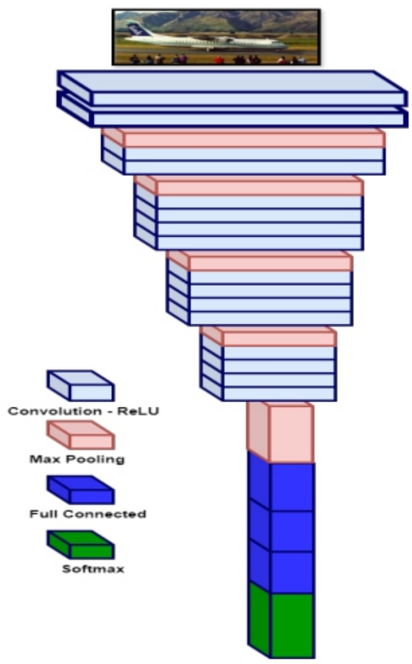


Figure 1. The Architecture of VGG19

VGG19 has been a pivotal model in the field of deep learning, showcasing the significance of depth in neural network architectures. It has become a popular choice for numerous computer vision applications, including object detection and localization, due to its ability to acquire complex features [18], [19], [20]. In this study, the VGG19 model is employed to extract features from provided images of airplanes.

B. One-Class Support Vector Machine (OCSVM)

The OCSVM is a machine learning algorithm primarily used for anomaly detection or novelty detection tasks [21]. Unlike traditional Support Vector Machines (SVM), which are designed for binary classification, One-Class SVM is focused on learning and identifying a single class or a specific region in the feature space that represents ”normal” or typical instances [22].

In essence, One-Class SVM constructs a decision boundary or a hyperplane that encapsulates the majority of the training data, aiming to enclose the ”normal” data points while minimizing the outliers or anomalies. This objective is accomplished by identifying the hyperplane that optimizes the margin among the point of origin as well as the nearest points of data belonging to the target class, successfully segregating it from the remaining data points [22], [23].

During the testing or inference phase, the algorithm then identifies instances that fall outside the defined boundary as potential anomalies or deviations from the established norm [24].

Mathematically, the objective of One-Class SVM is to find a hyperplane represented by the equation [25]:

$$\mathbf{w}^T + \phi(\mathbf{x}) - b = 0 \qquad (1)$$

where:

The weight vector, denoted as w, is orthogonal to the hyperplane. Additionally, $\phi(x)$ represents a feature mapping for the input data. The variable ”b” represents the bias term [25].

The hyperplane effectively partitions the feature space by maximizing the margin, which refers to the distance from the hyper plane and the nearest data point, commonly referred to as the support vector [23].

To accommodate a certain level of error or deviation from the boundary, a slack variable ξ is introduced, allowing some data points to fall within the margin or on the wrong side of the hyperplane. The optimization problem for One-Class SVM can be formulated as [25]:

$$\min \frac{1}{2}\mathbf{w}^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho \qquad (2)$$

Subject to:

$$\mathbf{w}^T + \phi(\mathbf{x}_i) - b \geq \rho - \xi_i, \quad \xi_i \geq 0; \quad i = 1, \ldots, n \qquad (3)$$

In these equations, the hyper parameter v regulates the maximum proportion of margin errors, whereas ”ρ” represents the radius for the hyper sphere, which encompasses the transformed average data points [22].

C. Transfer Learning (TL)

Deep learning, also known as deep structured learning, represents an advanced category of machine learning meth ods that has significantly impacted the field of artificial intelligence. The term ”deep” refers to the network’s depth, indicating the presence of multiple layers, which enables it to learn intricate patterns and features from data. Convolutional Neural Networks (CNNs) are a prominent deep learning technique widely utilized for feature extraction and data classification across various domains. A typical CNN architecture comprises several layers, including input, convolutional, pooling, fully connected, and output layers [5].

Transfer learning (TL) is a valuable technique employed to address data limitations by leveraging knowledge from a different domain. Instead of training a neural network from scratch, TL involves utilizing a pre-trained model's weights and architecture, which have been optimized on a large dataset. This approach allows for the extraction of common features and structures present in images, thereby facilitating the recognition of distinctive features specific to a particular dataset [15].

In TL, the pre-trained model's lower layers, which capture general features, are typically preserved, while the upper layers, responsible for more task-specific features, are fine-tuned to adapt to the new dataset. This transfer of knowledge from the source task to the target task enables the model to achieve good performance with minimal data. However, it is important to note that the pre-trained model's last few layers, initially designed for the source task, may require re-training to better suit the target task's characteristics. [26].

Transfer learning is a technique that entails employing pre-trained deep networks, which have been trained on large datasets, to tackle particular tasks with restricted data. Using the target dataset, it is customary to fine-tune the concluding layers of the pre-trained network. This enables the model to transfer information from the source task to the targeting task. Nevertheless, it is important to note that the ultimate layers, which were initially developed on the source task, might not acceptable with the target goal [27].Implementations of transfer learning that utilize pre-trained models tend to be more efficient and reliable compared to models trained from scratch [28].

In this study, two transfer learning approaches are employed:

1) Feature Extraction Approach

The feature extraction technique involves utilizing VGG19 network for the purpose of feature extraction, retaining its original design and learned weights. This approach entails using the network up to a predefined layer as an arbitrary feature extractor, with the outputs of these layers serving as features for further processing [28]. The VGG19 model, known for its 16 convolutional layers and three output layers, leverages its convolutional layers for feature extraction, preserving the original features of input images in the form of feature maps [17].

2) Fine-Tuning Approach

In the fine-tuning strategy, the pre-trained VGG19 architecture is adapted by replacing the original fully connected layers with newly initialized ones. These new fully con nected layers are then trained to predict the input classes [29]. Additionally, the last three layers of the VGG19 model are replaced with OCSVM to enhance the model's performance in specific tasks.

3. The Proposed System

The fundamental concept presented in this paper in volves utilizing the VGG19, which is modeled as a feature extractor through the 16 convolutional layers. Subsequently,these extracted features are then employed for airplane detection using OCSVM.

The architecture for aircraft detection in the proposed system comprises five primary stages, depicted in 2. These stages encompass datasets, preprocessing, feature extraction, detection, and evaluation. Pre-processing plays a vital role in priming the aircraft images for subsequent processing. During the feature extraction phase, deep learn ing filtering is employed to extract and filter out irrelevant features. These features are then fed into the detection phase to accurately detect airplanes. In the last stage, the system's performance is assessed by evaluating the results obtained. Further elaboration on the system's specifics is provided in the subsequent subsections.

A. Dataset Stage

Three dataset used in this experiment. These are Caltech-101 dataset, Military Aircraft dataset, and MTARSI dataset that collected from the Kaggle website.

1) Caltech -101 Dataset

The Caltech -101 [26] is a widely used computer vision dataset designed for object recognition tasks. It was created by the California Institute of Technology (Caltech) and contains a diverse set of images belonging to 101 different object categories [10]. It comprises around 9146 images distributed among 101 object classes and an additional background clutter class. The number of images in each class ranging between 40 to 800 images, as in the aircraft category, which includes 800 images. The images have a uniform dimension of 300 x 200 pixels.

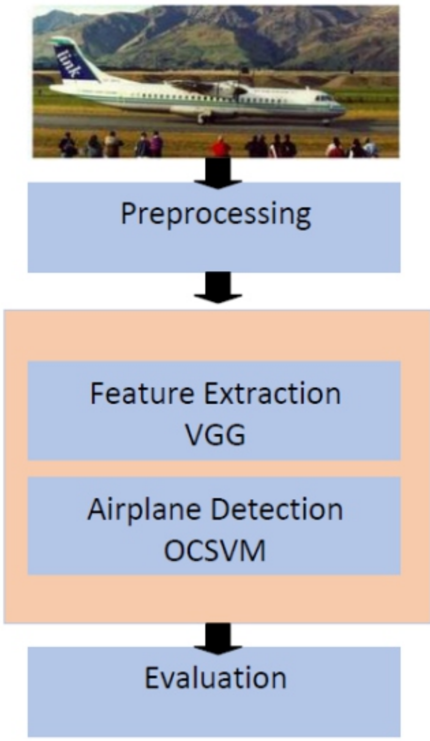


Figure 2. The Proposed System Flowchart

2) Military Aircraft Dataset

This is a remote sensing image Military Aircraft Recog nition dataset. It consists of a total of 3,842 images, encom passing 20 distinct types of military aircraft. Each image within the dataset has been meticulously annotated with both horizontal boundary boxes and orientated bounding boxes, resulting in a comprehensive collection of 22,341 instances.

3) MTARSI Dataset

MTARSI, short for Multi-type Aircraft of Remote Sens ing images [27] represents the initial publicly accessi ble dataset encompassing detailed aircraft classification designed for remote sensing photos. A group of seven esteemed professionals specializing in remote sensing image interpretation diligently annotated each instance of the provided images. Hence, the dataset exhibits significant credibility [28]. MTARSI comprises 9,385 remote sensing images derived from satellite imagery provided by Google Earth. The collection of aircraft images includes 36 various airports and 20 distinct type of aircrafts [29].

Additionally, one more dataset was generated based on the aforementioned Caltech-101 dataset. This generated dataset involves converting images into object boundaries using the Canny edge detection technique, a method known for its effectiveness in identifying edges with reduced noise and accurate edge localization in digital images. Figure 3 illustrates the main steps of the Canny algorithm.

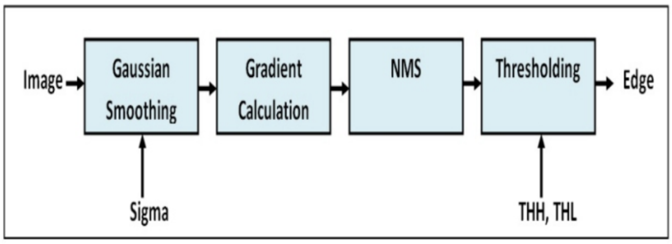


Figure 3. The Traditional Canny Method of Edge Detection

• **Gaussian Smoothing (G)** reduces noise in the image, which can be caused by factors such as sensor imperfections or compression artifacts. It also helps in suppressing small, insignificant edges that might be the result of noise. This is important because the Canny algorithm aims to detect strong, meaningful edges. Canny technique commonly employs a twodimensional Gaussian function as shown in equation (1) to smooth and remove noise from images [30].

$$G(x, y) = \frac{\exp\left[-\frac{(x^2+y^2)}{2\sigma^2}\right]}{2\pi\sigma^2} \tag{4}$$

The symbol "σ" represents the parameter of the Gaussian filter, which governs the smoothing degree of image.

• **Gradient Calculation** involves computing the gradient magnitude and direction for each pixel in the image. The gradient provides information about how the intensity of the image changes at each point and helps identify regions of rapid intensity change, which often correspond to edges. Edges can be detected in locations where the magnitudes of the picture gradients exhibit significant values. The magnitudes are obtained by convolving the image with the gradient masks [31]. The partial derivatives about the x and y axes can be represented as Px(i,j) and Py(i,j), respectively, with the image I(x,y). The conversion from rectangular coordinates to polar coordinates involves transforming the given coordinates Px(i, j) and Py(i, j) into the gradient amplitude M(i,j) and the gradient direction θ(i,j) for a pixel. M(i,j) denotes the edge strength of any point (i , j), while θ(i,j) denotes the normal vector of any point (i, j) [32].

$$M(i, j) = \sqrt{P_x(i, j)^2 + P_y(i, j)^2} \tag{5}$$

$$\theta(i, j) = \arctan\left(\frac{P_y(i, j)}{P_x(i, j)}\right) \tag{6}$$

• **Non-Maximal** Suppression plays a crucial role in the Canny edge detection method by aiding in the refinement of detected edges through thinning and retaining only the most prominent ones. This step ensures that the final edge map contains only thin,single-pixel-wide edges by suppressing non-maximal gradient values in the gradient magnitude image. To refine the edge features, it is necessary to suppress all values along the gradient line, with the exception of the local maxima [31]. The NMS method can in ensuring that each edge is one pixel wide. The Canny technique utilizes 3 x 3 neighboring regions, encompassing eight directions each, for interpolating the gradient magnitude based on the direction of the gradient. A potential edge point is identified when the magnitude M(i,j) exceeds the cumulative interpolation results along the gradient's direction. In contrast, the point is classified as non-edge if the magnitude is smaller. The method yields the candidate edge image as a result [30].

• **Double Thresholding** is a pivotal step in the Canny algorithm that aims to classify the edges into strong, weak, and non-edge pixels based on gradient magnitude values. This step involves applying two thresholds to the gradient magnitude image to distinguish between different levels of edge strength. Two thresholds are used: a high threshold (THH) and a low threshold (THL). A pixel is classified as a firm edge if its gradient exceeds a threshold value, denoted as THH. If the gradient value falls below the predetermined threshold level (THL), excluding or erasing the pixel corresponding to such a gradient value is imperative [30], [31]. Figure 4 shows an example of each dataset images.

B. Preprocessing Stage

Preprocessing plays a pivotal role in the preparation of images for subsequent processes [33]. It involves a series of transformations applied to an initial image, aimed at improving its quality and rendering statistical analysis more consistent and comparable [34]. In this study, various preprocessing techniques are employed to optimize the data, including resizing the images to a standard size of (224,224). Additionally, for the purpose of data augmentation, methods such as data cropping and rotation are applied.

C. Feature Extraction Stage

Feature extraction stands as a critical phase in airplane detection, as the model's efficacy significantly hinges on the quality and pertinence of the extracted features [35], [36].
This stage is dedicated to capturing pertinent visual characteristics from images that aid in distinguishing airplanes from other objects or backgrounds. As mentioned before the VGG19 model comprised of sixteen convolutional layers, is employed for this purpose.

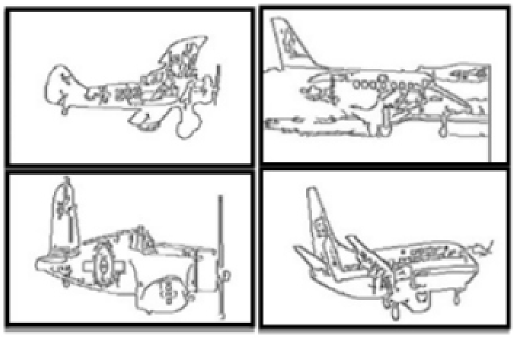
D. Detecion Stage

The detection phase commences right after the completion of the feature extraction stage. To achieve this, the final three layers of the VGG19 model configuration are replaced with OCSVM. OCSVM is a robust and wellregarded detector known for its effectiveness in object detection, particularly in the case of aircraft.

((a))

((b))

((c))



((d))

Figure 4. An Example of Used Datasets ((a)) CalTech-101 Dataset.((b)) Military Aircraft Dataset, ((c)) MTARSI Dataset, ((d)) CannyObject Boundaries Dataset

E. Evaluation Stage

The efficacy of the proposed method was assessed by utilizing various assessment measures in the experimental results. The criteria employed include F1-score and accuracy. The following are the most fundamentals [33], [37]:

True Positives (TP) refers to the rate at which the model accurately classifies airplanes as positive.
True Negatives (TN) refers to the rate at which the model accurately classifies airplanes as negative.
False Positives (FP) represents the number of airplanes erroneously identifies as positive when, in reality, they are actually negative.
False Negatives (FN) is the number of airplanes erroneously predicted as negative when, in reality,they are actually positive.The metrics can be defined as following:

$$P(\text{Precision}) = \frac{TP}{TP + FP} \tag{7}$$

$$R(\text{Recall}) = \frac{TP}{TP + FN} \tag{8}$$

$$F1\text{-score} = \frac{2 \times P \times R}{P + R} \tag{9}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

4. Experimental Results

The suggested system was executed on a personal computer according to the given specifications: Windows 10 Pro operating system, an Intel® Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz, installed RAM 8.00 GB, and system type 64-bit operating system. The proposed system was designed using Python programming language, version 3.10, within a Jupyter Notebook environment, chosen for its interactive and collaborative features.

For this study, the VGG19 and OCSVM methods were selected to harness the advantages of CNN and SVM techniques and demonstrate their efficacy in aircraft detection.

The training data undergoes feature extraction using VGG19 to determine class values. This involves putting all images through the CNN network to extract relevant features, which are then fed to OCSVM to perform the detection process.

Three datasets were chosen to showcase their outcomes, which were subsequently examined in depth to measure efficacy of the proposed methodology. Each data set was separated manually into two subsets (70:30) : the training set and the test set.

In the experimental evaluation, three detection models were applied: VGG19, OCSVM, and the proposed system.

To evaluate and compare the detection performance of the proposed system, we utilized three datasets: Caltech-101, the Military dataset, and MTARSI. In order to achieve optimal outcomes and guarantee accurate evaluation of the proposed system, It was configured with precise parameters, as outlined in Table I below.

TABLE I. Related Parameters of the Proposed System

Parameter	Value
kernal	linear
gamma	0.01
nu	0.01
cache size	100

Figure 5 illustrates the training and validation loss as well as the accuracy of the proposed system using the Caltech-101 dataset.

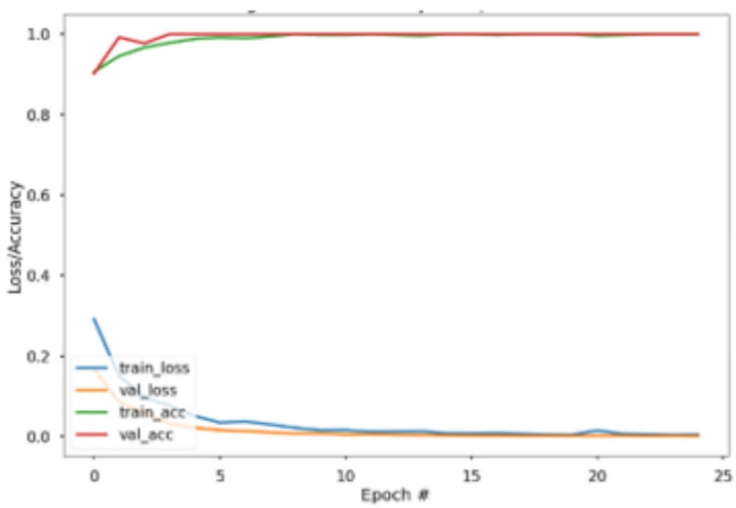


Figure 5. The Loss and Accuracy of Training Phase Utilizing Caltech-101

The effectiveness of the proposed system was assessed by combining VGG19 with OCSVM. The results are detailed in Table II.

TABLE II. Experimental Results of the Suggested System using F1-score

Datasets	Caltech-101	Military	MTARSI
OCSVM	94%	95%	96%
VGG19	88%	96%	97%
Proposed System	96%	99%	99%

Table II reveals that the proposed system outperformed both the VGG19 and OCSVM systems, achieving higher F1-score results for all datasets. For the Caltech-101 dataset, the F1-scores were 96% for the proposed system, 88% for both VGG19 and 94% for OCSVM. In the case of the military aircraft dataset, the obtained F1-scores were 99%, 96%, and 95% for the proposed system, VGG19, and OCSVM respectively. Meanwhile, for the MTARSI dataset, the proposed system exhibited an F1-score of 99% as opposed to 97% for VGG19 and 96% for OCSVM.

Figure 6 showcases an example of the proposed system's results, demonstrating accurate and efficient detection of aircraft in the testing image across all datasets.

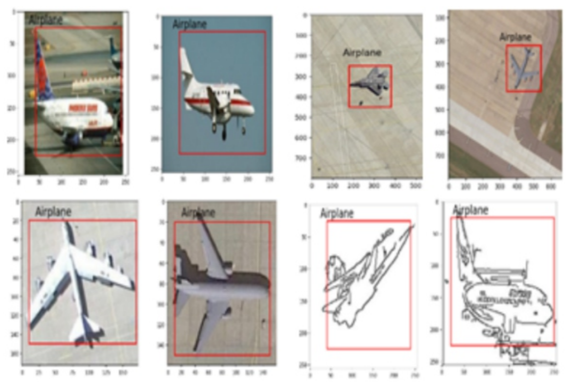


Figure 6. The Results of the Proposed System for Airplane Detection

5. Comparison and Discussion

To measure the efficiency of the proposed method with the chosen datasets, a comprehensive comparative analysis was conducted. This analysis involved contrasting the experimental results with recent research findings concerning the Caltech-101 and MTARSI datasets. The comparison encompassed factors such as dataset size, and the employed detection methodology. Table III presents a holistic view of these comparisons with relevant studies focused on aircraft detection.

TABLE III. A Comparison of the Proposed System with Recent Related Works: F1-score and Accuracy metrics

Reference	Dataset	Detection method	Evaluation metrics
[30]	MTARSI	VGG16	87.5% AC
[38]	Caltech	R-CNN	90.4% F1
[39]	Caltech	(MEsSP)	78.4% F1
[40]	Caltech	Modified Fuzzy C-Mean	70.9% F1
[41]	MTARSI	VGG16	72.1% AC
Proposed System	Caltech	VGG19-OCSVM	96% F1
	MTARSI		99% F1

Table III clearly presents the performance results on the Caltech-101 dataset. Akanksha et al. [38] achieved an F1- score of 90.4% using R-CNN. In contrast, Rafique et al. [39] attained a lower F1-score of 78.4% with their proposed approach by using Maximum Entropy scaled Super Pixels segmentation (MEsSP). Jalal et al. [40] used Modified Fuzzy C-Mean and Maximum Entropy and achieved an Fscore of 70.9%. Remarkably, the method introduced in this paper achieved an outstanding 96% F1-score, signifying a significant enhancement over previous methodologies.

Turning to the MTARSI dataset, Wu et al. [30] introduced MTARSI as the first public database for aircraft remote-sensing images. Their VGG-16 model achieved an accuracy of 87.5%. Similarly, Mo et al. [41] employed VGG-16 and reached an accuracy of 72.10%. Conversely, the approach suggested in this research demonstrated exceptional performance, achieving a flawless 99% F1-score, surpassing the performance of other methods.

This study is limited in its focus on detecting a single airplane within an image, without addressing scenarios involving multiple airplanes, which are common in real-world aerial imagery. This limitation reduces the generalizability of the proposed approach, particularly in complex situations where detecting and localizing multiple instances of the same object class is necessary. Future research could focus on extending the proposed method to effectively handle the detection of multiple airplanes within a single image, thereby enhancing its practical utility in applications such as aerial surveillance and reconnaissance.

6. Conclusion

This paper presents an approach for detecting airplanes, utilizing deep learning techniques and a transfer learning methodology. Initially, VGG19 model was employed for feature extraction. Then, the feature vector that has been acquired is afterward inputted into the OCSVM algorithm for aircraft detection. Following numerous experiments, the results obtained from the proposed system demonstrated notable performance improvements compared to traditional unmodified systems (OCSVM, VGG19) and recent related works.

References

[1] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, "A review of object detection based on deep learning," Multimedia Tools and Applications, vol. 79, pp. 23 729–23 791, 2020.

[2] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," Proceedings of the IEEE, vol. 111, no. 3, pp. 257–276, 2023.

[3] F. Chen, R. Ren, W. Xu, and T. Van de Voorde, "Airplane recognition from remote sensing images with deep convolutional neural network," in IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2020, pp. 264–267.

[4] L. Shi, Z. Tang, T. Wang, X. Xu, J. Liu, and J. Zhang, "Aircraft detection in remote sensing images based on deconvolution and position attention," International Journal of Remote Sensing, vol. 42, no. 11, pp. 4241–4260, 2021.

[5] M. Abdulla and A. Marhoon, "Agriculture based on internet of things and deep learning," Iraqi Journal for Electrical and Electronic Engineering, vol. 18, no. 2, pp. 1–8, 2022.

[6] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikainen, "Deep learning for generic object detection: A survey," International journal of computer vision, vol. 128, pp. 261–318, 2020.

[7] Q. Liu, X. Xiang, Y. Wang, Z. Luo, and F. Fang, "Aircraft detection in remote sensing image based on corner clustering and deep learn ing," Engineering Applications of Artificial Intelligence, vol. 87, p. 103333, 2020.

[8] U. Alganci, M. Soydas, and E. Sertel, "Comparative research on deep learning approaches for airplane detection from very highresolution satellite images," Remote sensing, vol. 12, no. 3, p. 458, 2020.

[9] J. Wang, H. Xiao, L. Chen, J. Xing, Z. Pan, R. Luo, and X. Cai, "Integrating weighted feature fusion and the spatial attention module with convolutional neural networks for automatic aircraft detection from sar images," Remote Sensing, vol. 13, no. 5, p. 910, 2021.

[10] Q. Hu, R. Li, Y. Xu, C. Pan, C. Niu, and W. Liu, "Toward aircraft detection and fine-grained recognition from remote sensing images," Journal of Applied Remote Sensing, vol. 16, no. 2, pp. 024 516– 024 516, 2022.

[11] X. Xiao, H. Jia, P. Xiao, and H. Wang, "Aircraft detection in sar images based on peak feature fusion and adaptive deformable network," Remote Sensing, vol. 14, no. 23, p. 6077, 2022.

[12] X. Chen, J. Liu, F. Xu, Z. Xie, Y. Zuo, and L. Cao, "A novel method of aircraft detection under complex background based on circular intensity filter and rotation invariant feature," Sensors, vol. 22, no. 1, p. 319, 2022.

[13] Z. Liu, Y. Gao, Q. Du, M. Chen, and W. Lv, "Yolo-extract: Improved yolov5 for aircraft object detection in remote sensing images," IEEE Access, vol. 11, pp. 1742–1751, 2023.

[14] X. Zhang, D. Hu, S. Li, Y. Luo, J. Li, and C. Zhang, "Aircraft detection from low scnr sar imagery using coherent scattering enhancement and fused attention pyramid," Remote Sensing, vol. 15, no. 18, p. 4480, 2023.

[15] A. Karacı, "Vggcov19-net: automatic detection of covid-19 cases from x-ray images using modified vgg19 cnn architecture and yolo algorithm," Neural Computing and Applications, vol. 34, no. 10, pp.8253–8274, 2022.

[16] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using vgg19: Caltech-101 image data set," Journal of ambient intelligence and humanized computing,

pp. 1–12, 2023.
[17] N. L. Tun, A. Gavrilov, N. M. Tun, H. Aung et al., "Remote sensing data classification using a hybrid pre-trained vgg16 cnn-svm classifier," in 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). IEEE, 2021, pp. 2171–2175.
[18] H. Zhang and X. Hong, "Recent progresses on object detection: a brief review," Multimedia Tools and Applications, vol. 78, pp. 27 809–27 847, 2019.
[19] V. Sonde, P. Shirpurkar, M. Giripunje, and P. Ashtankar, "Experimental and dimensional analysis approach for human energy required in wood chipping process," in International Conference on Advanced Machine Learning Technologies and Applications. Springer, 2020, pp. 683–691.
[20] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," Digital Signal Processing, vol. 126, p. 103514, 2022.
[21] F. Harrou, A. Dairi, B. Taghezouit, and Y. Sun, "An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class support vector machine," Solar Energy, vol. 179, pp. 48–58, 2019.
[22] Y.-P. Zhao, G. Huang, Q.-K. Hu, and B. Li, "An improved weighted one class support vector machine for turboshaft engine fault detection," Engineering Applications of Artificial Intelligence, vol. 94, p. 103796, 2020.
[23] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and grey wolf optimization for iot botnet detection," Journal of Ambient Intelli gence and Humanized Computing, vol. 11, no. 7, pp. 2809–2825, 2020.
[24] H.-J. Xing and L.-F. Li, "Robust least squares one-class support vector machine," Pattern Recognition Letters, vol. 138, pp. 571– 578, 2020.
[25] H.-J. Xing and W.-T. Liu, "Robust adaboost based ensemble of oneclass support vector machines," Information Fusion, vol. 55, pp. 45–58, 2020.
[26] N. M. Almoosawi and R. S. Khudeyer, "Resnet-34/dr: a residual convolutional neural network for the diagnosis of diabetic retinopathy," Informatica, vol. 45, no. 7, 2021.
[27] S. S. Basha, S. K. Vinakota, V. Pulabaigari, S. Mukherjee, and S. R. Dubey, "Autotune: Automatically tuning convolutional neural networks for improved transfer learning," Neural Networks, vol. 133, pp. 112–122, 2021.
[28] S. Aggarwal, S. Gupta, A. Alhudhaif, D. Koundal, R. Gupta, and K. Polat, "Automated covid-19 detection in chest x-ray images using fine-tuned deep learning architectures," Expert Systems, vol. 39, no. 3, p. e12749, 2022.
[29] J. A. Alhijaj and R. S. Khudeyer, "Integration of efficientnetb0 and machine learning for fingerprint classification," Informatica, vol. 47, no. 5, 2023.
[30] Z.-Z. Wu, S.-H. Wan, X.-F. Wang, M. Tan, L. Zou, X.-L. Li, and Y. Chen, "A benchmark data set for aircraft type recognition from remote sensing images," Applied Soft Computing, vol. 89, p. 106132, 2020.
[31] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in 2004 conference on computer vision and pattern recognition workshop. IEEE, 2004, pp. 178–178.
[32] B. Zhao, W. Tang, Y. Pan, Y. Han, and W. Wang, "Aircraft type recognition in remote sensing images: Bilinear discriminative extreme learning machine framework," Electronics, vol. 10, no. 17, p. 2046, 2021.
[33] Z. A. Khalaf, S. S. Hammadi, A. K. Mousa, H. M. Ali, H. R. Alnajar, and R. H. Mohsin, "Coronavirus disease 2019 detection using deep features learning." International Journal of Electrical & Computer Engineering (2088-8708), vol. 12, no. 4, 2022.

[32] B. Zhao, W. Tang, Y. Pan, Y. Han, and W. Wang, "Aircraft type recognition in remote sensing images: Bilinear discriminative extreme learning machine framework," Electronics, vol. 10, no. 17, p. 2046, 2021.
[33] Z. A. Khalaf, S. S. Hammadi, A. K. Mousa, H. M. Ali, H. R. Alnajar, and R. H. Mohsin, "Coronavirus disease 2019 detection using deep features learning." International Journal of Electrical & Computer Engineering (2088-8708), vol. 12, no. 4, 2022.
[34] H. Moradmand, S. M. R. Aghamiri, and R. Ghaderi, "Impact of image preprocessing methods on reproducibility of radiomic features in multimodal magnetic resonance imaging in glioblastoma," Journal of applied clinical medical physics, vol. 21, no. 1, pp. 179–190, 2020.
[35] H. Jia, Q. Guo, J. Chen, F. Wang, H. Wang, and F. Xu, "Adaptive component discrimination network for airplane detection in remote sensing images," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 7699–7713, 2021. [36] M. Weiss, S. Staudacher, D. Becchio, C. Keller, and J. Mathes, "Steady-state fault detection with full-flight data," Machines, vol. 10, no. 2, p. 140, 2022.
[37] Z. A. Khalaf and I. Sheet, "News retrieval based on short queries expansion and best matching," J. Theor. Appl. Inf. Technol, vol. 97, no. 2, pp. 490–500, 2019.
[38] E. Akanksha and P. R. Koteswara Rao, "A feature extraction approach for multi-object detection using hog and ltp." International Journal of Intelligent Engineering & Systems, vol. 14, no. 5, 2021.
[39] A. A. Rafique, M. Gochoo, A. Jalal, and K. Kim, "Maximum entropy scaled super pixels segmentation for multi-object detection and scene recognition via deep belief network," Multimedia Tools and Applications, vol. 82, no. 9, pp. 13 401–13 430, 2023.
[40] A. Jalal, A. Ahmed, A. A. Rafique, and K. Kim, "Scene semantic recognition based on modified fuzzy c-mean and maximum entropy using object-to-object relations," IEEE Access, vol. 9, pp. 27 758–27 772, 2021.
[41] H. Mo and G. Zhao, "Ric-cnn: Rotation-invariant coordinate convo lutional neural network," Pattern Recognition, vol. 146, p. 109994, 2024.

Marwa A. Hameed : She completed her undergraduate studies in computer science at the College of Science in 2011. Currently, she is pursuing a master's degree in Computer Sciences at the College of Computer Science & Information Technology, Basrah University, Basrah, Iraq.

Dr. Zainab A. Khalaf : She is a lecturer in the Department of Computer Science at the University of Basrah. She completed her PhD at the University of USM in Malaysia in March 2015. Her research interests include artificial intelligence, natural language processing (NLP), data mining, speech recognition, and computer vision.

ISSN 2210-142X

A Novel Framework for Mobile Forensics Investigation Process

Mohammed Moreb1, Saeed Salah2 and Belal Amro 3
1 Smart University College for Modern Education, Hebron, Palestine, P.O. Box 777
2Department of Computer Science, Al-Quds University, Jerusalem, Palestine, P.O. Box 20002
3College of Information Technology, Hebron University, Hebron, Palestine, P.O. Box 40

ABSTRACT

Investigating digital evidence by gathering, examining, and maintaining evidence that was stored in smartphones has attracted tremendous attention and become a key part of digital forensics. The mobile forensics process aims to recover digital evidence from a mobile device in a way that will preserve the evidence in a forensically sound condition. This evidence might be used to prove being a cybercriminal or a cybercrime victim. To do this, the mobile forensics process lifecycle must establish clear guidelines for safely capturing, isolating, transporting, storing, and proving digital evidence originating from mobile devices. There are unique aspects of the mobile forensics procedure that must be considered. It is imperative to adhere to proper techniques and norms for the testing of mobile devices to produce reliable results. In this paper, we develop a novel methodology for the mobile forensics process model lifecycle named Mobile Forensics Investigation Process Framework (MFIPF) which encompasses all the necessary stages and data sources used to construct the crime case. The developed framework contributes to identifying common concepts of mobile forensics through the development of the mobile forensics model that simplifies the examination process and enables forensics teams to capture and reuse specialized forensic knowledge. Furthermore, the paper provides a list of the most commonly used forensics tools and where we can use them in our proposed mobile forensic process model.

Keywords: Mobile Forensics, Digital Forensics, Forensic tools, Acquisition, iOS, Android, Extraction, Artifact.

1. Introduction and Overview

In the current era of the digital age, it is undoubtedly shown that mobile applications have profoundly trans formed every aspect of human lives. Users are now relying on mobile applications to do many online activities such as browsing the internet, shopping, transferring money, doing business, communicating using audio or video calls, texting, entertainment, and education. This massive growth of smart phone usage is still incredibly popular and will continue to be for the foreseeable future. According to Figure 1, the annual sales of smartphones have tremendously increased to around (1.56) billion devices worldwide, smartphones running the Android operating system held an (87%) share of the global market in 2019 and this is expected to increase over the forthcoming years, while Apple iOS; the second most popular operating system has a (13%) market share across all devices. With this tremendous use of smartphones worldwide, the wide adoption of these devices to carry out technology-oriented services, and the uncontrolled use of mobile applications have turned the mobile environment into a fertile spot to carry out many unethical and illegal activities. Consequently, smartphones became a famous target for cyber-attacks bearing in mind that these devices contain private data [1]. The portability of these devices and the sensitivity of the data they contain raised great concern about the feasibility of using traditional digital forensic methodologies and to what extent they fit this field [2]. Smartphones are equipped with many capabilities that make forensic steps difficult to handle and require great attention. These capabilities include the availability of different communication technologies such as Short Message Service (SMS), 3G, Wi-Fi, Global Positioning System (GPS), etc., the ability to remotely instruct the device to switch on or off, and the ability to remotely wipe data using different mobile applications. These issues and others created a big challenge for the investigators when dealing with mobile digital evidence [3]. In this regard, a set of terminologies, definitions, and legal issues have appeared that describe the new criminal situations raised due to this new computing paradigm. One of these terminologies is digital forensics which refers to the process of collecting digital evidence from a digital device and analyzing it to prove the guilt or innocence of persons [4]. Mobile forensics is another terminology derived from digital forensics; it aims to recover digital evidence from a smartphone in a way that will preserve the evidence in a forensically sound condition. To conduct mobile forensics analysis, the mobile forensic process lifecycle needs to set out precise rules that will seize, isolate, transport, store, and proof of digital evidence safely originating from

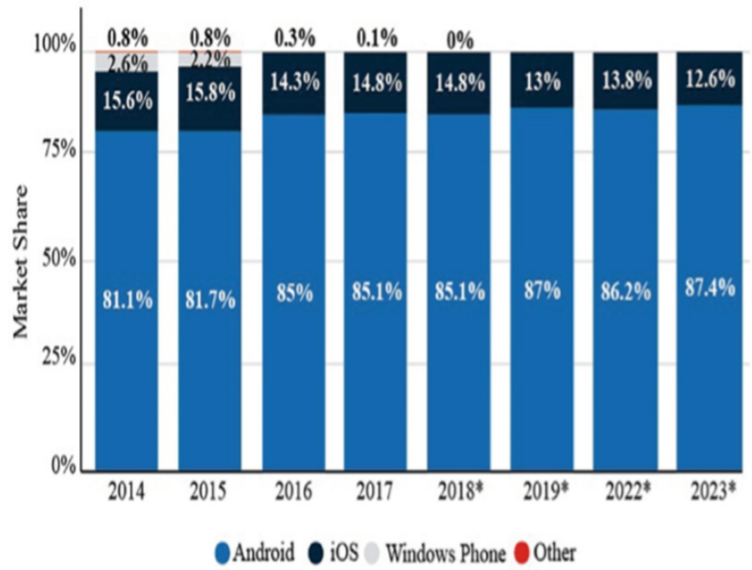


Figure 1. Share of global smartphone shipments by operating system from 2014 to 2023 [5]

smartphones. Mobile forensics investigation frameworks are essential for gathering, examining, and preserving digital evidence from mobile devices in a forensically sound manner, the research [6] discussed the challenges faced by forensic investigators in extracting data from mobile devices and suggested a new model for mobile forensic acquisition, but it does not provide a detailed explanation of a mobile forensics investigation framework. [7] helped in simplifying the examination process and enabled the capture and reuse of specialized forensic knowledge, the result compared the forensics investigation framework for the association of Chief Police Officers (ACPO) and Digital Forensic Research Workshop (DFRWS) frameworks and DFRWS, which has the most complete stages to support the investigation process, DFRWS includes five substations of digital forensics in general [7]. DFRWS framework is considered one of the best frameworks, as it encompasses all the necessary stages and data sources used to construct a crime case, the results show that the Belkasoft Evidence Center forensic tool has the highest accuracy rate of 78.69%, while Magnet AXIOM has an accuracy rate of 26.23% and MOBILedit Forensic Express has an accuracy rate of 9.84%. [8] supports the investigation process by providing a comprehensive set of stages. The use of mobile forensics tools, such as Belkasoft Evidence Center, Magnet AXIOM, and MOBILedit Forensic Express, can aid in the extraction of digital evidence from mobile ap plications like Signal Messenger. These tools have varying levels of accuracy and capabilities in recovering different types of data. Overall, the development and use of mobile forensics investigation frameworks and tools are crucial for effectively investigating and analyzing digital evidence from mobile devices. This research introduced the frameworks that provide guidelines for capturing, isolating, transporting, storing, and proving digital evidence including all details as a novel framework used by mobile forensics to extract and present results. The process of digital forensics has become an important analysis and systematic approaches were proposed and adopted by many specialized governmental and private organizations and institutions such as The American Academy of Forensic Sciences (ACFS), the European Network of Forensic Science Institutes (ENFSI), the International Institute of Certified Forensic Investigation Professionals (IICFIP), and many other institutions worldwide. Besides, there are some well-known standards and good practices designed for digital forensics such as the two standards provided by ATSM [9], where issues related to digital forensics education challenges are provided as well as specifying the digital forensics steps with details about the requirements for each step. As thoroughly explained in the literature, the digital forensics process is divided into the following steps, these steps are common in most references with some slight modifications of the details and functionalities of the following steps:

(i) Identification, this step involves finding the evidence and where the required data is located;
(ii) Preservation, in this step, the evidence is isolated, secured, and data is preserved as well. Access to the evidence and data is allowed only for investigators who are working on the case to prevent people from tampering with the data and hence making the evidence illegal;
(iii) Analysis, in this step, the reconstruction of evidence fragments is performed and conclusions about the evidence are found;
(iv) documentation, a record of all the required data is preserved; this record can be used to recreate the crime scene;
(v) presentation, a summary of the case and the conclusion are performed at this step,
(vi) Case Closure, in this step, the case is closed by having a legal decision and the evidence is returned or archived accordingly.

These steps may vary in their details from one institution to another; however, all of them will lead to a similar sequence of steps that will finally lead to a successful handling of digital crime. The mobile forensics process has its particularities that need to be considered. Thus, following a correct methodology and guidelines are vital preconditions for the examination of smartphones to yield good results. In this paper, we develop a novel methodology for the mobile forensics process life cycle called Mobile Forensics Investigation Process Framework (MFIPF) encompassing all the necessary stages and data sources used to construct the crime case. The developed methodology will contribute to identifying common concepts of mobile forensics through the development of the mobile forensics model that simplifies the examination process and enables forensics teams to capture and reuse specialized forensic knowledge, furthermore, it reduces the difficulty and ambiguity in the mobile forensics domain. Unlike other models, this proposal divides the evidence life cycle into several modules and describes each module along with its main components, data sources, tools, intra-module, and intermodule interactions easily and clearly. The rest of the paper is organized as follows. Section 2 discusses the related work including the most common mobile forensic process models as well as common mobile forensics tools. Section 3 details the proposed mobile forensics process model (MFIPF), describing its various modules and sub-modules and their connectivity and the associated data sources, mechanisms, and tools. In section 4, the common mobile forensic tools are classified and mapped to our proposed model based on their applicability at different stages. In Section 5, we conclude the paper and outline some ongoing and

and future research lines.

A. Related work

In this section, a brief review of the related literature will be conducted. First, we will introduce the work done in mobile forensics models and stages, and then, we will talk about the common tools used in mobile forensics.

1) Mobile Forensics models and phases

Due to the previously mentioned reasons and challenges, many researchers have proposed some specific mobile forensics procedures and methods to deal with special mobile investigation cases. The existence of such methods is important for the success probability of an investigation and the avoidance of corrupting the evidence or failing to extract some necessary information. Among these proposed is a model proposed by Moreb [10], where the author discussed the four process phases used for conducting mobile forensics, are (i) the identification phase which includes many details such as identifying, acquiring, and protecting the data collected at the crime scene; (ii) the collection phase which starts by processing the collected data or evidence, then extracting the relevant information; (iii) the analysis phase analyzes the extracted informationto connect the dots and be able to build a robust and admissible case, and (iv) the reporting phase is the final step that presents the findings of the analysis stage into an admissible and understandable format. In [11], the authors mentioned that there are five phases in the forensic process (identification, preservation, acquisition, analysis, and reporting) which are similar to what was proposed by Moreb [10]. The study [12] concentrated on android forensics and proposed a framework of seven stages namely: Intake, Identification, Preparation, Isolation, Processing, Verification, and Documentation. A comparative analysis of five common process models was provided by [13], these models are the Smartphone Forensic Investigation Process Model (SFIPM), Windows Mobile Device Forensic Model (WMDFM), National Institute of Standards and Technology (NIST), Harmonized Digital Forensic Investigation (HDFI), and USFIPM. The authors also proposed a secure model by deploying blockchain using Ethereum or a hyperledger platform. In [14], the authors proposed an Efficient and Reliable Forensics Framework (ERFF), which helps the investigator to securely obtain evidence more easily, ERFF is an efficient and reliable forensics framework as comparedwith other frameworks such as SNIF, LFCCF, and LRFF. It uses edge computing to improve reliability, efficiency, and accuracy. Moreover, it helps identify criminal activities more quickly using low-cost edge devices and involves a detective module and a validation model that detects the interaction between a client terminal and the edge resource. In [15] an analysis of the forensic-by-design framework is proposed which includes investigating the limits of the forensic-by-design and its Insufficiency that could be rewritten as ”deficiencies” or ”shortcomings”. Please let me know if you need any further help with this. in a Cloud systems context, and it proposes three new forensic-by-design key factors and associated standards and best practices, it also suggests a new generic systems and software engineering driven forensic-by-design framework. In [16], the Goel authors demonstrate the DFWM that provides a general and updated description of the DF investigation process at the workflow level and can be used as a management tool for unboxing the procedures, tasks, and risks involved in the workflow of the individual DF investigations. Using the investigative strategy for the specific case, DFWM serves as a framework for packaging the digital forensic investigation process, providing a detailed structure and visualization of the physical and investigative chores and decisions. DF workflow which guided by the overall investigative strategy of the particular case as follows:

(i) Review of client requirements and planning stage,

(ii) Evaluation of deployed workflow stage,
(iii) Identify the physical and cognitive tasks, and
(iv) Make decisions and their associated risks at the respective stage.

Based on the existing process and models, the layered framework for mobile forensics is proposed [17], the results have shown that using only one tool is not sufficient to complete the investigation process, the four layers are organized as a framework, the number of layers can be increased or reduced as per the case type, the six layers can be grouped to small categories with tools to use for each one as acquisition process with various tools such as MOBILedit, Bulk extractor; data analysis is carried out with various tools like Autopsy and CellDEK, and reporting the case can be generated using MOBILedit Forensic and CellDEK. In [18], the authors reviewed about 100 Mobile forensics models with the main conclusion that suggests improving and validating the investigation process model, developing a meta-modeling language, and developing a definite mobileforensics source to store and retrieve the knowledge formed in the mobile forensics field. Many forensics investigation process models are used for the Internet of Things (IoTs) such as CIPM for IoTFs [19], the proposed model assists IoTF users in facilitating, managing, and organizing the investigation tasks, it consists of four common investigation processes, preparation process, collection process, analysis process, Patiland report process. The roadmap of DFIP discovery of tools [20] discussed in detail the challenges and opportunities of the digital forensics process concerning different fields such as networks, IoT, cloud computing, database systems, big data, mobile and handheld devices, disk and different storage media, and operating system. As seen from the literature, there is a necessity for adopting a robust model to carry out mobile forensic investigations efficiently.

2) Mobile forensics tools

The definition of mobile phone forensics is the science of extracting digital evidence from a mobile device [21]. It provided a wonderful list of resources for catching online criminals who utilize mobile devices for illegal purposes. With their vast number of applications and current proper ties, mobile devices' ever-increasing storage and processing power provide new hurdles for digital forensics [22]. To collect digital evidence for use in court trials, mobile forensic tools and applications are essential. They can unearth call metadata, SMS, GPS data, application data, and locally stored files. A set of mobile forensics tools [23] can be used such as Cellebrite UFED Physical Analyzer and Oxygen Forensic Suite to get details about the mobile device, Oxygen and UFED forensic tools [24] are used to recover app data. In general, digital forensic tools for data extraction are categorized into three types: manual, logical, and physical [25]. Many mobile forensics tools [26] such as Belkasoft Evidence Center [27], FINALMobile Forensics [28], 3uTools [29], and Magnet [30] are used to extract artifacts from both Android and iOS devices. The SDCA [31] tool is designed to perform the analysis of the differences between two versions of SQL schema, in addition to its ability to analyze the query. According to [32], SecureRS aided forensic investigation in general, by developing a model and a platform to secure potential digital evidence, the SecureRS model can help to prevent unauthorized access and comply with regulations and pri vacy policies, and the result shows a method of ensuring forensically sound digital evidence for DFR as well as for digital forensics processes in general. In [10] the authors discussed the tools used to acquire the data from iOS or Android devices for both rooted and jailbroken mobile. The work of [33]found that the data used in the media directory will not change even after jailbreaking the device, which means that the integrity of the data is maintained. As a result of this study, jailbreaking is considered acceptable to help forensic tools extract more data while preserving user data. There was a previous study in the use of forensic tools in the process of acquiring data on iOS, Android, andWindows using forensic tools Oxygen and UFED to recover applications' data, and the tools were able to restore the list of contacts that WhatsApp installed on iOS and Android and were unable to recover anything from the Windows device. In addition to the ability of the tools to restore and decrypt the backups of the Android and iOS devices, and were unable to find the encryption key for the Windows device. The result was that it could restore conversations even if the application has been deleted if there are backup copies stored on the device for WhatsApp [24]. In [31] it is noted that the developers of forensic tools have limited knowledge of the changes that have occurred to the SQL Lite schema for iOS backups and need to preserve the tools' compatibility with recent versions. The SDCASQLite Database Comparison Analyzer (SDCA) tool is designed to perform the analysis of the differences automatically between two versions of SQL schema, in addition to its ability to analyze the query, it also demonstrates that using the tool is feasible to update the Forensic Targeted Data Extraction Application called FTDEA developed by the authors. As mentioned in [34] the growth of using smartphones from 2016 until 2021 increased from 2.5 to 3.8 billion smartphones. As reported by [35], the number of users who use social media is about 4.20 billion active users worldwide. According to the comparison as shown in. Commercial and open-source forensic tools are available for mobile device investigations. The availability of many mobile forensics tools might cause some dilemmas in the selection of the best tool, for this reason, details about these tools will be provided in Section 4.

B. Proposed mobile forensics framework

In this section, we will deeply describe our MFIPF provided in Figure 2. The stages of the framework (Data Preparation, Information Analysis, Case construction, and Case Closing) will be explained showing the detailed steps at each phase.

1) Data Preparation

The data preparation phase aims to generate a processed dataset that is technically usable for the analysis phase. In this phase, four steps are carried out to guarantee that the acquainted data is gathered systematically and legally. The four steps shown in Figure 2 are described below:

a) Resource seizure

In this step, the mobile device is seized in a way that guarantees that the device will not be modified and there should be no ability to connect with the device. To achieve this step, we have to follow the following process [36]:

(i) issuance of research warrant from legal representatives;
(ii) turning off all wireless communications and putting the mobile device in Airplane Mode; (iii) shielding the mobile device in a Faraday bag that prohibits any external signals to reach the mobile, and (iv) Document these steps and send the mobile device to the digital forensics lab for investigations.

b) Resource identification

Once the mobile device arrives at the digital forensics lab, the resource identification process is carried out. The process aims to identify the mobile device under investigation and choose the suitable tools that can be used for the data extraction phase. A description of the mobile device is provided here, the description includes the model and type,

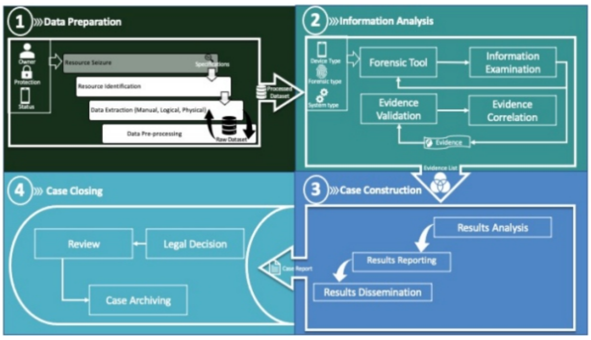


Figure 2. The proposed Mobile Forensics Investigation Process Framework (MFIPF)

physical status (if the device is broken), and logical status (the device is on or off, the device is functioning or not). Based on this information, the investigator will be able to determine the suitable tools required for the data extraction process. This process should be formally documented [37].

c) Data extraction

This is a very important process where the data is extracted from the mobile device, the extracted data will then be used in further stages to extract evidence. The information gathered in the identification phase is the basis of the data extraction method to be used, this method includes:

1) Manual data extraction: here the investigator manually navigates the mobile device to search for the required evidence; documentation of this process is essential and might be done by video recording of the screen of the mobile device during the navigation process [37]. It is important here for the investigator to conduct the boundaries of the research warrant and never explore data that is not included in the research warrant. This process requires the ability of the investigator to access the device by having the password or pattern. It is worth mentioning here that manual data extraction will affect the integrity of the files and hence the investigator should precisely document the steps he took and the findings as well.

2) Logical extraction: When applying this method, the investigator will be able to generate a copy of the file system that can be used later to extract data using some tools designed for this purpose. This copy will enable the investigator to view the same data that can be generated using manual extraction [38]. However, this method does not affect the integrity of the files of the mobile device and the investigator can only work on the copy of the files and the original device will be kept safely in an evidence container.

3) Physical extraction: in this method, a raw image in a binary format of the mobile device's memory is generated, and the output is a bitwise copy of the memory of the mobile device [39]. This copy includes all system files and can also be used to retrieve some of the deleted files as well. However, to generate this copy usually we need to root the device which will affect the integrity of the evidence, so the investigator has to document the details of this step. The generated copy can then be used to retrieve system files as well as some of the deleted files using dedicated data analysis tools.

It is worth noting that the aforementioned methods can be applied only when the mobile device is functional, i.e., not broken, and does not work for broken or malfunctioning mobile devices. In such a case some other methods might be used such as chip-off by which the memory chip of the mobile device is physically removed and attached to a memory reader, or a similar device, and the data is then extracted [40]. This method requires high skills in electronic device maintenance and may cause the chip to be destroyed if not removed or attached correctly. Another extremely hard method that might be used in very rare cases such as national security is called Micro-read where an electronic microscope is used to read the contents of the memory on gate level base [41]. This method is very expensive and takes too much time but might be used to extract some data from broken devices.

d) Data preprocessing

In this process, the characteristics of the mobile device operating system are studied, and data is categorized based on applications to pinpoint potential evidence(s). Classification techniques are used here to group data based on file system analysis and system log analysis. The output of this process is a well-prepared dataset that can be used in the analysis stage to extract evidence. The preprocessing step might also include putting the data in a proper file format that is compatible with mobile forensics tools in the analysis [42].

2) Information analysis

In the analysis phase, evidence(s) is/are extracted by formally interpreting the information generated by the previous phase – data extraction-. The investigator should follow standards and best practices in the field of forensic analysis so that the evidence will be intact, and results are reproducible and acceptable. For a robust mobile forensic analysis, the following steps are suggested to be followed:

a) Selection of the Forensic Tools

The first step in the analysis includes the selection of a forensic tool. The selection of the tool depends on many factors including cost, user interface, the familiarity of the examiner, computing platform, environment, and legislative –whether the tool is legally approved or not [43]. A list of mobile forensics analysis tools and their properties are provided in Section 4. Typically, the examiner may use different tools to generate different information and events, there is also a possibility to use different tools to generate the same event to ensure the use and follow up of reproducibility of the event and to prove its validity [44] Therefore, an examiner should be familiar with different tools to conduct his analysis successfully.

b) Information examination

After selecting the appropriate tool(s), the examiner will feed the tool with the preprocessed data and perform a variety of tests and processing tasks against the data. The process aims to generate an event from the evidence file. There might be many events generated from the same or multiple tools. These events are then stored and fed to the next step which is evidence validation [45], Events in a mobile device might be found at different locations according to the information the examiner is trying to find. Some of the events might be found in SMS and call logs, others might be found in saved pictures or emails. Some complex events might require retrieving deleted files using special tools while other events require the use of different tools and gathering information to reconstruct that event. The selection of the tool and the process depends on the examiner and requires skilled persons to successfully perform the task [41].

c) Evidence validation

According to [46], validation is the process of proving the validity of the evidence to a jury. The process implies proving acceptable error rates as well as using scientifically proven valid data, applications, and results. The validation process is applied to all stages in mobile forensics and covers data collection and storage, system, application, user, and algorithm applicability validation. A very important issue related to validation is the use and following up of standards and best practices developed for this purpose. Many countries have developed standards for digital and mobile forensics through their dedicated institutions such as NIST in the states. Besides, some well-known digital forensics developers have also proposed some best prac tices that are proven to generate valid evidence with an acceptable error rate [47]. The examiner must follow these standards and verify the validity of the evidence during the entire investigation process.

d) Evidence correlation

Correlation involves the ability to extract the semantics from different sources such as SMS, social media messag ing, emails, . . . , etc, and to generate a knowledge base that clearly shows the correlation among these generated events. Domain and application ontology's might be used to correlate different events to a knowledge base [48].

Event correlation and reconstruction might be carried out using different techniques and technologies including rule based, semantic models, tree/graph-based, timestamp-based, finite state machines, and live event construction [49], such techniques aim to construct valid evidence from different sources of events with acceptable error rate. The output of this stage will be used as input for the next phase which is case construction.

3) Case Construction

The output of the second stage - information analysis - is fed as an input to the case construction stage, which takes the evidence list to prepare results and move towards closing the case. Four steps are necessary in the process of case construction: results analysis, results examination, results reporting, and results dissemination. In what follows, a detailed explanation is provided for each step.

a) Results analysis

In this step, examiners must analyze all the technical findings extracted from the information analysis phase consistently and clearly. When analyzing the results, examiners can divide the analysis sequential logical parts into multiple headings and comment on results as they are described to ease the decision-making process, the results could be supported by figures, tables, and equations to enrich the findings. In addition, the results' conclusion must be kept very brief and aggregates the findings with robust paragraphs [50]. During the process of validating the results of a mobile forensic scene, several methods can be used to verify the validity of the results such as calculating the hash value with two different forensics tools, or the various steps might be revisited using the same tool to obtain the digital evidence and recalculate the hash value to validate the results. At some point, the results generated using experimental and validation stages must be repeatable. Any variable that might affect the outcome of the validation should be determined after several test runs. However, some cases require more runs to generate valid results, and; examiners need to utilize the literature to assess the results' validations [51].

b) Results reporting

The most fruitful result that should be created following the forensic process is the documentation of the findings. Once completed, investigators can use the report to their advantage in several ways:

- Sharing the results with other investigators and decision-makers.
- Communicating the facts that may support the investigation of other cases.
- Offering a clear justification for gathering more digital evidence.
- Using the report to evaluate the specific case.

The final report must be written by digital examiners considering all conditions and guidelines established by national law. To ensure that the report complies with the law, they must first independently review it. Any divergent opinions will eventually be examined for flaws to bolster the assertions. In general, there is no set format or structure for reporting the findings, but any final report must include the bare minimum of the following data: jurisdiction, the nature of the case, the court's document format, and the reason ID, calendar of all depositions (timestamps), deponent's name and ID, and other details like time and date the case created, phone physical situation, the phone status on or off, mobile manufacturer information, pictures for each accessory and the phone itself, which tools used

TABLE I. Mapping iOS and Android forensic tools with the MFIPF framework.

Phases	Capability	Magnet AXIOM		FINALMobile		BelkaSoft		MOBILedit	
		IOS	Andriod	IOS	Andriod	IOS	Andriod	IOS	Andriod
Phase 1: Data Preparation	Logical Imaging	✔	✔	✔	✔	✔	✔	✔	✔
	Physical Image	✖	✖	✖	✖	✔	✔	✔	✔
	Manual	✔	✔	✔	✔	✔	✔	✔	✔
Phase 2: Information Analysis	SQLite	✔	✔	✔	✔	✔	✔	✔	✔
	Hash-Comparison	✔	✔	✔	✔	✔	✔	✔	✔
	Retrieves Deleted Files	✔	✔	✔	✔	✔	✔	✔	✔
	Information examination	✔	✖	✔	✖	✔	✔	✔	✔
	Evidence validation	✔	✔	✔	✔	✔	✔	✔	✔
Phase 3: Case Construction	Results examination	✖	✖	✖	✔	✔	✔	✔	✔
	Results analysis	✔	✔	✔	✔	✔	✔	✔	✔
	Results reporting	✔	✔	✔	✔	✔	✔	✔	✔
	Results dissemination	✖	✖	✖	✖	✖	✖	✔	✔
Phase 4: Case Closing	Case Archiving	✖	✖	✖	✖	✖	✖	✖	✖
	Legal Decision	✔	✔	✔	✔	✖	✖	✔	✔
	Categorization	✔	✔	✔	✔	✖	✖	✖	✖
	Advance Search	✔	✔	✔	✔	✔	✔	✖	✖

in the investigation, any additional data added during an examination. Many forensics reporting tools provide ways to automatically annotate evidence fragments and generate automatic reports according to the examiner's configuration. These tools enable the examiner to perform sub-functions such as tagging, bookmarking, log reports, or even report generation. The report relies on solid documentation, photos, notes, and tool-generated content. The examiner should then check the report and edit his configuration if necessary [52].

c) Results dissemination

It describes the procedure the examiner uses to communicate to policymakers the findings from the analysis phase. The major goal of this method is to provide action reports for each detected artifact and its analysis. The investigator's defensive strategy and any potential implementation difficulties can also be included in the presentation phase. In an iterative approach, the results from this phase might be used to conduct additional acquisitions. As a result, each process produces more analytical artifacts, which are then provided as feedback to other processes. For lengthy criminal investigations, this feedback iterative procedure may go through numerous iterations. This step might helpother investigators working on similar cases to proceed with their cases accordingly, or to criticize the case, and hence further steps might be required to be performed for the disseminated case [53].

4) Case Closing

Case closing is the last stage in the mobile forensics investigation process framework (MFIPF) which undergoes three main steps to ensure the successful termination of the process model. They are case closing, making the legal decision, and case archiving. Understanding how to close and archive the case is also crucial to performing a targeted analysis of the data for future updates. The digital examiner must have good knowledge of how to store and collect similar cases which might help in case examination.

a) Legal decision

The constructed case should be finally put in its legal context, here, the final legal decision should be a judicial determination of all parties' rights and obligations reached by a court based on facts and law. A decision can mean either the act of delivering a court's order or the text of the court's opinion on the case and the accompanying court after you complete a case. Since every user owns his/her data and digital device, forensic examiners face ethical and legal issues in accessing and collecting the required information [54].

b) Review

The final step in the lifecycle is to review the case to identify successful decisions and actions and determine how the system performance should be improved in terms of time, and accuracy. Critique the case, self-evaluation, and peer review are essential parts of professional growth. Investigators must keep the OS and digital forensics tools up-to-date for everything to be consistent. This necessitates updating the OS frequently, installing all-new system updates and patches, and regularly checking the tools' websites for new updates or patches [55].

c) Case archiving

When work on a case is completed and immediate access to it is no longer necessary, that case can be archived. This step aims at closing the case after its resolution. Digital forensics cases include the storage of electronic copies of evidence as well as the case report and the generated artifacts and the documentation of the whole stages of the case. Case archiving aims to enable examiners to review the procedures carried out to use them in similar cases. The case archive should enable the examiner to reconstruct the case from scratch based on the available copies of the case evidence which will help if the case is legally re-opened [56]. Many tools might be used in case archiving that enable ease of use and retrieval of cases, some of these tools will be provided in Section 4.

C. Common mobile forensic investigation tools

In this section, we will explain a list of 4 commonly used mobile forensics tools and map them to our proposed model MFIPF.

1) Common tools

In the following, we list the common forensics investigation tools and compare and reflect on their operations with the modules of the proposed MFIPF framework.

• Belkasoft Evidence Center: It is a comprehensive forensic tool for locating, retrieving, and analyzing digital evidence stored on desktops and mobile devices. This tool makes it simple for investigators to collect, examine, analyze, preserve, and share digital evidence from computers and mobile devices. By analyzing hard disks, drive pictures, memory dumps, iOS, Blackberry, Android backups, UFED, JTAG, and chip-off dumps, the toolkit will efficiently extract digital evidence from many sources. It evaluates the data source automatically and lays out the most forensically significant artifacts for the investigator to study the case or add to the report [27].

• FINALMobile: It is a powerful software and mobile solution for legal inspectors that provides the legal community with the most cutting-edge data mining and information extraction capabilities. Thanks to its extensive understanding of system files and informa tion patterns, this software can transform raw data into executable and ready files in just a few clicks. On mobile devices, data is stored in specialized forms and is frequently left behind after a device is entirely cleaned. The FINALMobile forensics software can easily retrieve deleted (hidden) files by scanning for specific patterns. Additionally, as the majority of mobile devices adhere to the same pattern, data can be gathered for upcoming mobile devices [28].

• 3uTools: It is a program for flashing and jailbreaking Apple's iPhone, iPad, and iPod touch. It offers three ways to flash Apple mobile devices: easy mode, profes sional mode, or multiple flash. It automatically selects the proper firmware and supports a fast download speed. 3uTools can be freely downloaded for Windows PC Latest Version. It has a complete 3uTools offline setup installer [29].

• Magnet ACQUIRE: This tool combines an easy user interface with dependable and speedy extractions to provide you with the information you need quickly and effortlessly. Furthermore, the data quality will be max imized, and activity logging and documentation will help to understand which procedures were employed [30].

For comparative analysis between our approach and existing frameworks, we have utilized the comparison done by [13] who compared five forensic frameworks, they are SFIPM, WMDFM, NIST, HDFI, and USFIPM and NIST. Table 1 shows an updated version of this comparison including our approach as proof of its usability. Further more, Table 2 provides a comparative analysis between iOS and Android forensic tools for mobile forensics tools and their reflection on our proposed MFIPF based on a set of capabilities.

2) Practical Example of Using MFIPF Over a Digital Crime Case

It is worth mentioning here that MFIPF is a compre hensive model to be used during the mobile investigation process. As an example, we will assume that we are supposed to work on child pornography conducted using the suspect's WhatsApp account. Below practical example which summarizes the steps to be followed based on the proposed MFPIF model.

1) Data Preparation, a search warrant is issued. The device is seizure and a report of the device status is done: iPhone 8, 128G, iOS version 13.1.1 WhatsApp version 14.0.1. Assuming the device was on and we had access to it, we chose logical extraction using Mobile Edit Forensics Express

2) Information Analysis, we use Mobile Edit Forensics Express to analyze our image. A set of images and videos as well as conversations was found to contain child pornography. We may use another tool such as Belkasoft to perform the analysis and verify the results.Correlation among evidence might be done to find all victims and criminals from the contact list

TABLE II. A comparative analysis of five common forensics process models with the proposed one.

Phases	Capability	SFIPM	WMDFM	NIST	HDFI	USFIPM, NIST	MFIPF
Phase 1: Data Preparation	Preparation	✔	✔	✖	✔	✔	✔
	Handling and securing the evidence scene	✔	✔	✖	✖	✖	✖
	Mode selection shielding	✔	✖	✖	✖	✖	✔
	Offset/online storage	✔	✖	✖	✖	✖	✖
Phase 2: Information Analysis	Examination and analysis	✔	✔	✔	✔	✔	✔
	Cell state analysis	✔	✖	✖	✖	✖	✔
	Non-volatile evidence collection	✔	✔	✖	✖	✖	✔
	Volatile evidence collection	✔	✔	✖	✖	✖	✔
	Evidence validation	✔	✔	✔	✔	✔	✔
Phase 3: Case Construction	Presentation	✔	✔	✔	✔	✔	✖
	Communication Scheduling	✖	✔	✖	✖	✖	✖
Phase 4: Case Closing	Review	✔	✔	✔	✖	✔	✖
	Documentation	✖	✖	✔	✖	✔	✖
	Survey and Recognition	✔	✔	✖	✖	✖	✖

3) Case Construction, each evidence is analyzed and related to a victim and a list of contacts who were shared with each evidence is listed as well. This might lead to the identification of some suspects who might be colluding together. Results reporting is to be done, it might be done automatically using a specialized tool such as Mobile Edit Forensics express. Detectors will then conduct their interviews and interrogations with witnesses and suspects and come up with a final report to the corresponding agencies.

4) Case Closing, a legal decision is carried out; a Case Review is done for any new updates about the consid ered crime case. finally, case Archiving is the last step that saves the complete case for future reference.

2. Conclusion and future work

Cybercrimes are rapidly increasing due to the tremen dous reliance on information and telecommunication tech nologies. This rapid increase is being faced by develop ing the necessary tools and legislation to fight against these crimes. One of the most challenging investigation issues is mobile device forensics. This challenge is be cause mobile devices are becoming more powerful with tremendous processing and communication capabilities as well as containing sensitive data related to the mobile user. For these reasons, a framework for mobile device forensics must be developed to systematically engineer the investigation process and avoid any issues that might cause the rejection of the investigation. In this paper, we proposed a mobile forensics lifecycle called Mobile Forensics Inves tigation Process Framework (MFIPF). MFIPF encompasses all forensics stages and steps that must be followed in each stage. Furthermore, we also proposed a list of the most commonly used mobile forensics tools that might be used in each stage or step. In future work, we will apply this model to different investigation scenarios with different mobile platforms and report the findings and if necessary, we will update the model accordingly, we will also test the utility of using our model MFIPF with different mobile digital forensics scenarios and compare our utility results against other models.

References

[1] S. N. Zakaria and M. F. Zolkipli, "Review on mobile attacks: Op erating system, threats and solution," Borneo International Journal eISSN 2636-9826, vol. 4, no. 2, Jun. 2021. [Online]. Available: https://majmuah.com/journal/index.php/bij/article/view/81

[2] A. M. Alashjaee, N. Almolhis, and M. Haney, "Mobile malware forensic review: Issues and challenges," p. 367–375, 2021.

[3] M. Kumar, "Mobile phone forensics – a systematic approach, tools, techniques and challenges," International Journal of Electronic Security and Digital Forensics, vol. 13, no. 1, p. 53–63, 2021.
[4] O. Ameerbakhsh, F. M. Ghabban, I. M. Alfadli, A. N. Abuali, A. Al-Dhaqm, and M. A. Al-Khasawneh, "Digital forensics domain and metamodeling development approaches," 2021 2nd Interna tional Conference on Smart Computing and Electronic Enterprise: Ubiquitous, Adaptive, and Sustainable Computing Solutions for New Normal, ICSCEE 2021, p. 67–71, Jun. 2021.
[5] Www.statista.com, "Share of global smartphone shipments by operating system from 2014 to 2023," 2021. [Online]. Available: https://www.statista.com/statistics/272307/ market-share-forecast-for-smartphone-operating-systems/
[6] A. J. Karjagi and S. A. Quadri, "Design of a framework for data extraction and analysis from android-embedded smartphones," Russian Law Journal, vol. 11, no. 3, p. 3, Apr. 2023. [Online]. Available: https://typeset.io/papers/ design-of-a-framework-for-data-extraction-and-analysis-from-162hm86l
[7] I. Riadi, A. Yudhana, G. Pramuja, and I. Fanani, "Comparative analysis of forensic software on android-based michat using acpo and dfrws framework," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 7, no. 2, p. 286–292, Mar. 2023. [Online]. Available: https://typeset.io/papers/comparative-analysis-of-forensic-software-on-android-based-25qqa6xn
[8] J. R. Hildebrandt, E. M. Schomakers, M. Ziefle, and A. Calero Valdez, "Understanding indirect users' privacy concerns in mobile forensics — a mixed method conjoint approach," Frontiers in Computer Science, vol. 5, Jul. 2023. [Online]. Available: https://typeset.io/papers/ understanding-indirect-users-privacy-concerns-in-mobile-2l1zkdkm
[9] J. Howe, M. Baylor, and R. H. Liu, "Advancing the practice of forensic science in the united states–practitioners' efforts." Forensic Science Review, vol. 34, no. 1, p. 7–15, Jan. 2022.
[10] M. Moreb, "Introduction to android forensics," Practical Forensic Analysis of Artifacts on iOS and Android Devices, p. 71–108, 2022.
[11] H. H. Lwin, W. P. Aung, and K. K. Lin, "Comparative analysis of android mobile forensics tools," 2020 IEEE Conference on Computer Applications, ICCA 2020, p. 1–6, 2020.
[12] A. Al-Sabaawi and E. Foo, "A comparison study of android mobile forensics for retrieving files system," Ernest Foo International Journal of Computer Science and Security (IJCSS), no. 13, p. 2019–148, 2019.
[13] W. Asghari, A. Suresh Kumar, A. S. Singh, and K. Thirunavukkarasu, "A comparison analysis of mobile forensic investigation framework," p. 595–602, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-15-6707-0 58
[14] A. Razaque, M. Aloqaily, M. Almiani, Y. Jararweh, and G. Sri vastava, "Efficient and reliable forensics using intelligent edge computing," Future Generation Computer Systems, vol. 118, p. 230–239, May 2021.
[15] A. Akilal and M. T. Kechadi, "An improved forensic-by-design framework for cloud computing with systems engineering standard compliance," Forensic Science International: Digital Investigation, vol. 40, Mar. 2022.
[16] G. Horsman and N. Sunde, "Unboxing the digital forensic inves tigation process," Science and Justice, vol. 62, no. 2, p. 171–180, Mar. 2022.
[17] M. Goel and V. Kumar, Layered Framework for Mobile Forensics Analysis.
[18] A. Al-Dhaqm, S. A. Razak, R. A. Ikuesan, V. R. Kebande, and K. Siddique, "A review of mobile forensic investigation process models," IEEE Access, vol. 8, p. 173359–173375, 2020.
[19] M. A. Saleh, S. Hajar Othman, A. Al-Dhaqm, and M. A. Al Khasawneh, "Common investigation process model for internet of things forensics," 2021 2nd International Conference on Smart

Computing and Electronic Enterprise: Ubiquitous, Adaptive, and Sustainable Computing Solutions for New Normal, ICSCEE 2021, p. 84–89, Jun. 2021.

[20] A. Patil, S. Banerjee, D. Jadhav, and G. Borkar, "Roadmap of digital forensics investigation process with discovery of tools," Cyber Security and Digital Forensics, p. 241–269, Jan. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/ 10.1002/9781119795667.ch11

[21] K. Curran, A. Robinson, S. Peacocke, and S. Cassidy, "Mobile phone forensic analysis," International Journal of Digital Crime and Forensics, vol. 2, no. 3, p. 15–27, 2010.

[22] D. Hamdi, F. Iqbal, T. Baker, and B. Shah, "Multimedia file signature analysis for smartphone forensics," Proceedings - 2016 9th International Conference on Developments in eSystems Engineering, DeSE 2016, p. 130–137, 2017.

[23] M. Al-Hadadi and A. AlShidhani, "Smartphone forensics analysis: A case study," International Journal of Computer and Electrical Engineering, vol. 5, no. 6, p. 576–580, 2013.

[24] A. Shortall and M. A. B. Azhar, "Forensic acquisitions of whatsapp data on popular mobile platforms," Proceedings - 2015 6th International Conference on Emerging Security Technologies, EST 2015, p. 13–17, 2016.

[25] M. Moreb, "Introduction to ios forensics," Practical Forensic Anal ysis of Artifacts on iOS and Android Devices, p. 37–70, 2022.

[26] H. Azhar, . Cox, R., and A. Chamberlain, "Forensic investigations of popular ephemeral messaging applications on android and ios platforms," International Journal on Advances Security, vol. 13, no. 1 & 2, p. 41–53, 2020.

[27] Belkasoft.com, "Belkasoft evidence center," 2021. [Online]. Available: https://belkasoft.com/ru/bec/en/Evidence Center.asp

[28] Finaldata, "Finalmobile forensics," 2021. [Online]. Available: https://finaldata.com/mobile/ [29] 3uTools, "http://www.3u.com/," 2021. [Online]. Available: http: //www.3u.com/

[30] Magnet, "Magnet acquire," 2021. [Online]. Available: https: //www.magnetforensics.com/resources/magnet-acquire/

[31] S. S. Shimmi, G. Dorai, U. Karabiyik, and S. Aggarwal, "Analysis of ios sqlite schema evolution for updating forensic data extrac tion tools," 8th International Symposium on Digital Forensics and Security, ISDFS 2020, 2020.

[32] A. Singh, R. A. Ikuesan, and H. Venter, "Secure storage model for digital forensic readiness," IEEE Access, vol. 10, p. 19469–19480, 2022.

[33] A. Aenurahman Ali, N. Dwi Wahyu Cahyani, and E. Musthofa Jadied, "Digital forensic analysis on idevice: Jailbreak ios 12.1.1 as a case study," Indonesia Journal of Computing, vol. 4, no. 2, p. 205–218, 2019.

[34] A. Turner, "How many people have smartphones worldwide (April 2021)," Jan. 2021. [Online]. Available: https://www.bankmycell. com/blog/how-many-phones-are-in-the-world [35] S. Kemp, "Digital 2021: Global overview report — datareportal – global digital insights," Jan. 2021. [Online]. Available: https://datareportal.com/reports/digital-2021-global-overview-report

[36] A. Hrenak, "Mobile device forensics: An introduc tion," Cyber Forensics, p. 291–322, Sep. 2021. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10. 1201/9781003057888-8/mobile-device-forensics-andrew-hrenak

[37] C. Arumugam and S. Shunmuganathan, "Digital forensics: Essential competencies of cyber-forensics practitioners," p. 843–851, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-15-5243-4 81

[38] A. e. Majdoub, C. Saadi, and H. Chaoui, "Mo bile forensics data acquisition," ITM Web of Con ferences, vol. 46, p. 02006, 2022. [Online]. Avail able: https://www.itm-

[39] L. A. Herrera, "Challenges of acquiring mobile devices while minimizing the loss of usable forensics data," 8th International Symposium on Digital Forensics and Security, ISDFS 2020, Jun. 2020.
[40] A. M. da Costa, A. O. de Sa, and R. C. Machado, "Data acquisition and extraction on mobile devices-a review," 2022 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2022 - Proceedings, p. 294–299, 2022.
[41] M. Kumar, "Mobile forensics: Tools, techniques and approach," Crime Science and Digital Forensics, p. 102–116, Sep. 2021.
[42] D. Kim and S. Lee, "Study of identifying and managing the potential evidence for effective android forensics," Forensic Science Interna tional: Digital Investigation, vol. 33, p. 200897, Jun. 2020. [On line]. Available: https://koreauniv.pure.elsevier.com/en/publications/ study-of-identifying-and-managing-the-potential-evidence-for-effe
[43] M. Lovanshi and P. Bansal, "Comparative study of digital forensic tools," Data, Engineering and Applications, p. 195–204, 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/ 978-981-13-6351-1 15
[44] J. E. Oliveira, T. Silva, A. Zorzo, and C. Neu, "Digital forensics experimentation: Analysis and recommendations." Forensic Science Review, vol. 34, no. 1, p. 21–42, Jan. 2022.
[45] S. Dogan and E. Akbal, "Analysis of mobile phones in digital forensics," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings, p. 1241–1244, Jul. 2017. [46] R. F. Erbacher, "Validation for digital forensics," ITNG2010 - 7th International Conference on Information Technology: New Genera tions, p. 756–761, 2010.
[47] H. Arshad, A. b. Jantan, and O. I. Abiodun, "Digital forensics: Review of issues in scientific validation of digital evidence," Journal of Information Processing Systems, vol. 14, no. 2, p. 346–376, 2018.
[48] H. Arshad, A. Jantan, G. K. Hoon, and I. O. Abiodun, "Formal knowledge model for online social network forensics," Computers and Security, vol. 89, Feb. 2020. [Online]. Available: https://dl.acm.org/doi/10.1016/j.cose.2019.101675
[49] L. F. Sikos, "Ai in digital forensics: Ontology engineering for cy bercrime investigations," Wiley Interdisciplinary Reviews: Forensic Science, vol. 3, no. 3, p. e1394, May 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/wfs2.1394
[50] M. R. Al-Mousa, "Analyzing cyber-attack intention for digital forensics using case-based reasoning," International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 6, p. 3243–3248, Jan. 2021. [Online]. Available: http://arxiv.org/abs/2101.01395
[51] H. Page, G. Horsman, A. Sarna, and J. Foster, "A review of quality procedures in the uk forensic sciences: What can the field of digital forensics learn?" Science & justice: journal of the Forensic Science Society, vol. 59, no. 1, p. 83–92, Jan. 2019. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/30654972/
[52] A. R. Javed, W. Ahmed, M. Alazab, Z. Jalil, K. Kifayat, and T. R. Gadekallu, "A comprehensive survey on computer forensics: State of-the-art, tools, techniques, challenges, and future directions," IEEE Access, vol. 10, p. 11065–11089, 2022.
[53] G. Horsman, "Tool testing and reliability issues in the field of digital forensics," Digital Investigation, vol. 28, p. 163–175, Mar. 2019.
[54] G. Horsman and N. Sunde, "Part 1: The need for peer review in dig ital forensics," Forensic Science International: Digital Investigation, vol. 35, p. 301062, Dec. 2020.
[55] G. Horsman, "Tool testing and reliability issues in the field of digital forensics," Digital

Investigation, vol. 28, p. 163–175, Mar. 2019.
[56] Z. Bartliff, Y. Kim, F. Hopfgartner, and G. Baxter, "Leveraging digital forensics and data exploration to understand the creative work of a filmmaker: A case study of stephen dwoskin's digital archive," Information Processing and Management, vol. 57, no. 6, Nov. 2020.

Dr. Mohammed Moreb is a Vice President of Academic Affairs at Smart University College for Modern Education. He obtained his Ph.D. iIn Electrical and Computer Engineering Dr. Moreb has Expertise in Cybercrimes and Digital Evidence Analysis, specifically focusing on Information and Network Security, with a strong publication track record, many books published by Dr. Moreb specialize in mobile forensics such as Practical Forensic Analysis of Artifacts on iOS and Android Devices: Investigating Complex Mobile Devices 1st ed. published by Apress.

Dr. Saeed Salah is an Assistant Professor and researcher at the Department of Computer Science at Al-Quds University in Jerusalem. He received his BSc. in Electrical/Computer Engineering from Al-Najah National University in 2003, his MSc. degree in Computer Science from Al-Quds University in 2009, and his Ph.D. from the Department of Signal Theory, Telematics and Communications of the University of Granada in 2015. His research interests are focused on network management, information and network security machine learning, data mining, MANETs, routing protocols, and blockchain. Dr. Salah published many peer-reviewed research papers in recognized international journals and conferences. Moreover, he acts as a reviewer for several journals in his field.

Dr. Belal M. Amro is an assistant professor at the College of IT at Hebron University – Palestine. Dr. Belal received his PhD in Computer Science and Engineering from Sabanci University- Istanbul, Turkey in 2012. In 2004 he received his MSc in complexity and its interdisciplinary applications from IUSS, Pavia, Italy. His BSc degree was awarded from Palestine Polytechnic University in computer systems engineering in 2003. Dr. Amro has served as a technical program committee member for different international conferences and journals and reviewed more than 50 papers in the field of information technology including privacy and security. Currently, Mr. Amro is conducting research in network security, wireless security, and privacy preserving data mining techniques and has published more than 22 papers in international journals and conferences in the field of computer security and privacy.

ISSN 2210-142X

Evaluation of Deep Learning Models for Detection of Indonesian Rupiah

Charleen Charleen1 and Gede Putra Kusuma2
1,2Computer Science Department, Bina Nusantara University, Jakarta, Indonesia, 11480

ABSTRACT

This study compares the performance of current object detection models, namely YOLOv7-tiny, YOLOv8n, and EfficientDet d0, using YOLOv5n as the baseline model in addressing the challenge of Rupiah banknote detection. The challenge involves recognizing unique features on the banknotes, which may have higher complexity compared to common objects in object detection tasks. The dataset used covers 2022 Emission Year Rupiah banknotes, is manually created, and covers various real-world scenarios for comprehensive evaluation. This research also explores the impact of data augmentation to optimize model performance. Results show that YOLOv8 is the top-performing model, with mAP@0.5 scoring 0.995 and mAP@0.5:0.95 scoring 0.994 on the test data, also consistently maintaining high performance even without augmentation. YOLOv5 also showed impressive mAP scores of 0.995 and 0.973 with augmentation. YOLOv7, although it did not surpass YOLOv8 and in accuracy, achieved results, especially with data augmentation. In terms of inference time, YOLOv5 excels with 6.7 ms without augmentation and6.5 ms with augmentation, emphasizing its efficiency. YOLOv8, although slightly less efficient, with inference time of 7.8 ms without augmentation and 8.2 ms with augmentation, provides higher accuracy. The choice between the two depends on the balance between accuracy and efficiency. This research also highlights the positive impact of data augmentation, especially in YOLOv5's responsiveness to additional data. While EfficientDet is efficient in inference time and resource usage, it suffers in performance, especially without augmentation. This study attempts to develop a dependable method for identifying banknotes. By achieving this, the aim was to improve accessibility in financial tasks and everyday life, particularly benefiting those with visual impairments or other disabilities.

Keywords: Deep Learning, Object Detection, Indonesian Rupiah, YOLOv5 model, YOLOv7 model, YOLOv8 model, EfficientDet model

1. INTRODUCTION

Object detection is a complex and vital challenge within the field of computer vision. It revolves around the exact localization and recognition of objects in input data. These algorithms work by analyzing the distinctive characteristics of objects and applying machine learning techniques to both characterize and identify these objects [1]. Object detection has a wide range of uses, from enabling autonomous vehicles to assisting in the field of medical imaging [2].

The recognition of Indonesian Rupiah banknotes, however, is a specialized aspect of object detection that is the focus of this study.

The general availability of money in daily financial transactions serves to highlight the importance of this particularized object detection task [3]. This is especially important in the context of Indonesia, where Bank Indonesia issues and governs the country's official currency, the Indonesian Rupiah (IDR). Coins and banknotes are the two primary forms of currency. In this study, we utilized the 2022 issuance of the Indonesian Rupiah, which represents the most recent design iteration. It is important to note that the Indonesian Rupiah's design has undergone multiple revisions over the years, with the primary objective being the enhancement of currency quality and the reinforcement of security features [4].

Disabled individuals often encounter numerous chal lenges in their daily activities. These difficulties span a range of aspects, including mobility, dining, and shopping, with many of these activities involving financial transactions. For physically disabled individuals, maintaining these operations can prove to be significantly more demanding than for those who are able-bodied [5]. However, among the disabled community, visually impaired individuals face even greater obstacles compared to others. Despite the growing popularity of electronic financial transactions, cash remains a primary form of exchange [6]. This presents a unique challenge, as effective strategies are required to enable precise handling and recognition of banknotes, something that can be especially challenging for those with visual impairments. Consequently, this group of people often encounters substantial difficulties when attempting to carry out financial transactions securely and independently [7].

Visual impairments have wide-ranging effects on people's lives, affecting everything from their level of inde pendence overall to their ability to access education and employment opportunities [8]. For this group, being able to identify and distinguish banknotes emerges as a crucial component of financial wellness. While Braille text, tactile features, and currency readers have all offered some support, there is still a critical need for cutting-edge solutions. Here, computer vision technology holds out the possibility of helping those who are blind due to its potential for advancement [9].

Object detection is an important subject that is con stantly researched in computer vision research [10]. Deep learning-based object detection technology has made significant advances in recent decades [11]. These cutting-edge techniques, demonstrated by models such as You Only Look Once (YOLO) [12], have attracted considerable attention due to their real-time detection capabilities and astounding levels of accuracy. On the other hand, EfficientDet was proposed as an object detector model that offers better effi ciency. This research consistently demonstrated superiority in terms of accuracy and efficiency under various resource conditions [13]. This highlights the significant potential of applying these models in the context of Rupiah banknote detection.

This study compares four cutting-edge object detection models: YOLOv5 [14], YOLOv7 [15], YOLOv8 [16], EfficientDet [13] in the context of detecting Indonesian Rupiah banknotes. The aim of this research is to determine the most effective model for developing an application that helps visually impaired people in accurately recognizing and differentiating Indonesian Rupiah banknotes. The study's goal is to identify these models' distinct strengths and their applicability in the context of banknote recognition.

2. RELATED WORKS

Several methods for detecting banknotes have been investigated. Notably, feature-based approaches like Oriented FAST and Rotated BRIEF (ORB) algorithm have shown promise. Sarker and his colleagues proposed ORB as a system for assisting visually impaired individuals in realtime Bangladeshi currency detection [17]. This method demonstrated rapid matching times and achieved a 100% accuracy rate. Their system, which was created as a mobile app, proved to be a valuable tool for visually impaired people, assisting them in precise and real-time banknote identification.

Another noteworthy feature-based algorithm for ban knote detection is Speeded Up Robust Features (SURF). To identify different key points and extract relevant features from the banknote image, SURF employs an arbitrary feature transformation technique. Gillich et al. applied SURF to detect the position and potential occlusion of randomly distributed textured Egyptian banknotes using a smartphone camera [18]. This method employs Random Sample and Consensus (RANSAC) algorithm to filter out false results. In all categories, the accuracy was reported to be 93%.

Another study conducted by Sufri and his team also investigates the development of an automated banknote recognition system to assist visually impaired individuals in identifying Malaysian Ringgit banknotes [19]. It assesses the influence of region and orientation on the performance of feature extraction-based Machine Learning algorithms (K-Nearest Neighbor (KNN), Direct Torque Control (DTC), Support Vector Machine (SVM), Bayesian Classifier (BC)) as well as Deep Learning via AlexNet. SVM and BC achieved 100% accuracy, whereas Deep Learning (AlexNet) performed well with similar orientation but struggled with new orientation. With the goal of enhancing the independence and standard of living for the blind and visually impaired individuals during monetary transactions, this dual approach offers valuable insights for creating strong and flexible banknote identification systems.

Deep learning-based models have brought a transforma tive impact to banknote detection. One successful example is the work conducted by Park and his team [20]. This research suggests a three-stage banknote and coin detection technology using a smartphone camera. To overcome the limitations of earlier approaches, this technique combines Pretrained Faster Region-based Convolutional Neural Net work (R-CNN) with ResNet architecture and geometric con straints. Experiments conducted using Dijkstra's Algorithm with Buckets v1 (DKB v1) and Jordanian dinar (JOD) databases demonstrate greater accuracy when compared to current techniques. The proposed model's accuracy rates for coins, banknotes, and coins and banknotes were 95.48%, 98.8%, and 97.21%, respectively, for DKB v1 and JOD, and they were 92.11%, 97.47%, and 96.04%, respectively, for DKB v1 and JOD, respectively.

YOLOv3 is a prominent deep learning technique for detecting banknotes [11]. Park and his teams [21] present the Multinational Banknote Detecting Model (MBDM). By adding specialized structures including convolution layers, residual layers, and downsampling techniques, MBDM em ploys Improved YOLOv3 in this study. MBDM, with its 69 convolution layers, enhances the operations of feature extraction, prediction, and upsampling, leading to better performance in banknote detection. The alterations are intended to enhance the model's precision in identifying and localize banknotes, particularly by utilizing mosaic data enhancement during the training process. MBDM performed better than current techniques, with 83.96% accuracy. MBDM performs better in the detection of dif ferent currencies thanks to its efficient feature extraction capabilities.

The YOLOv5 algorithm is a promising option for real-time banknote detection applications because of its efficiency and versatility, which have propelled it to the forefront of banknote detection in recent developments. Notably, with an image size of 640 pixels, YOLOv5x significantly obtained an Average Precision (AP) of 50.7% with an image size of 640 pixels when tested on the Microsoft Common Objects in Context (MS COCO) dataset test-dev 2017. Additionally, it can attain a remarkable 200 frames per second (FPS) using a 32-batch batch size on an NVIDIA V100. With test-time augmentation and a larger input size of 1536 pixels, YOLOv5 attains an even higher AP of 55.8% [22]. According to a study by Dande and colleagues, the YOLOv5 model can successfully identify Indian banknotes, as evidenced by the consistently high mean Average Precision (mAP), precision, and recall in banknote detection [23]. These results collectively under score YOLOv5's competence in delivering accurate and efficient banknote recognition.

Another noteworthy advancement is the YOLOv7 [15], as introduced by Wang and his team, presents a novel architecture for real-time object detection and model scaling, offering high accuracy of 56.8% AP, with YOLOv7-E6 delivering outstanding performance at 56 FPS and 55.9% AP. Several architectural adjustments and a set of ”bag-of freebies” were suggested by YOLOv7 to increase accuracy while maintaining the same inference speed and training time [22].

Furthermore, the state-of-the-art YOLOv8 pushes the boundaries of speed and accuracy, outperforming its pre decessor YOLOv5 with an AP of 53.9% on the MS COCO dataset for 640-pixel images. The exceptional processing speed of YOLOv8 distinguishes it. It runs at an impres sive 280 frames per second (FPS) on an NVIDIA A100 with TensorRT. This high FPS indicates that YOLOv8 can process and analyze a large number of frames or images per second, making it ideal for real-time applications and scenarios requiring speed. All of these improvements put YOLOv8 at the forefront of object

Figure 1. 2022 Edition Rupiah Banknotes

TABLE I. Data Information [24]

Nominal	Front Image	Back Image	Size	Color
Rp100.000	Dr. (H.C.) Ir. Soekarno and Dr. (H.C.) Drs. Mohammad Hatta	Topeng Betawi dance, Raja Ampat landscape and Moon Orchid flower	151 mm x 65 mm	Red
Rp50.000	Ir. H. Djuanda Kartawidjaja	Legong dance, Komodo National Park landscape and Bali Jepun flower	146 mm x 65 mm	Blue
Rp20.000	Dr. G. S. S. J. Ratulangi	Gong dance, Derawan landscape and Black Orchid flower	141 mm x 65 mm	Green
Rp10.000	Frans Kaisiepo	Pakarena dance, Wakatobi National Park landscape, and Cempaka Kasar forest	136 mm x 65 mm	Purple
Rp5.000	Dr. K. H. Idham Chalid	Gambyong dance, Mount Bromo, and Sedap malam flower	131 mm x 65 mm	Brown
Rp2.000	Mohammad Hoesni Thamrin	Piring dance, Ngarai Sianok landscape, and Jeumpa flower	126 mm x 65 mm	Grey
Rp1.000	Tjun Meutia	Tifa dance, Banda Neira landscape, and Larat Orchid flower	121 mm x 65 mm	Green

A representative dataset was created using Bank Indone sia (BI) Rupiah banknotes from the 2022 emission year.This dataset contains seven Rupiah banknote denominations issued in the year 2022: Rp.100,000, Rp.50,000, Rp.20,000, Rp.10,000, Rp.5,000, Rp.2,000, and Rp.1,000. We captured separate images for each banknote denomination, encom passing both the front and back classes, resulting in a total of 14 classes. The resolution was set to 640 x 640 pixels for the input size. Table I contains information about the dataset that was used.

The size of each class in the dataset is 75 images per class, with a total of 1050 images of Rupiah banknotes were collected for the 2022 emission year. The dataset is carefully balanced, with each class having an equal proportion of 75 images. This balanced distribution ensures that each banknote denomination is adequately represented in the dataset, facilitating a fair evaluation of the object detection methods under consideration.

The goal of gathering this data is to create a large and representative dataset from which to conduct an accurate evaluation of the object detection method to be tested. With a targeted dataset, this study can facilitate a comprehensive assessment of the effectiveness of the object detection techniques under consideration. By collecting a dataset that includes a variety of conditions, backgrounds, viewing angles, and irregularities as described above, this study hopes to provide accurate and reliable results in evaluating the performance of the compared object detection methods in Rupiah banknote detection.

B. Preprocessing

The preprocessing stage is shown in Figure 2. The preprocessing of the banknote dataset involves everal cru cial steps and is primarily managed using the Roboflow platform. First, the data collection phase includes man

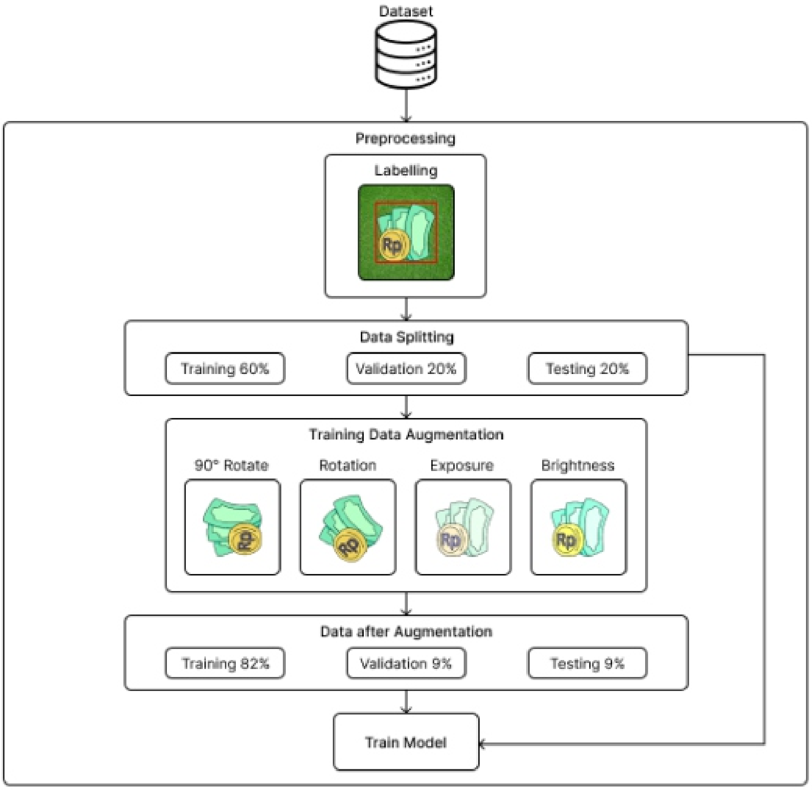


Figure 2. Preprocessing Steps

ual labelling, where each banknote instance in an image is meticulously annotated with a bounding box, and a corresponding label specifying its denomination is added. Following this labelling process, the next step is to adjust the resolution of the images. The original resolution of 3024 x 3024 is modified to 640 x 640. This resolution adjustment is necessary because the chosen YOLO detection model has a maximum resolution limitation of 640 x 640. This change ensures that the dataset complies with the requirements of the YOLO model. After the resolution adjustment is complete, the resolved dataset is ready to proceed to the data augmentation stage.

Data augmentation is an essential step in dataset processing aimed at enhancing the variety and volume of the training data. 3-time augmentation approach is applied, where one image is created with the preprocessing settings applied, and the other two undergo augmentation. This results in tripling the number of images for each source image. The augmentations used involve rotations of 90° rotations in clockwise, counter-clockwise, and upside-down orientations, as well as rotations ranging from -15° to +15°. Additionally, brightness adjustments are applied in the range of -25% to +25%, and exposure variations between -20% and +20% were also applied. The main purpose of this augmentation is to introduce diversity and increase volume of the training dataset. These variations allow the model to learn effectively and handle various situations that may arise during banknote recognition.

Data augmentation is an essential step in dataset processing aimed at enhancing the variety and volume of the training data. 3-time augmentation approach is applied, where one image is created with the preprocessing settings applied, and the other two undergo augmentation. This results in tripling the number of images for each source image. The augmentations used involve rotations of 90° rotations in

clockwise, counter-clockwise, and upside-down orientations, as well as rotations ranging from -15° to +15°. Additionally, brightness adjustments are applied in the range of -25% to +25%, and exposure variations between -20% and +20% were also applied. The main purpose of this augmentation is to introduce diversity and increase the volume of the training dataset. These variations allow the model to learn effectively and handle various situations that may arise during banknote recognition.

The datasets are divided into two distinct sets: one without augmentation and another with augmentation. The dataset without augmentation comprises 1050 images, and it is further split into three subsets: 60% for training (630 images), 20% for validation (210 images), and 20% for testing (210 images). In contrast, the dataset with aug mentation contains 2310 images, with 82% designated for training (1890 images) to benefit from the augmented data, while 9% is allocated to both the validation (210 images) and testing (210 images) subsets. These divisions allow the model to be trained on a diverse range of data examples, while the validation and testing sets remain consistent and unaltered to assess the model's performance under realworld conditions.

C. Model Building

In this study, the performance of object detec tion on banknote datasets will be compared using the YOLOv5, YOLOv7, YOLOv8 and EfficientDet methods. The model used are YOLOv5n, YOLOv7-tiny, YOLOv8n and EfficientDet-d0, selecting these specific versions for their streamlined and lightweight architecture, making them optimal choices for efficient object detection tasks. Each method will be customized and trained using appropriately labelled training data. Here are some details about the model that was used:

1) YOLOv5n

Jocher and his team introduced the YOLOv5n detection algorithm, which distinguishes itself through its simplic ity, speed, and portability [25]. The core structure of the YOLOv5n model comprises three essential components: Backbone, Head, and Output. To extract features, the CSP Darknet serves as the backbone network. Within the CSP Darknet, the architecture leverages both the Focus and CSP structures. The Focus structure executes an image-slicing index operation that transforms spatial dimensions infor mation into channel dimensions. This operation results in a double downsampled feature map, enhancing the model's inference speed. The CSP structure adopts the design prin ciples from the CSPNet network, enabling the model to acquire a richer set of features while addressing the issue of over-computation during inference, an outcome of smart structural design [25]. The Head component is a single head architecture that takes the output of the backbone network and generates predictions. The Output component converts the predictions from the Head component into a format that can be used for the object detection task.

2) YOLOv7-tiny

YOLOv7, is built upon the ELAN (Efficient Layer Aggregation Network) architecture, known for its efficiency and accuracy. YOLOv7-tiny architecture consists of three key components: the backbone, responsible for feature extraction using CSPNet (Cross Stage Partial Network); the neck, which fuses features from different backbone lev els using E-ELAN (Extended Efficient Layer Aggregation Network); and the head, which makes object predictions. YOLOv7 employs novel architecture called E-ELAN, which enhances learning capabilities through group convolution without disrupting gradient flow paths. Furthermore, it em ploys a strategy of using coarse features from the neck for the auxiliary head and fine features for the lead head, which improves accuracy for objects of varying sizes. YOLOv7 improves model stability and accuracy by using both aux iliary and lead heads, making it a significant advancement in real-time object detection [15].

3) YOLOv8n

YOLOv8 is an advanced real-time object detection model that enhances YOLOv5 architecture with additional features. It improves detection accuracy by fusing con textual information with high-level features using the C2f module (cross-stage partial bottleneck with two convolutions). YOLOv8 achieves higher accuracy by processing objectness, classification, and regression tasks independently through the use of a decoupled head and an anchor-free model approach. The model uses the softmax function for class probabilities and the sigmoid function for objectness score activation in the output layer. Moreover, YOLOv8 incorporates sophisticated loss functions for classification loss and bounding box loss, such as DFL and CIoU, which enhance object detection, especially for small objects [22].

YOLOv8n is a lightweight version of YOLOv8 that is optimized for speed and efficiency. YOLOv8n ac quires residual features using an innovative C2f structure that preserves gradient-flow information while ensuring a lightweight design [26]. However, the YOLOv8n archi tecture achieves high accuracy on object detection tasks, making it an excellent choice for applications requiring speed and efficiency, such as real-time object detection on mobile devices.

4) EfficientDet-d0

EfficientDet uses the EfficientNet technique as the basis of its architecture, which is a CNN model designed to improve computational efficiency with respect to limited computing resources. In addition, the EfficientDet architecture also uses the BiFPN technique to combine features from different levels of image resolution and produce richer and more representative features [27]. EfficientDet uses twoheads namely, Classification and Regression head. Clas sification Head is responsible for classifying the detected objects into predefined classes. The Classification Head generates confidence scores for each class of objects to be detected, allowing EfficientDet to recognize objects more accurately. The Regression Head is responsible for generat ing bounding box coordinates for each detected object, thus allowing EfficientDet to recognize objects more precisely [13].

5) Model Implementation

This research refrains from using pre-trained weights or transfer learning due to the following reasons. Pretrained models are tailored to broader datasets like COCO, resulting in inefficient feature extraction for banknotes. Moreover, the absence of pre-trained models for banknote detection limits the advantages of transfer learning [28]. To address these challenges, we opt for a training-from-scratch approach, enabling custom model design specifically tuned to banknotes' unique characteristics, reducing uncertainty, and enhancing efficiency.

The experimentation process takes place on the Google Colab platform, which offers a cloud-based and collabo rative environment for research and development. Google Colab is favored for its accessibility, as it provides a free and powerful computing resource, particularly beneficial for resource-intensive tasks like deep learning [29]. The use of Google Colab eliminates the need for extensive local hardware and eases the setup process. The chosen framework, PyTorch, continues to serve as the foundation for model development and evaluation, as it is fully compatible with Google Colab [29].

Two main experiments were conducted regarding model training. First, model training was conducted without aug mented data, which means that the training data was not increased by adding variance or diversity. Second, model training was conducted with augmented data, where vari ous augmentation operations were applied to the training data to introduce variety and diversity into the dataset. Furthermore, both experiments involve a hyperparameter tuning process. Hyperparameter tuning is the process of optimizing key parameters of the model, such as the number of training epochs (the number of iterations through the entire training dataset) and the learning rate (the rate at which the model learns from the data), to achieve optimal model performance. In the context of this research, the focus of hyperparameter tuning is to fine-tune these parameters so that the model can achieve maximum performance in detecting Indonesian Rupiah banknotes.

D. Evaluation

During the assessment step, several metrics are employed to comprehensively examine the performance of the developed object detection model. The mAP is the fundamental statistic for measuring performance. mAP is a widely used object detection assessment metric. The average precision (AP) throughout all classes is calculated by mAP at a given IoU threshold [30].

The evaluation results are graphed to provide an in depth overview of the model's efficacy. The graph displays mean Average Precision (mAP) scores at different IoU thresholds, with a focus on mAP@0.5 and mAP@0.5:0.95. ”mAP@0.5” denotes the average mAP at an IoU threshold of 0.5, whereas ”mAP@0.5:0.95” denotes the average mAP calculated with a step size of 0.05 across multiple IoU thresholds ranging from 0.5 to 0.95 [31]. This graph serves as a valuable tool for conducting a nuanced analysis of the model's precision-recall trade-off at different IoU levels, allowing for a deeper understanding of its accuracy under varying conditions.

The loss function, with its components ”box,” ”obj” (objectness), and ”cls” (classification), is integral to the training and evaluation of object detection models and is visualized as a loss graph to represent its performance over time. The ”box” component quantifies errors in bounding box localization, ensuring alignment with ground truth [32]. ”Obj” assesses the model's ability to distinguish objects from non-objects, contributing to accurate identification [32]. Meanwhile, ”cls” gauges the model's proficiency in categorizing objects into predefined classes [32]. Examining these loss functions in both training and validation phases through the loss graph offers valuable insights into the model's capacity to localize objects effectively, discern their presence, and achieve precise classification, aiding in model refinement and optimization.

During testing, images from the dataset are processed through the model, generating prediction results that are meticulously compared with the ground truth labels within the dataset. This assessment calculates key performance indicators such as precision and recall at varying thresholds. Moreover, the mAP@0.5 and mAP@0.5:0.95 score for the testing is calculated based on the precision and recall met rics obtained, signifying the model's overall performance. Average inference time per image is a crucial metric that indicates the efficiency of a device in completing image processing tasks. It reflects the speed at which the device can process images, with faster inference times indicating better performance. In the context of analyzing the results of inference time measurements on test data, an evaluation was conducted. These measurements, recorded in milliseconds (ms), encompassed both model implementations without and with the application of data augmentation. The test data comprised 210 images depicting various scenarios. Each processed image was timed from the moment it entered the processing phase until the result was available, providing insights into the efficiency of the models in completing object detection tasks.

4. RESULT AND ANALYSIS

In this section, we present the results and analysis of the experiments conducted using YOLOv5, YOLOv7, YOLOv8 and EfficientDet, with and without data augmentation. The experiments were carried out with specific hyperparameters, including 320 epochs, an image size of 640x640, and a batch size of 16. The evaluation metrics used for analysis include Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95.

A. Training and Validation Results
1) YOLOv5
a) Without Augmentation

The results for YOLOv5 without data augmentation in the validation set are presented in the Table ??. The line graph shown in Figure 3 presents YOLOv5 performance without data augmentation.

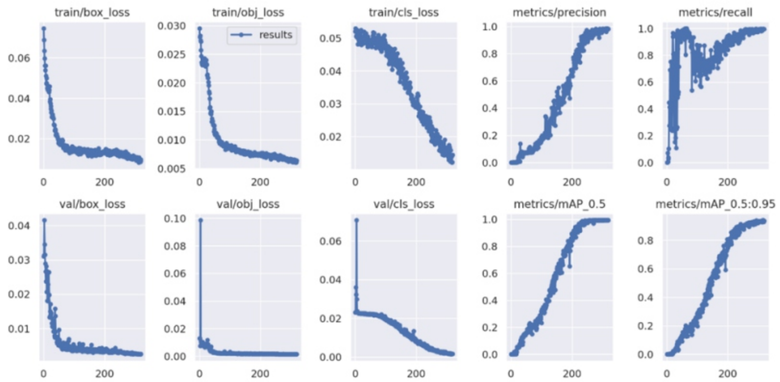


Figure 3. YOLOv5 Results without Data Augmentation

TABLE II. YOLOv5 without Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.983	0.998	0.995	0.941
1000-B	0.998	1.000	0.995	0.946
1000-D	0.989	0.974	0.995	0.934
10000-B	0.998	1.000	0.995	0.965
10000-D	1.000	1.000	0.995	0.968
100000-B	0.974	1.000	0.995	0.941
100000-D	0.986	1.000	0.995	0.904
2000-B	0.963	1.000	0.995	0.956
2000-D	0.987	1.000	0.995	0.958
20000-B	0.985	1.000	0.995	0.956
20000-D	0.991	1.000	0.995	0.942
5000-B	1.000	1.000	0.995	0.938
5000-D	0.978	1.000	0.995	0.945
50000-B	0.941	1.000	0.995	0.919
50000-D	0.969	1.000	0.995	0.908

b) With Augmentation

The results for YOLOv5 with data augmentation in the validation set are presented in Table ??. The line graph shown in Figure 4 presents YOLOv5 performance with data augmentation.

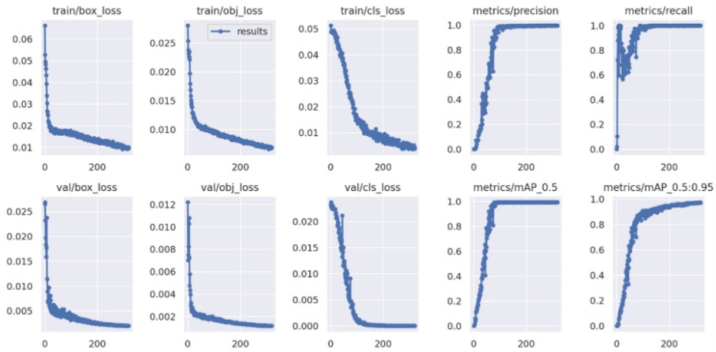


Figure 4. YOLOv5 Results with Data Augmentation

TABLE III. YOLOv5 with Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.996	1.000	0.995	0.975
1000-B	0.997	1.000	0.995	0.960
1000-D	0.997	1.000	0.995	0.969
10000-B	0.996	1.000	0.995	0.962
10000-D	0.998	1.000	0.995	0.981
100000-B	0.994	1.000	0.995	0.970
100000-D	0.995	1.000	0.995	0.970
2000-B	0.995	1.000	0.995	0.980
2000-D	0.996	1.000	0.995	0.957
20000-B	0.993	1.000	0.995	0.989
20000-D	0.993	1.000	0.995	0.984
5000-B	0.996	1.000	0.995	0.984
5000-D	0.993	1.000	0.995	0.977
50000-B	0.996	1.000	0.995	0.981
50000-D	1.000	1.000	0.995	0.977

2) YOLOv7

a) Without Augmentation

The results for YOLOv7 without data augmentation in the validation set are presented in the Table IV. The linegraph shown in Figure 5 presents YOLOv7 performancewithout data augmentation.

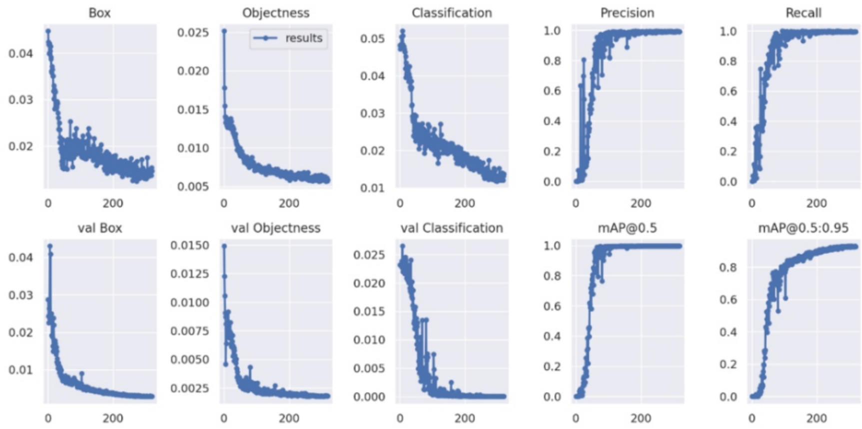


Figure 5. YOLOv7 Results without Data Augmentation

b) With Augmentation

The results for YOLOv7 with data augmentation in the validation set are presented in the Table ??. The line graph shown in Figure 6 presents YOLOv7 performance with data augmentation.

TABLE IV. YOLOv7 without Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.992	0.995	0.997	0.927
1000-B	1.000	0.976	0.997	0.936
1000-D	1.000	0.956	0.997	0.922
10000-B	1.000	0.995	0.996	0.927
10000-D	0.988	1.000	0.996	0.947
100000-B	0.988	1.000	0.996	0.946
100000-D	0.986	1.000	0.997	0.919
2000-B	0.989	1.000	0.997	0.898
2000-D	1.000	1.000	0.997	0.924
20000-B	0.997	1.000	0.998	0.947
20000-D	0.983	1.000	0.997	0.927
5000-B	0.989	1.000	0.997	0.913
5000-D	0.990	1.000	0.997	0.929
50000-B	0.988	1.000	0.996	0.911
50000-D	0.984	1.000	0.997	0.933

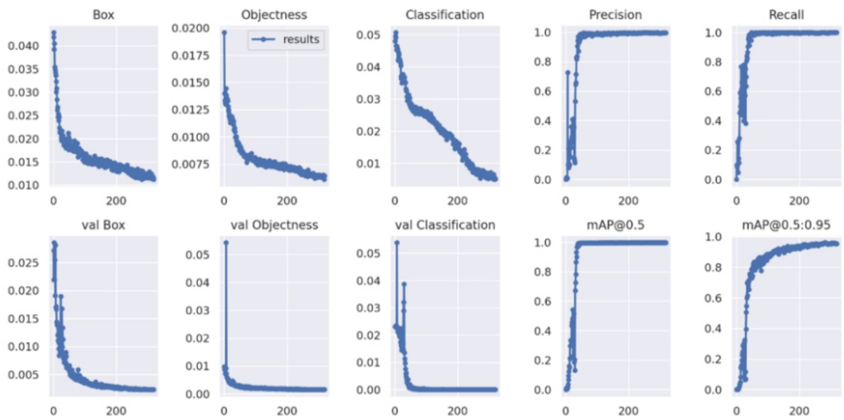


Figure 6. YOLOv7 Results with Data Augmentation

TABLE V. YOLOv7 with Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.995	1.000	0.998	0.954
1000-B	0.998	1.000	0.997	0.938
1000-D	0.994	1.000	0.999	0.956
10000-B	1.000	1.000	0.998	0.951
10000-D	0.996	1.000	0.998	0.964
100000-B	0.994	1.000	0.999	0.956
100000-D	0.993	1.000	0.999	0.967
2000-B	0.996	1.000	0.997	0.960
2000-D	0.994	1.000	0.999	0.959
20000-B	0.994	1.000	0.998	0.978
20000-D	0.994	1.000	0.997	0.921
5000-B	0.998	1.000	0.998	0.967
5000-D	0.993	1.000	0.997	0.952
50000-B	0.995	1.000	0.997	0.946
50000-D	0.996	1.000	0.997	0.943

3) YOLOv8

a) Without Augmentation

The results for YOLOv8 without data augmentation in the validation set are presented in the Table VI. The line graph shown in Figure 7 presents YOLOv8 performance without data augmentation.

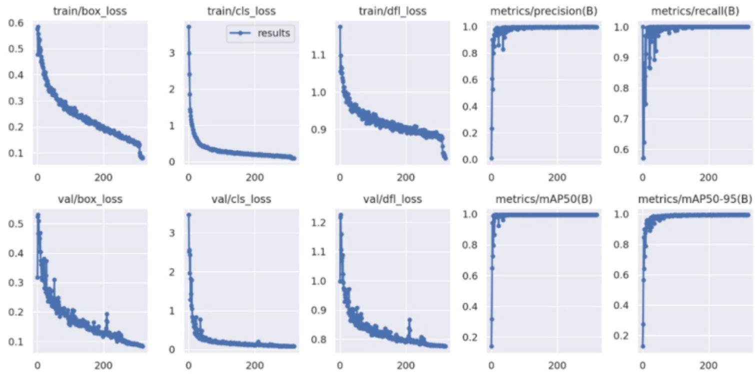


Figure 7. YOLOv8 Results without Data Augmentation

TABLE VI. YOLOv8 without Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.996	1.000	0.995	0.995
1000-B	1.000	1.000	0.995	0.995
1000-D	0.997	1.000	0.995	0.995
10000-B	0.996	1.000	0.995	0.995
10000-D	0.996	1.000	0.995	0.995
100000-B	0.996	1.000	0.995	0.995
100000-D	0.995	1.000	0.995	0.995
2000-B	0.996	1.000	0.995	0.995
2000-D	0.997	1.000	0.995	0.995
20000-B	0.996	1.000	0.995	0.995
20000-D	0.995	1.000	0.995	0.995
5000-B	0.995	1.000	0.995	0.995
5000-D	0.995	1.000	0.995	0.995
50000-B	0.996	1.000	0.995	0.995
50000-D	0.996	1.000	0.995	0.995

b) With Augmentation

The results for YOLOv8 with data augmentation in the validation set are presented in the Table ??. The line graph shown in Figure 8 presents YOLOv8 performance with data augmentation.

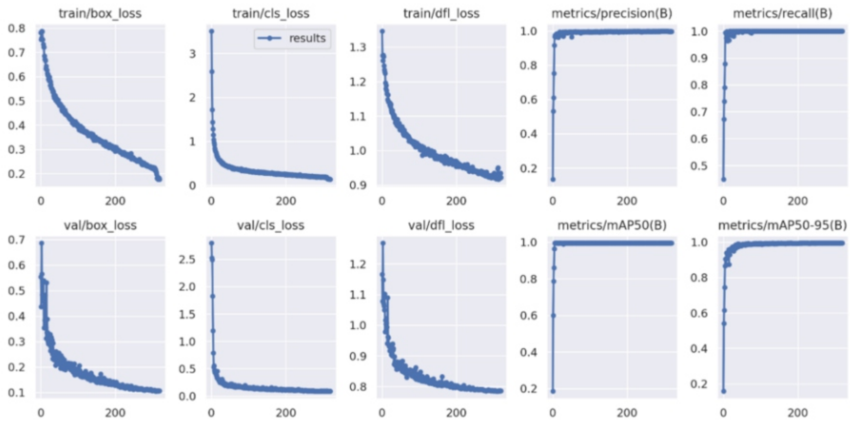


Figure 8. YOLOv8 Results with Data Augmentation

TABLE VII. YOLOv8 with Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.996	1.000	0.995	0.995
1000-B	0.997	1.000	0.995	0.995
1000-D	0.997	1.000	0.995	0.995
10000-B	0.996	1.000	0.995	0.995
10000-D	0.996	1.000	0.995	0.995
100000-B	0.996	1.000	0.995	0.995
100000-D	1.000	1.000	0.995	0.995
2000-B	0.998	1.000	0.995	0.995
2000-D	0.998	1.000	0.995	0.995
20000-B	0.994	1.000	0.995	0.995
20000-D	0.994	1.000	0.995	0.995
5000-B	0.996	1.000	0.995	0.995
5000-D	0.996	1.000	0.995	0.995
50000-B	0.995	1.000	0.995	0.995
50000-D	0.995	1.000	0.995	0.995

4) EfficientDet

The results for EfficientDet without and with data aug mentation in the validation set are presented in Table VIII. The line graph shown in Figure 9 illustrates EfficientDet performance without data augmentation, while Figure 10 presents EfficientDet performance with data augmentation.

TABLE VIII. EfficientDet Validation Results

	mAP@0.5	mAP@0.5:0.95
Without Augmentation	0.245	0.217
With Augmentation	0.784	0.679

B. Testing Results

The testing results for each model, both with and without data augmentation, are summarized in Table IX.

C. Inference Time Results

The average inference time for each model, both with and without data augmentation, are summarized in Table X.

D. Analysis

In this section, we provide a comprehensive analysis of the results, covering the performance of each model, theimpact of data augmentation.

1) Model Performance

YOLOv5 also performs well in both validation and testing scenarios, and its performance benefits signif icantly from data augmentation. When augmentation is applied, YOLOv5 achieves notable improvements in mAP@0.5:0.95, particularly for specific classes. This sug gests that YOLOv5 is adaptive and can benefit from addi tional data during training. It offers a good balance between performance and adaptability, making it a strong choice

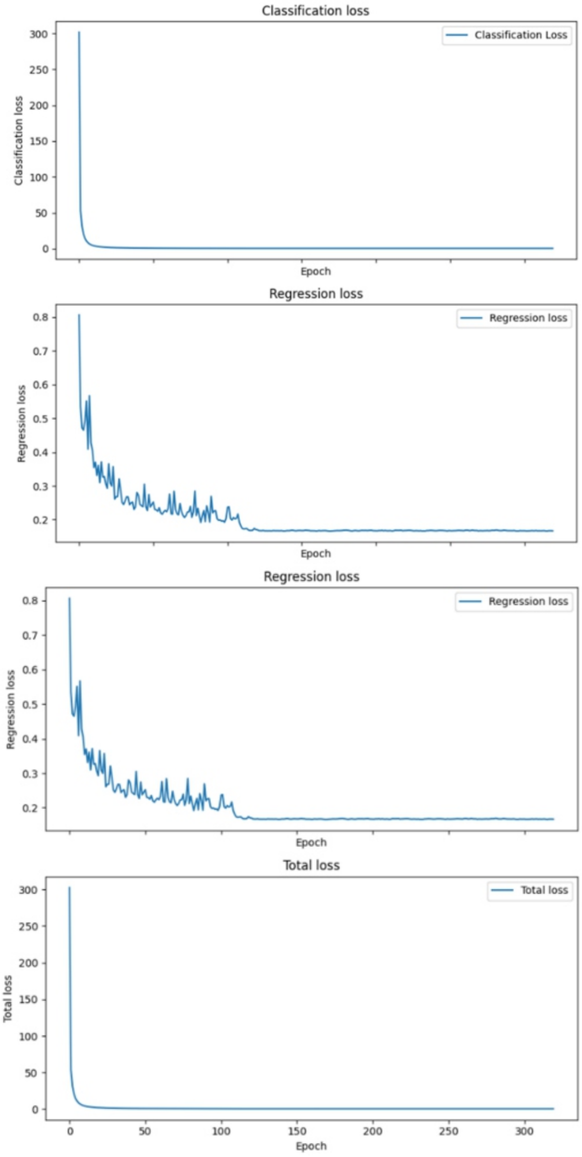


Figure 9. EfficientDet without Data Augmentation

TABLE IX. Testing Results

Model	mAP@0.5	mAP@0.5:0.95
YOLOv5 without augmentation	0.995	0.934
YOLOv5 with augmentation	0.995	0.973
YOLOv7 without augmentation	0.997	0.917
YOLOv7 with augmentation	0.998	0.947
YOLOv8 without augmentation	0.995	0.995
YOLOv8 with augmentation	0.995	0.994
EfficientDet without augmentation	0.239	0.209
EfficientDet with augmentation	0.740	0.632

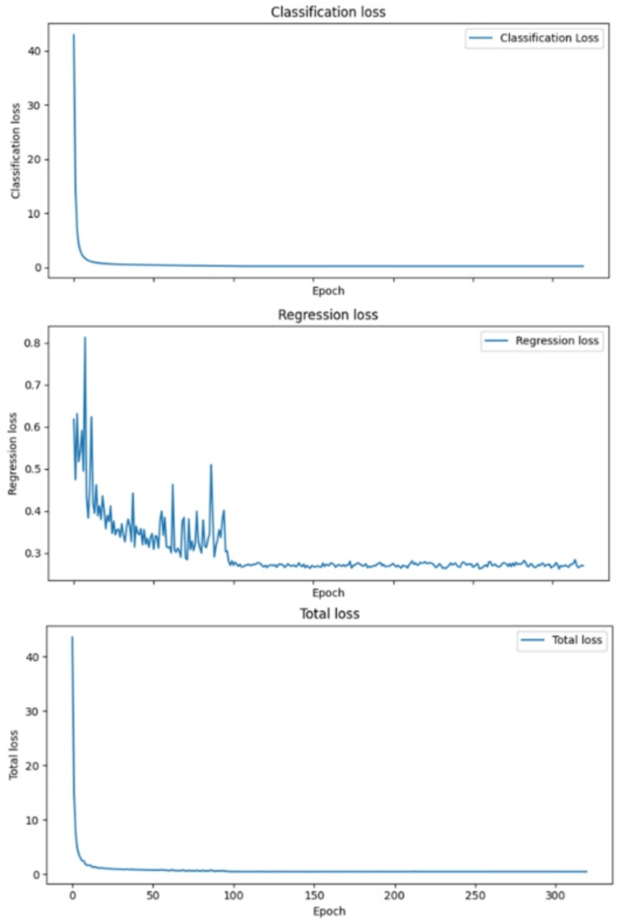


Figure 10. EfficientDet Results with Data Augmentation

TABLE X. Inference Time Results

Model	Inference Time
YOLOv5 without augmentation	6.7 ms
YOLOv5 with augmentation	6.5 ms
YOLOv7 without augmentation	5.8 ms
YOLOv7 with augmentation	6.1 ms
YOLOv8 without augmentation	7.8 ms
YOLOv8 with augmentation	8.2 ms
EfficientDet without augmentation	4.9 ms
EfficientDet with augmentation	4.5 ms

for various applications. While YOLOv5's performance is commendable, it falls slightly behind YOLOv8, which maintains consistent performance without augmentation.

YOLOv7 exhibits strong performance, especially when data augmentation is applied, resulting in higher mAP@0.5 and mAP@0.5:0.95 scores. However, when compared to YOLOv8 and YOLOv5, YOLOv7 falls slightly behind in terms of overall performance. While YOLOv7 offers good results and benefits from data augmentation, it does not outperform YOLOv8 and YOLOv5. Slightly lower mAP scores suggest that, in comparison, YOLOv7 may have limitations in specific scenarios where the highest precision and recall are critical.

YOLOv8 demonstrates consistent and robust performance in both validation and testing scenarios. The model excels without the need for data augmentation, showcasing its inherent strength. This is a significant advantage, as it implies that YOLOv8 can perform reliably in a wide range of real-world applications where augmenting data might not always be feasible or practical. YOLOv8 maintains high precision and recall across various classes, making it a top performer.

Meanwhile, EfficientDet shows a lower level of accu racy both in validation and testing without augmentation. However, with data augmentation, there was a significant improvement. Loss function analysis shows that augmentation has a positive impact on EfficientDet, improving object classification and reducing total loss.

2) Model Efficiency

YOLOv5 exhibited an average inference time of 6.7 ms without data augmentation and 6.5 ms with data augmentation. In comparison, the YOLOv7 model achieved an average inference time of 5.8 ms without data augmentation and 6.1 ms with data augmentation. On the other hand, YOLOv8 demonstrated an average inference time of 7.8 ms without data augmentation and 8.2 ms with data augmentation. With superior inference time efficiency, EfficientDet showed the best performance. It achieved the average in ference time of 4.9 ms without data augmentation, and 4.5 ms with data augmentation. EfficientDet emerged as a highly efficient choice for inference time, with stable performance across the test scenarios. However, the decision in choosing the best model should consider the balance between accuracy and efficiency.

3) Impact of Data Augmentation

Comparing the results with and without data augmenta tion across all models, it's evident that data augmentation significantly impacts the model's performance, particularly in terms of mAP@0.50:0.95. Augmentation enhances the models' ability to generalize and detect objects under vari ous conditions, resulting in higher precision and recall.

The analysis of loss functions reveals the positive impact of augmentation, particularly on YOLOv5 and EfficientDet models, enhancing object classification and reducing total loss. These findings offer valuable insights into the effectiveness and adaptability of each model in the banknote object detection task. Overall, the addition of data augmentation proves beneficial for all models, particularly in scenarios with tighter Intersection over Union (IOU) thresholds.

Specifically, YOLOv5 exhibits a notable increase in mAP@0.5:0.95 when augmentation is applied, showcasing its adaptability to additional training data. Simi larly, YOLOv7 demonstrates significant improvement in mAP@0.5:0.95 with augmentation, indicating the model's responsiveness to data enrichment. Although YOLOv8 maintains high performance even without augmentation, it still benefits from data augmentation, although with less pronounced differences in performance, suggesting its robustness and generalization capabilities. Notably, Effi cientDet shows a dramatic improvement with augmentation, emphasizing its responsiveness to enhanced training data.

These results underscore the importance of augmen tation in enhancing object detection performance across different models. They provide comprehensive insights into the effectiveness and adaptability of each model in object detection tasks, emphasizing the value of data augmentation in improving model performance and adaptability to diverse scenarios.

4) Hyperparameter Tuning

Hyperparameter tuning plays a crucial role in optimizing the performance of deep learning models. Throughout this research, extensive efforts were dedicated to fine-tuning these parameters. It's noteworthy that despite these efforts, the default hyperparameters provided for the models consistently produced the best results. The number of epochs cho sen for training was set to 320, based on empirical evidence indicating that YOLOv5, the baseline model, achieved its highest accuracy at this epoch count. This decision aimed to allow the model to iterate sufficiently through the training data to converge to its optimal performance. Regarding the learning rate, a value of 0.01 was used for all models. Experimentation with different rates, including 0.1 and 0.05 for YOLOv5, did not yield satisfactory results. Higher learning rates may lead to faster convergence but can also cause overshooting or suboptimal solutions. Conversely, lower learning rates promote smoother convergence and better generalization to unseen data. By selecting a learning rate of 0.01, the aim was to strike a balance between con vergence speed and generalization ability, ensuring robust performance across diverse datasets and conditions. Despite the exploration of alternative rates, the default value con sistently demonstrated superior accuracy and generalization capabilities.

5. CONCLUSIONS AND FUTURE WORK

In this study, our primary objective was to evaluate the performance of cutting-edge object detection mod els—YOLOv5, YOLOv7, YOLOv8 and EfficientDet—in the context of Indonesian Rupiah banknote detection. Our findings indicate that YOLOv8n stands out as the top performer in terms of accuracy, consistently deliver ing robust results with remarkable precision and recall. YOLOv8n achieved an outstanding mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.995, both with and without data augmentation, highlighting its exceptional accuracy. YOLOv5n also performed admirably, showcasing adaptability to addi tional training data and substantial improvement with data augmentation. Without augmentation, it achieved mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.941, while with augmentation, it reached mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.975, emphasizing its high precision and recall. YOLOv7- tiny demonstrated strong performance, particularly with data augmentation, achieving a notable mAP@0.5 of 0.997 and mAP@0.5:0.95 of 0.927 without augmentation, and mAP@0.5 of 0.998 and mAP@0.5:0.95 of 0.954 with augmentation, though it slightly trailed YOLOv8n and YOLOv5n in overall performance. However, the Efficient Det model does not match the performance of the other three models, with mAP@0.5 and mAP@0.5:0.95 values of 0.740 and 0.632, even with the application of data augmentation. Data augmentation significantly enhanced model performance, improving their generalization and ob ject detection capabilities, with YOLOv5 and EfficientDet benefiting the most from increased data diversity.

In terms of inference time, EfficientDet stands out as a highly efficient model with the lowest time of 4.9 ms without augmentation and 4.5 ms with augmentation. However, it should be noted that this advantage comes with low accuracy. YOLOv7 also proved to be efficient with an inference time of 5.8 ms without augmentation and 6.1 ms with augmentation, making it the second most efficient model. YOLOv5 shows quite good efficiency with an inference time of about 6.7 ms without augmentation and 6.5 ms with augmentation. On the other hand, YOLOv8, although slightly less efficient, provides adequate perfor mance with an inference time of about 7.8 ms without augmentation and 8.2 ms with augmentation. In choosing the best model, consideration is needed regarding the bal ance between accuracy and efficiency of i ference time. The YOLOv8 and YOLOv5 models can be considered as superior choices, depending on the prioritization between accuracy and efficiency.

Future work includes developing a mobile app with the most effective model for real-time Indonesian Rupiah banknote recognition, prioritizing mobile optimization, user feedback, and usability enhancements. Integration of fea tures such as voice commands and multi-currency support will be explored. Extensive real-world testing with visually impaired users is planned to validate the app's effectiveness in improving their financial independence and daily transaction experiences.

6. LIMITATIONS AND CHALLENGES

The data collection process had limitations that could have influenced the dataset's representativeness and generalizability. Sampling bias was a concern due to limitations in capturing diverse scenarios and differences in image quality from mobile phone cameras. While efforts were made to include a wide range of environmental conditions and lighting scenarios in the dataset, it's possible that some conditions were not adequately represented. Furthermore, focusing solely on 2022 Rupiah banknotes and taking separate images for each denomination may result in im balances. Additionally, ensuring consistency and accuracy in the manual annotation and labeling process posed chal lenges, potentially affecting dataset reliability. Recognizing these challenges is essential to improving the quality of datasets and guaranteeing the validity of research findings.

References

[1] K. Potdar, C. D. Pai, and S. Akolkar, "A convolutional neural network based live object recognition system as blind aid," arXiv preprint arXiv:1811.10399, Nov. 2018.

[2] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," Digital Signal Processing, vol. 126, p. 103514, Jun. 2022.

[3] S. W. Sari, "Perkembangan dan pemikiran uang dari masa ke masa," An-Nisbah: Jurnal Ekonomi Syariah, vol. 3, no. 1, Oct. 2016.

[4] T. Hartono and F. R. Cahaya, "The effectiveness of blind codes and security features for the blind and the prevention of fraud on the indonesian rupiah," Asia Pacific Fraud Journal, vol. 2, no. 2, p. 165, Feb. 2018.

[5] S.-C. Ng, C.-P. Kwok, S.-H. Chung, Y.-Y. Leung, and H.-S. Pang, "An intelligent banknote recognition system by using machine learning with assistive technology for visually impaired people," 2020 10th International Conference on Information Science and Technology (ICIST), Sep. 2020.

[6] J. Lee, H. Hong, K. Kim, and K. Park, "A survey on banknote recognition methods by various sensors," Sensors, vol. 17, no. 2, p. 313, Feb. 2017.

[7] L. P. Sousa, R. M. S. Veras, L. H. S. Vogado, L. S. Britto Neto, R. R. V. Silva, F. H. D. Araujo, and F. N. S. Medeiros, "Banknote identification methodology for visually impaired people," 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Jul. 2020.

[8] N. Jones, H. E. Bartlett, and R. Cooke, "An analysis of the impact of visual impairment on activities of daily living and vision-related quality of life in a visually impaired adult population," British Journal of Visual Impairment, vol. 37, no. 1, p. 50–63, Nov. 2018.

[9] A. Orabi, A. Nasr, and A. Al-Marakeby, "Banknote recognition system for visually impaired people," Journal of Al-Azhar University Engineering Sector, vol. 15, no. 57, p. 1069–1077, Oct. 2020.

[10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, p. 3212–3232, Nov. 2019.

[11] J. Park, J. Baek, K. You, S. W. Nam, and J. Kim, "Microalgae detection using a deep learning object detection algorithm, yolov3," Journal of Korean Society on Water Environment, vol. 37, no. 4, pp. 275–285, 2021.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.

[13] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficientobject detection," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10 781–10 790, 2020.

[14] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon,K. Michael, J. Fang, Z. Yifu, C. Wong, D. Montes et al., "ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation,"Zenodo, 2022.

[15] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainablebag-of-freebies sets new state-

of-the-art for real-time object detectors," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7464–7475, 2023.

[16] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics (version 8.0.0)," Jan. 2023. [Online]. Available: https://github.com/ ultralytics/ultralytics

[17] M. F. Rahman Sarker, M. Israfil Mahmud Raju, A. A. Marouf, R. Hafiz, S. A. Hossain, and M. Hossain Khandker Protik, "Real time bangladeshi currency detection system for visually impaired person," 2019 International Conference on Bangla Speech and Language Processing (ICBSLP), Sep. 2019.

[18] E. Gillich, A. Fritze, K.-F. Henning, A. Pfeifer, H. Dorksen, and V. Lohweg, "Camera-based occlusion detection for banknote sorting," 2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI), Sep. 2019.

[19] N. A. J. Sufri, N. A. Rahmad, N. F. Ghazali, N. Shahar, and M. A. As'ari, "Vision based system for banknote recognition using different machine learning and deep learning approach," 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC), Aug. 2019.

[20] C. Park, S. W. Cho, N. R. Baek, J. Choi, and K. R. Park, "Deep feature-based three-stage detection of banknotes and coins for assisting visually impaired people," IEEE Access, vol. 8, p. 184598–184613, 2020.

[21] C. Park and K. R. Park, "Mbdm: Multinational banknote detecting model for assisting visually impaired people," Mathematics, vol. 11, no. 6, p. 1392, 2023.

[22] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," arXiv preprint arXiv:2304.00501, Apr. 2023.

[23] S. A. Dande, G. R. Uppunuri, and A. S. Raghuvanshi, "Yolov5 based web application for indian currency note detection," Int. Res. J. Eng. Technol, pp. 2801–2807, 2022.

[24] D. Sihwinarti and I. D. M. Sutedja, "Wacana redenominasi kaitanya uang rupiah baru di indonesia," SENADA (Seminar Nasional Manajemen, Desain dan Aplikasi Bisnis Teknologi), vol. 6, pp. 304–309, 2023. [Online]. Available: https://eprosiding.idbbali.ac.id/ index.php/senada/article/view/769

[25] Y. Jin, Y. Lu, G. Zhou, Q. Liu, and Y. Wang, "Glass wool defect detection using an improved yolov5," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),Jun. 2023.

[26] Y. Du, X. Liu, Y. Yi, and K. Wei, "Optimizing road safety: Advancements in lightweight yolov8 models and ghostc2f design for real-time distracted driving detection," Sensors, vol. 23, no. 21, p. 8844, Oct. 2023.

[27] M. L. Mekhalfi, C. Nicolo, Y. Bazi, M. M. Al Rahhal, N. A. ` Alsharif, and E. Al Maghayreh, "Contrasting yolov5, transformer,and efficientdet detectors for crop circle detection in desert," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2021.

[28] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with yolo network," Computers and Electronics in Agriculture, vol. 169, p. 105214, Feb. 2020.

[29] S. Ray, K. Alshouiliy, and D. P. Agrawal, "Dimensionality reduction for human activity recognition using google colab," Information, vol. 12, no. 1, p. 6, Dec. 2020.

[30] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," arXiv preprint arXiv:2305.09972, 2023.

[31] C.-y. SONG, F. ZHANG, J.-s. LI, J.-y. XIE, C. YANG, H. ZHOU, and J.-x. ZHANG, "Detection of maize tassels for uav remote sens ing image with an improved yolox model," Journal of Integrative Agriculture, vol. 22, no. 6, p. 1671–1683, Jun. 2023.

[32] X. Zhang, Y. Feng, S. Zhang, N. Wang, and S. Mei, "Finding nonrigid tiny person with densely

cropped and local attention object detector networks in low-altitude aerial images," IEEE Journal ofSelected Topics in Applied Earth Observations and Remote Sensing, vol. 15, p. 4371–4385, 2022.

Charleen is a dedicated college student in the Bina Nusantara University Graduate Program (BGP). Pursuing a degree in Computer Science, Charleen has actively contributed to this paper through writing this paper. showcasing a strong foundation in computer vision. With a passion for research and a well-rounded approach to education, Charleen aspires to make significant future contributions to the field of computer science. She is currently in her 9th semester of her college year at Bina Nusantara University, Jakarta 11480, Indonesia. She can be contacted at her email: charleen@binus.ac.id

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Head of Department of Master of Computer Science, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R – A*STAR, Singapore. His research interests include computer vision, deep learning, face recognition, appearance-based object recognition, gamification of learning, and indoor positioning system.

ISSN 2210-142X

Advancing Context-Aware Recommender Systems: A Deep Context-Based Factorization Machines Approach

Rabie MADANI1, Abderrahmane EZ-ZAHOUT1, Fouzia OMARY1 and Abdelhaq CHEDMI1
1Intelligent Processing and Security of Systems Team, Computer Sciences Department Faculty of Sciences, Mohammed V University

ABSTRACT

Context-aware recommender systems (CARS) aim to offer personalized recommendations by incorporating user contextual information through analysis. By analyzing these contextual cues, CARS can better understand the preferences and needs of users in different situations, thereby improving the relevance and effectiveness of the recommendations they provide. However, integrating contextual information into a recommendation system presents challenges due to the potential increase in the sparsity and dimensionality. Recent studies have demonstrated that representing user context as a latent vector can effectively address these kinds of issues. In fact, models such as Factorization Machines (FMs) have been widely used due to their effectiveness and their ability to tackle sparsity and to reduce feature space into a condensed latent space. Despite these advantages, FMs encounter limitations when dealing with higher-order feature interactions, since the model's design, primarily focused on second-order interactions. Furthermore, a significant drawback of FMs is their inability to distinguish between different contexts effectively. By utilizing a uniform latent space to model interactions across all features, FMs overlook the nuanced differences that distinct contexts bring to the interactions. This article introduces a CARS model called Deep Context-Based Factorization Machines (DeepCBFM). The DeepCBFM combines the power of deep learning with an extended version of Factorization Machines (FMs) to model non-linear feature interactions among user, item, and contextual dimensions. Additionally, it addresses specific shortcomings of FMs with the goal of enhancing recommendation accuracy. We implemented our method using two datasets that incorporate contextual information, each having distinct context dimensions. Experimental findings demonstrate that the DeepCBFM model surpasses baseline models, thereby validating its efficacy.

Keywords: Recommender systems, Context Aware Recommender Systems, Factorization Machines, Context-Based Factorization Machines, Deep Learning, DNNs

1. INTRODUCTION

In light of the vast and ever-expanding array of products and services available today, Recommender Systems (RS) have become indispensable across a range of sectors, includ ing ecommerce, video content, cinema, travel, and, notably, the gaming industry. RS belong to the category of data fil tering systems, and their primary function is to provide user specific recommendations based on individual preferences [1]. The adoption of RSs offers two significant advantages: firstly, they streamline the user experience by minimizing the time and effort required to find desired items, and sec ondly, they contribute to increased company sales and rev enue generation. Classical Recommender Systems primarily rely on just two key dimensions: the user dimension and the item dimension, to offer personalized recommendations. In the content-based approach [2], a more detailed strategy is employed. This method incorporates both item features and user profiles in the recommendation process. By analyzing the characteristics of items and the preferences and behavior of users, content-based systems provide recommendations that are tailored to individual users based on the content.

Conversely, Collaborative filtering approaches [3] take a different route. They don't require detailed information about users or items. Instead, they depend on user ratings to assess preferences for specific items. However, these approaches have shortcomings because they overlook various factors that can impact a user's preferences, such as the contextual factor. For example, a father's movie preference may shift based on whether he's watching alone or with his kids, with context (like the presence of companions) influencing his choices.

Recommendation systems often face two primary challenges: sparsity and high dimensionality in the data. Sparsity arises when there are limited interactions or ratings between users and items, while high dimensionality results from the addition of contextual information, which effectively increases the number of dimensions in the dataset. Paradoxically, incorporating contextual data can exacerbate the sparsity issue by introducing more dimensions to the data, further complicating the recommendation process. Sparsity refers to the situation where there are significant missing data points in a dataset, while high dimensionality results from the addition of new features, which effectively increases the number of dimensions.

Factorization Machines FM [4] which is a supervised algorithm effectively resolves aforementioned problems and delivers impressive performance. This method involves the transformation of user-item dimensions into latent spaces, creating a low-dimensional representation of the data. The majority of research within the realm of CARSs [5] has been dedicated to improving and advancing Factorization Machines. One of the primary drawbacks of FM is its reliance on a fixed interaction function, often an inner product, to estimate high-order interactions between User and Item. This fixed function may not adequately capture the complexity and nuances of real-world user-item relationships. High-order interactions refer to interactions among three or more features. For example, in a movie recommendation system, a second-order interaction might consider how the genre and the director of a movie affect its rating. A high-order interaction could look at how the genre, director, user age and time of day together influence the rating. These interactions can provide more nuanced insights and improve the predictive performance of the model. However, capturing and utilizing these higherorder interactions effectively is challenging, primarily due to computational complexity, data sparsity, and the potential for model overfitting. The computational cost of explicitly modeling high-order interactions increases exponentially with the order of interaction. For an interaction of order n, a naive approach would require considering all n-way combinations of features, which becomes computationally infeasible for large n. This complexity is further compounded by the sparsity of datasets, where occurrences of higher-order feature combinations are rare, complicating the learning process. Moreover, as the model complexity rises with the inclusion of higher-order interactions, the risk of overfitting intensifies, particularly in scenarios where the available dataset is insufficient to train the model on a multitude of parameters, undermining the model's generalizability and effectiveness. All this can result in less accurate recommendations compared to more advanced techniques that can better model the complexities of user item interactions. Moreover, FM employs a single latent vector even when dealing with features originating from distinct contextual dimensions or contexts. This can be a limitation of FM, since it tends to overlook the fact that features may exhibit distinct behavior when interacting with features from separate contexts. This lack of context awareness can limit FM's ability to capture the nuanced interactions that occur between features across different contexts. The main contributions proposed by the present work are:

- New Context-Aware Recommender Model: The main contribution is the introduction of a new context-aware recommender model. This model combines Deep Neural Networks (DNNs) and Factorization Machines (Fms). This hybrid approach likely aims to capitalize on the strengths of both DNNs and FMs to enhance CARS performance.

This approach considers additional information, like time, companion, location, etc., in order to provide personalized recommendations.

- Exploiting Deep Learning Techniques: The second con tribution is the utilization of DNN to capture non-linear feature interactions. This is important because deep learning models are known for their capability to capture intricate patterns in data. By doing so, the proposed model can address some of the limitations associated with traditional Factorization Machines, which may struggle with modeling higher-order interactions.
- New Variant of FM for CARSs: The final contribution is the introduction of a new variant of Factorization Ma chines specifically adapted to Context Aware Recommender Systems (CARSs). This variant is designed to capture the differences between different contexts and also to capture low order feature interactions.

The remainder of this paper is organized as follows: Section 2 provides a review of related works. Section 3 presents the proposed model (Deep Context-Based Factorization Machines model). Section 4 analyses results. Finally, the last section presents a conclusion of the realized work.

2. RELATED WORKS

Recently, many studies have been presented to enhance the exactness and the efficiency of recommenders, either by improving FM, which is a reference algorithm, by exploiting the strengths of deep learning or by using new methods.

Cheng et al. [6] presented a Wide and deep learning model, that uses the wide linear model to memorize the interaction of features and deep learning DNNs for feature generalization. The model was evaluated using PlayStore, the outcomes demonstrate that the model increased the acquisitions of apps.

Guo et al. [7] presented a Deep FM algorithm, which merges the power of FMs and DNNs to improve recommen dation performances with less manual feature engineering work. Both components of the model were trained jointly in order to gain in terms of performance and also to capture high order interactions.

Xiao et al. [8] presented an Attentional FM model, which captures the strengths of interactions between features using neural attention networks. The model tries to enhance FM by improving the interpretability and the representation ability of a FM algorithm.

Lian et al. [9] introduced the Compressed Interaction Net work(CIN) and integrated it with DNN to develop a unified approach. This approach strives to autonomously acquire the interaction between features in a direct manner, thereby circumventing the need for feature engineering.

Song et al. [10] presented a CTR model using a selfattentive neural network called AutoInt. It learns auto matically high order by allowing the interaction between features for relevance determination.

Yu et al. [11] presented a Input aware Factorization Machine to enhance FMs by considering the inputs influence on feature representations. The model uses neural networks to learn the input factors of each features in several instances. The model endeavors to augment predictive capability while preserving the linear complexity of the traditional FM.

Pan et al. [12] presented Field-weighted FM for recommendation in display advertising field. The model is an extension of FM that aims to capture the importance of interaction of different pairs of fields in reasonable complexity time.

Trigeorgis et al. [13] presented a deep MF to learn attribute representations. The model uses a semi Non-Negative Matrix Factorization algorithm to model representations of low dimensions.

Lara-Cabrera et al. [14] presented a collaborative filtering Recommender System that exploits DNNs and FMs to enhance recommendations precision. The method uses a deep learning paradigm not to capture high order features but to improve the MF model.

Ez-Zahout et al. [15] presented a hybrid movie Recommender System based on Matrix Factorization and KNN.

The model provides recommendation to a user by calculat ing movies similarity and generating top k movies.

While prior research has demonstrated strong results in terms of accuracy, performance, and interpretability in recommendation systems, they have generally overlooked the significant influence of contextual factors on user behaviors. In response to this limitation, other researchers have made efforts to incorporate contextual data into the recommenda tion process with the aim of achieving improved outcomes, based on Matrix Factorization and Factorization Machines. For instance, Baltrunas et al. [16] proposed a CARS model based on Matrix Factorization. This model takes into ac count the interplay between contexts and ratings given to items by using extra additional parameters. One key benefit of this solution is its lower computational overhead, and it also offers the flexibility to depict the interaction between context and items at various levels of detail or granularity. In their work, Madani and Ez-zahout [17] proposed a CARS model that utilizes a BERT model for personalized NER. The model enables the automatic extraction of contextual information. Furthermore, the authors adapted traditional Factorization Machines to accommodate contextual infor mation, enhancing their capability to provide accurate rating predictions.

Casillo et al. [18], presented a CARS that leverages em bedded context likely involves integrating contextual data directly in recommendations to enhance the system's performance. In this work, instead of treating contextual infor mation as an external or additional input to the recommen dation model, the system incorporates this context within the model itself. Moreover, the method leverages matrix factorization's computational efficiency to address scala bility issues. While Matrix Factorization and Factorization Machines offer advantages, they encounter challenges when it comes to capturing the intricacies of non-linear feature interactions in the learning process. Numerous researchers have endeavored to harness the capabilities of deep learning in crafting more intricate context-aware recommender mod els, aiming to address the complexities posed by non-linear problems in recommendation systems.

Jeong et al. [19] introduced a context-aware recommender system leveraging deep learning techniques. This approach considers contextual features to enhance recommendation accuracy. The model integrates a neural network and au toencoder, leveraging established deep learning architec tures. Through this combination, the model effectively extracts distinctive features and forecasts scores during the input data restoration process. Notably, the proposed model exhibits versatility in accommodating diverse contextual information types.

Sattar and Bacciu [20] introduced a Context-Aware Graph Convolutional Matrix Completion method. This approach encompasses a comprehensive understanding of graph struc tures by integrating user preferences, contextual cues conveyed through edges. Through a graph encoder mechanism, it generates nuanced representations of users and items by taking into account contextual cues, inherent features, and user opinions. These representations are aggregated and fed into the decoder, which predicts ratings.

VU et al. [21] introduced an innovative strategy that lever ages deep learning for the development of context aware multi criteria recommender systems. In their approach, DNN models play a pivotal role in predicting context aware multi criteria ratings and learning the aggregation function. The study effectively demonstrates the incorporation of contextual information into Multi Criteria Recommender Systems (MCRSs) through the application of DNN models. Specifically, they showcased the utility of DNN models in forecasting context aware multi criteria ratings and acquir ing insights into the aggregation function.

Vaghari et al. [22]proposed a novel context-aware frame work aimed at enhancing the precision of machine learning models in the identification of disease patterns. Their approach, characterized by the integration of multi-modal data sources, seeks to harness the synergistic effects of combining genetic, clinical, and environmental information.This comprehensive model not only improves diagnosticaccuracy but also tailors therapeutic interventions to in dividual patient profiles, thereby advancing personalized medicine. However, the application of this model faces certain limitations, including the high computational cost associated with processing large multimodal datasets and the potential for biases inherent in incomplete or unrepre sentative data. Addressing these challenges is crucial for ensuring the scalability and reliability of the framework in diverse clinical settings.

Vecchia et al. [23] proposed a groundbreaking model that leverages advanced computational techniques to enhance the accuracy of climate prediction models. Their approach, based on integrating satellite data with terrestrial observation networks, aims to fill critical gaps in meteorological data availability. This integration allows for a more com prehensive understanding of climatic patterns, facilitating better-informed decisions in climate-sensitive sectors such as agriculture and water resource management. However, the model also faces several limitations. One significant challenge is the potential for discrepancies and errors in satellite data, which can affect overall model accuracy. Additionally, the integration of heterogeneous data sources raises concerns about data compatibility and processing complexities. These challenges highlight the need for ongoing refinement of data integration techniques and the development of more robust error-correction algorithms to ensure the reliability and precision of climate forecasts.

Dilekh et al. [24] proposed a context-aware personalized recommendation system for smart homes, demonstrating the application of the FP-Growth algorithm and a Generalized Linear Model (GLM) to derive and utilize association rules for user behavior. Their study, which focused on a single user scenario, highlighted the system's high accuracy and favorable precision/recall metrics, underscoring its effective ness. Despite its achievements, the research acknowledged limitations, particularly its application within a single-user context, which may not fully capture the complexities of multi-user environments. Furthermore, the adaptability of the system to evolving user behaviors and the balance between personalization and privacy were not extensively tested. The integration of the system into broader smart home ecosystems and its interaction with various devices and platforms also remain areas for future exploration.

Previous research heavily relied on deep learning as the primary method for generating recommendations but failed to adequately address the challenges of sparsity and high dimensionality in real-world datasets. Unlike prior studies, our objective is to introduce a pioneering context-aware recommender system that specifically tackles the shortcom ings of Factorization Machines (FMs) by harnessing the capabilities of Deep Neural Networks. Additionally, our aim is to unveil an evolved FM variant optimized for CARS. This refined FM version excels in capturing intricate feature interactions, spanning across both low and high orders. More detailed insights into the DeepCBFM model will be elaborated upon in the subsequent section.

3. PROPOSED MODEL

Our main goal is to develop an advanced recommender system that not only utilizes traditional user and item data but also integrates contextual information to gain a deeper understanding of user behaviors across diverse scenarios. This holistic approach enables us to deliver personalized recommendations tailored to individual user preferences and needs. In pursuit of this goal, we introduce the Deep Context-Based Factorization Machines Model (DeepCBFM), a fusion of Factorization Machines(FM) and deep learning techniques.

Illustrated in figure 1 is the model architecture comprising two simultaneous components: the primary

CBFM component and the secondary deep component. The CBFM element expands upon Factorization Machines (FM) to capture second-order interactions, meanwhile the deep element

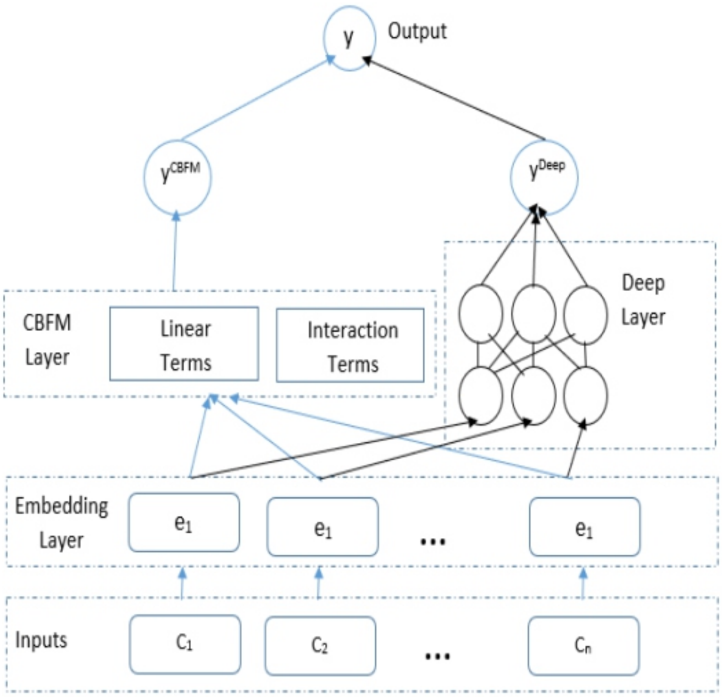


Figure 1. The architecture of the DeepCBFM

focuses on modeling higher-order feature interactions. Both elements utilize common input and embedding layers, and the model's ultimate prediction results from the combined outputs of these two components. This section provides a comprehensive breakdown of each component within the proposed model.

A. CBFM component

In previous works, Linear regression (LR) model has been widely employed in ratings prediction [25][26]. To predict ratings LR uses a linear combination of features as shown in the equation below:

$$y_{LR} = w_0 + \sum_{i=1}^{d} w_i x_i \tag{1}$$

However, the LR model does not perform well since it does not consider the interaction between features, which is crucial. Poly2 models [27] tackle this problem by adding order-2 feature interactions to the above equation, which gives as a result:

$$y_{Poly2} = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} x_i x_j w_{h(i,j)} \tag{2}$$

However, it is clear that this method suffers from some drawbacks. For instance, the interaction parameters of features can be trained only when these features appear in the same record, which means that unseen features will have insignificant predictions. The FM model outperforms Poly2 especially when the model deals with sparse data.

FM calculates interactions between two features via the dot product of their corresponding latent vectors. FM equation is stated as:

$$y_{FM}(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} < v_i, v_j > x_i x_j \quad (3)$$

FM can train embedding vector vi(vj) even if it never or rarely appeared in the data. To learn the effect of latent between features, FM uses only one latent vector even for features from different contexts. For instance, when computing interactions among three contexts (Day, Companion, and Mood), FM utilizes the same embed ding vector for Monday to capture its latent effects when paired with Companion < *vMonday, vS on* > and also with Mood < *vMonday, vHappy* >, despite these contexts being distinct. This approach overlooks the nuanced behavior of features when they interact across different contexts. To verify this observation, we utilize ANOVA, a statistical method developed by Ronald Fisher in the early 20th century. ANOVA is employed to detect and showcase potential similarities or differences in specific aspects within a studied population through variance analysis. The formula of ANOVA is defined as follow:

$$F = \frac{Meansumofsquaresduetotreatment}{Meansumofsquaresduetoerror} \quad (4)$$

We do not use ANOVA method to verify the existence of relationship between two features, but simply to capture the difference of features interaction strength. We conduct this experiment on DePaulMovie dataset [28]. Figure 2 shows the obtained results from the interaction of three contexts namely Time, Location and Companion using ANOVA twoway. This statistical tool helps us to compute the interaction strength between two contexts (Ci , Cj) and an outcome Y.

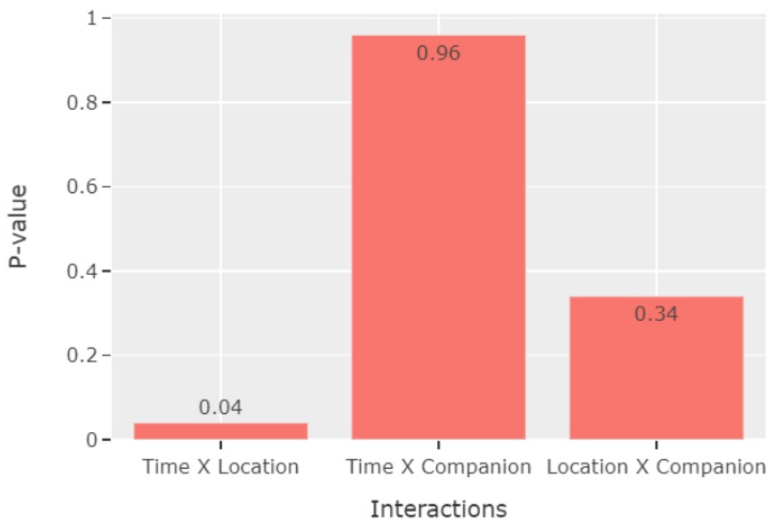


Figure 2. Interaction between contexts

As we can see, the interaction between features is expressed in P value, when the P value is under 0,05 it means that the interaction is significant, otherwise, there is no significant interaction between features. For the first interaction between Time and Location the P value achieves 0.04 which means inferior to 0.05 so Therefore, the interaction effect between these two contexts is considered significant. For the second interaction (Time, Companion) and the third interaction (Location, Companion), the obtained values are higher than 0.05 which means that, there is no significant interaction between these two pairs of contexts. The primary conclusion drawn from this experiment is that variables exhibit varying interaction patterns when paired with variables from distinct contexts. Addressing the limitations of FM highlighted earlier, we introduce an enhanced version termed CBFM. This extension of FM incorporates extra weights to discern variations between contexts and to dis tinguish the latent vectors of a feature when it interacts with other features from different contexts. Within our model, the primary goal of the CBFM component is to capture second order feature interactions while accommodating contextual differences. The equation of CBFM is stated as:

$$y_{CBFM}(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} < v_i w_{c(i)}, v_j w_{c(j)} > x_i x_j \tag{5}$$

Where $w0$ and w are respectively the bais term and weights of feature vectors, *vi, vj* the latent vectors of *f eaturei* , *f eaturej* , *wc(i) , wc(j)* ∈ **R** are weights to capture the importance between conte *xt(i)* and conte *xt(j)*.

B. Deep component

To model nonlinear feature interactions, we present a feed forward neural network [7]. In the initial phase of the deep component, sparse input features are converted into dense vectors through an embedding process. The original input vector is often characterized by high sparsity and dimensionality, encompassing a variety of data types (both categorical and continuous), and organized by contextual groupings like time, weather, and more. This scenario necessitates the use of an embedding layer to condense the input vector into a compact, dense form, facilitating its introduction to the first hidden layer for further processing. It's important to highlight that FM and deep components utilize shared feature embeddings due to two primary considerations: Firstly, by using the same embeddings, both the FM and deep components operate on a unified representa tion of the input features. This ensures consistency in how features are interpreted across different parts of the model, facilitating a more cohesive learning process. Embeddings capture the essence of each feature in a lower-dimensional space, preserving semantic relationships between features.

When these embeddings are shared, it ensures that both the FM component, which excels at capturing second order interactions, and the deep component, which models complex, non-linear relationships, base their computations on the same foundational understanding of the data.

Secondly, sharing embeddings between components simpli fies the overall architecture of the model. Instead of having separate mechanisms for translating raw features into a format suitable for each component, the model centralizes this function in the embedding layer. This not only reduces the number of parameters that the model needs to learn, saving computational resources but also streamlines the training process, as there is a single source of truth for feature representations. These embedding vectors are then concatenated to form a single, dense vector s (0), which serves as the input to the neural network. This vector effectively represents your original input data in a dense, lower-dimensional space, preserving important information while reducing dimensionality.

$$s^{(0)} = [e_1, e_2, e_3, ..., e_l] \tag{6}$$

l is the contexts' number, *ei* embedding feature of *i th* feature. The concatenated embeddings are fed into a feed-forward neural network. This network consists of multiple layers, each designed to capture increasingly complex patterns and interactions among the input features.

$$s^{(j)} = f(W^{(j)} s^{(j-1)} + b^{(j)}) \qquad (7)$$

f is the activation function, *j* is layer depth, *W(j)* the weight of jth layer, *b(j)* the bias at the *jth* layer and *s(j)* the *jth* layer output. let *y (DNN)* be the deep layer output, the final prediction of DeepCBFM model is:

$$y = \sigma(y^{(CBFM)} + y^{(DNN)}) \qquad (8)$$

Where σ is the sigmoid function.

4. Results and discussion

A. Datasets

To evaluate the DeepCBFM, two datasets provided by CARSKit [28] are used: The first is the DePaulMovie dataset encompasses a total of 5044 movie ratings, of which 1449 are non-contextual, and 3595 are contextual, based on specific viewing circumstances. This dataset distinguishes between three primary contextual dimensions: Location (with options of Home or Cinema), Companion (choices include Alone, Family, or Friends), and Time (categorized into Weekend or Weekday). These contextual factors enable a nuanced analysis of user preferences, tailored to specific viewing environments and social settings, thereby provid ing insights into how different contexts influence movie watching experiences. The second is the InCarMusic dataset comprises 4012 ratings collected from 43 users across 138 distinct music items, with 1004 ratings categorized as non-contextual and 3010 as contextual, reflecting the user's environment and situation during music listening. It features seven diverse contextual dimensions: Driving style (Relaxed or Sport driving), Landscape (Coastline, Countryside, Mountains, or Urban), Mood (Active, Happy, Lazy, or Sad), Natural phenomena (Afternoon, Daytime, Morning, or Night), Road type (City, Highway, or Serpen tine), Sleepiness (Awake or Sleepy), and Weather (Cloudy, Rainy, Snowing, or Sunny). These dimensions allow for a detailed exploration of how various driving contexts affect music preferences, facilitating a deeper understanding of user behavior and preferences in different driving scenarios. In recommender systems (RS), data is typically structured as a rating matrix, with users represented by columns, items by rows, and observed ratings populating the matrix cells.

Due to the incomplete nature of user ratings across all items, this matrix often exhibits significant sparsity. For context-aware recommendations that incorporate multiple dimensions, data is represented using tensors instead of matrices. However, the Factorization Machines (FM) model requires data to be formatted as Sparse Feature Vectors, also known as One Hot Encoding, a method predominantly used in deep neural network modeling.

To manage and manipulate the data effectively, we utilize the Pandas library [29], which provides robust tools for data analysis, cleaning, exploration, and manipulation. The Sparse Feature Vector representation converts the recommendation data into a series of tuples (x, y), where 'x' is a real-valued feature vector and 'y' the observed rating, enabling the function f(x) = y. This format reframes the recommendation challenge as a standard machine learning prediction problem, allowing for the straightforward appli cation of conventional machine learning algorithms to the data. Additionally, the Sparse Feature Vector format readily accommodates extra contextual dimensions, enhancing the model's capability to handle complex, multi-dimensional datasets effectively.

Table I illustrates more detail about the two datasets.

TABLE I. Statistics about the two datasets.

	DePaulMovie	InCarMusic
Users	124	43
Items	80	138
Ratings	5044	4014
Dimensions	4	7
Sparsity	94%	99%

B. Evaluation Measures

To measure DeepCBFM performance the RMSE and R Squared are used: Root Mean Squared Error (RMSE) is one of the largest used metrics for regression problems. It is the MSE square root which is calculated as the squared differences between actual the target values.

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{m}(y_j - \hat{y})} \quad (9)$$

yj is the real value and $\hat{}yj$ is the target value.

R-squared (R2) is a statistical measure that shows how close the data are to the regression line.

$$R^2 = 1 - \frac{Sumsquaredregression}{Totalsumofsquares} = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \quad (10)$$

Where SSR is the sum square of the difference between real and predicted variables, SST is the total of sum squares and y^{-} is the mean of all values.

C. Performance Comparisons

To confirm the effectiveness of CBFM and DeepCBFM, we select four baseline models:

FM [4]: Factorization Machines are a well-known method in Collaborative Filtering recommendation systems. They work by processing the rating matrix and converting it into a pair of low-rank matrices through a transformational procedure.

CBMF [18]: It is an extension of Matrix Factorization adapted with contextual information.

CANCF [30]: It is a hybrid approach that adapts and repurposes a prefiltering method for integrating context.

CAMCRS [21]: It employs deep learning models to predict ratings while considering contextual factors and simultaneously learns how to aggregate this information effectively.

D. Results of the Experiments

For the development environment, we use TensorFlow [31] to implement the model, installed on a computer using Windows 10 with 16GB of RAM. The implementation is inspired from [32]. Each dataset is splitted to train data (80%) and testing data (20%). The optimization method used is Adam, we fix the learning rate to 0.00001. we use a mini-batch of 4096. To avoid the overfitting problem, we use L2 regularization. We fine-tune two parameters to extract the best performance of DeepCBFM. The first parameter is the embedding size, as shown in figure 3, we observe that the RMSE reaches its lowest value when the size is 32, because a large size brings a better representation capacity to the model. Secondly, we analyze the dropout parameter, which is used to prevent the overfitting.

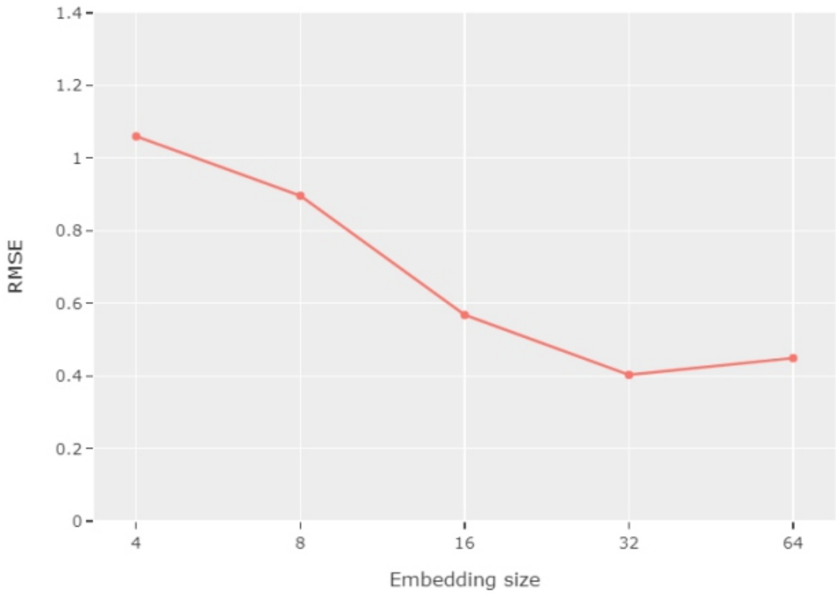


Figure 3. RMSE results of CBFM under the embedding size parameter.

As illustrated in figure 4, the dropout is set to a value

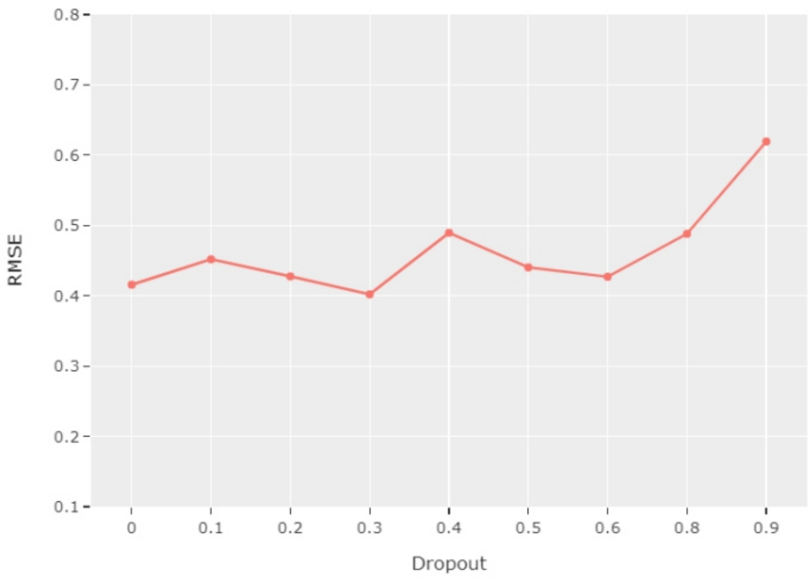


Figure 4. RMSE results of CBFM under the dropout parameter.

between 0 and 0.9 with a jump of 0.1 each time. The results show that the DeepCBFM reaches the lowest RMSE value as the dropout equals 0.3, which justify the usefulness of the dropout. As mentioned earlier, we conducted a comparative analysis of The DeepCBFM against four models, utilizing 2 distinct datasets. Outcomes from both datasets were evaluated based on RMSE and R^2 metrics. Additionally, a specific focus was placed on assessing the effectiveness of the CBFM component in comparison to other methodologies, with a particular emphasis on the FM algorithm.

The RMSE and R-squared results depicted in Figures 5 and 5 offer valuable insights into the performance and predictive capability of different models on the DePaulMovie dataset. Firstly, the DeepCBFM model's attainment of the lowest RMSE value of 0.4027 suggests that its predictions are closest to the actual ratings compared to other models evalu ated. This signifies the model's superior ability to minimize prediction errors and provide more accurate estimations of movie ratings.

Additionally, the observed favorable outcomes of the Context-based Factorization Machines (CBFM) component in contrast to the conventional Factorization Machines (FM) approach highlight the significance of considering contextual information in recommendation systems. Contextual factors such as user preferences and item characteristics play a crucial role in enhancing predictive accuracy, as evidenced by the improved performance of the CBFM com ponent. Moving to the R-squared metric, which measures the goodness of fit of the models, the DeepCBFM model's superiority becomes even more pronounced. By surpassing both the CBMF and FM models in terms of R-squared, the DeepCBFM model demonstrates its ability to explain a larger proportion of the variability in the ratings. This implies that the DeepCBFM model captures more nuanced patterns and underlying relationships within the data, leading to better predictions of movie ratings. The substantial margins by which the DeepCBFM model outperforms the

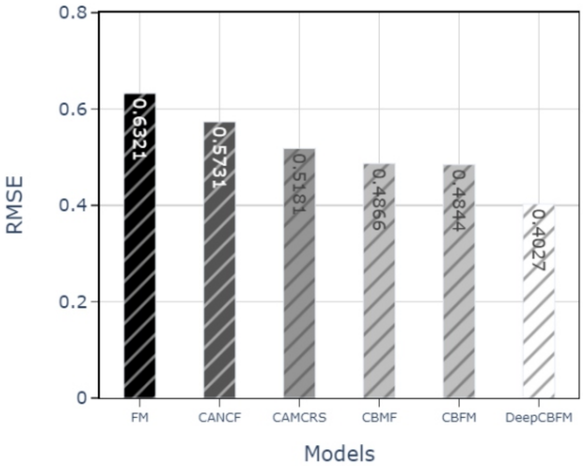


Figure 5. RMSE results obtained for DePaulMovie dataset.

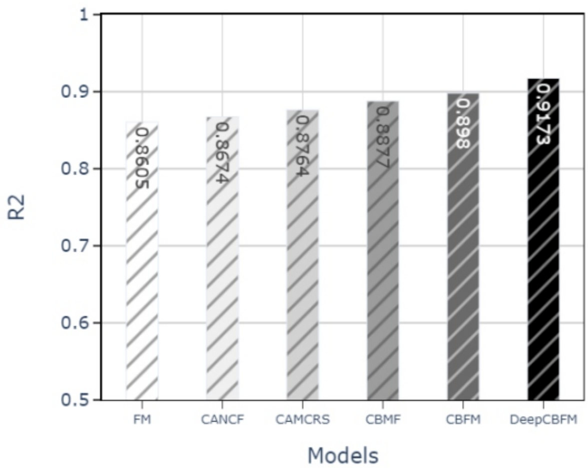


Figure 6. R^2 results obtained for DePaulMovie dataset.

CBMF and FM models in R-squared further underscore its effectiveness in modeling the complex dynamics of movie preferences. Consistent with the findings from the initial dataset, the outcomes from

the Music dataset (figure 7 and 8) reaffirm the ranking of performance among all models.
Notably, the FM model exhibits a decline in results, attributed to the dataset's inherent high contextual dimen sions. Additionally, noteworthy observations include the close similarity in results between CBMF and CBFM mod els. This similarity can be attributed to their shared design focus on capturing low-order feature interactions.
Overall, these results not only highlight the superior per formance of the DeepCBFM model but also emphasize the importance of incorporating contextual information and leveraging advanced modeling techniques, such as deep learning architectures, in recommendation systems. By do ing so, recommendation models can better capture the diverse and intricate factors influencing user preferences,

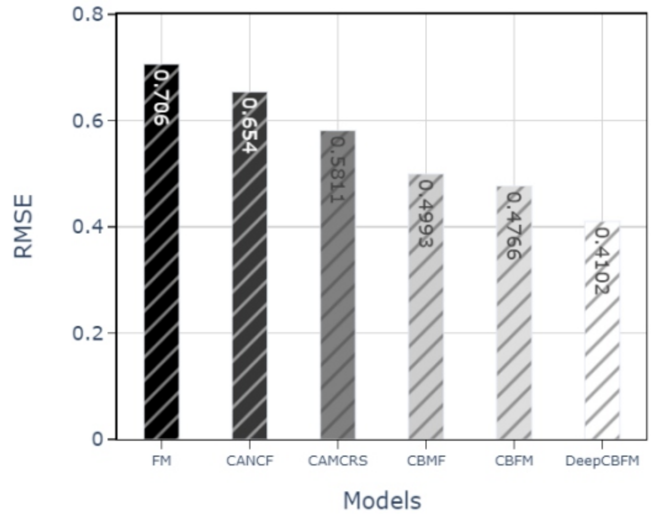


Figure 7. RMSE results obtained for InCarMusic dataset.

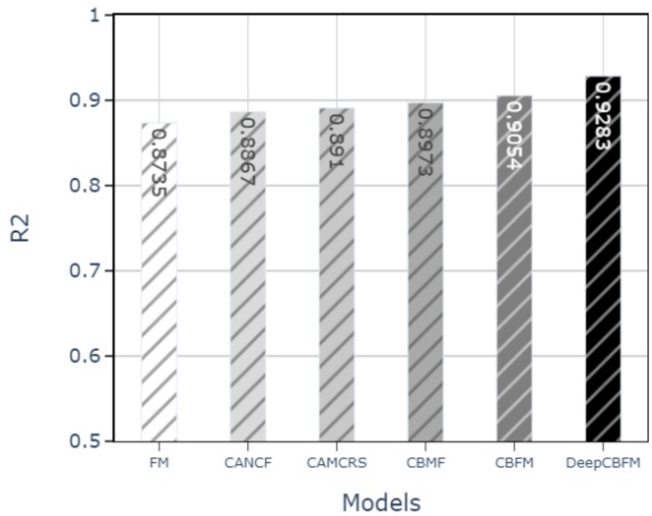


Figure 8. R^2 results obtained for InCarMusic dataset.

thereby enhancing the accuracy and effectiveness of movie recommendations for users.

E. Discussion

This research introduces an innovative methodology that integrates deep learning with a specialized version of Factorization Machine (FM), named DeepCBFM, designed specifically to enhance the performance of recommendation systems. Traditional methods have frequently struggled to fully comprehend and utilize the complex dynamics of contextual information and its profound influence on user preferences. Our approach marks a significant departure from these traditional methods, as it not only incorporates but also excels in parsing the nuanced interactions among features across diverse contexts. This ability to discern how features behave differently when interacting within various contextual frameworks is critical because it allows for more tailored and accurate recommendations. The results from our study underscore that merging deep learning techniques with factorization machines leads to a substantial improve ment in managing datasets characterized by rich contextual dimensions. This enhancement is particularly noticeable when contrasting the performance of the DeepCBFM model with that of traditional FM methods. Such comparisons highlight the limitations of older models in dealing with the complexities inherent in contemporary recommenda tion systems environments. Furthermore, the comparison between the CBMF and CBFM models reveals a significant insight; while these models perform similarly in terms of low-order feature interactions, it is the high-order interac tions, which are more complex and less apparent, captured by the DeepCBFM that are critical for achieving a higher level of accuracy.

Our investigation provides a comprehensive evaluation of the model's performance across two distinct datasets. However, the question of generalizability remains, given that the findings might be influenced by the unique characteristics of the datasets used. Future studies should therefore expand the scope of research to include a broader array of datasets, which vary not only in size but also in the complexity of contextual information they present. This expansion is essential to validate and possibly enhance the efficacy of the DeepCBFM model across different settings and applications.The demonstrated resilience of the DeepCBFM model in handling high contextual dimensions opens up new avenues for research, particularly in improving rec ommendation systems. Upcoming studies could look into integrating additional contextual factors, such as temporal and geographical data, or applying cutting-edge deep learning architectures like neural attention mechanisms or generative adversarial networks, which could further refine the model's predictive accuracy. The strong empirical support obtained from analyzing the DePaulMovie and InCarMusic datasets highlights the superior predictive capability of the DeepCBFM model over both traditional FMs and the more recent CBMF model.

This advancement is of paramount importance in the realm of recommendation systems, where the ability to accurately predict user preferences amidst complex contextual dimen sions can dramatically enhance user satisfaction and en gagement. Our findings suggest a significant paradigm shift towards integrating sophisticated deep learning components into recommendation models, aiming to harness the full spectrum of feature interactions. This shift could potentially lead to revolutionary improvements in how recommendation systems understand and cater to user needs, ultimately enhancing the overall effectiveness and user experience provided by these systems.

5. Conclusions and Future Work

This research presents DeepCBFM, a sophisticated neural network framework built upon Factorization Machines, specifically engineered for Context-Aware Recommender Systems (CARSs). DeepCBFM is designed to address the limitations of traditional baseline models and offers a more adaptable approach to efficiently modeling contextual data.

The model excels in two key areas: firstly, it adeptly manages the variable behaviors of features as they interact with other features from diverse contexts; secondly, it captures high-order feature interactions and addresses nonlinear challenges through advanced deep learning tech niques. Our experimental evaluation of DeepCBFM, uti lizing two real-world datasets—DePaulMovie and InCar Music—demonstrates that our model not only improves prediction accuracy but also outperforms other state-of the-art models. However, our findings also highlight areas for potential future research and development. One such area involves the higher time complexity observed in our model compared to baseline models, suggesting a need for further optimization to enhance computational efficiency and scalability, particularly as we extend the application to larger and more complex datasets. This brings us to another critical point of future exploration: dataset specificity.

The current application of our model may exhibit limited generalizability across different domains or more extensive datasets with varied contextual dynamics.

Future work will need to focus on validating and possibly refining DeepCBFM across a broader spectrum of datasets to ensure its effectiveness and applicability in diverse settings. Another promising direction for future research stems from the challenges associated with acquiring and integrat ing contextual information from real-world applications. To address this, we plan to investigate alternative sources of contextual data, including unstructured data sources, and develop efficient methods for their extraction and utilization. This exploration aims to enrich the contextual understand ing of the model, thereby enhancing its predictive precision and relevance in real-life scenarios.

References

[1] R. MADANI, A. EZ-ZAHOUT, and A. IDRISSI, "An Overview of Recommender Systems in the Context of Smart Cities," in 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), Nov. 2020, pp. 1–9.

[2] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classifi cation: using social and content-based information in recommen dation," in Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intel ligence, ser. AAAI '98/IAAI '98. USA: American Association for Artificial Intelligence, Jul. 1998, pp. 714–720.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in Proceedings of the 1994 ACM conference on Computer supported cooperative work, ser. CSCW '94. New York, NY, USA: Association for Computing Machinery, Oct. 1994, pp. 175–186.

[4] S. Rendle, "Factorization Machines," in 2010 IEEE International Conference on Data Mining, Dec. 2010, pp. 995–1000, iSSN: 2374-8486.

[5] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," in Recommender Systems Handbook, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA: Springer US, 2015, pp. 191–226. [Online]. Available: https://doi.org/10.1007/978-1-4899-7637-6 6

[6] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & Deep Learning for Recommender Systems," Jun. 2016, arXiv:1606.07792 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1606.07792

[7] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorizationmachine based neural network for CTR prediction," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, ser. IJCAI'17. Melbourne, Australia: AAAI Press, Aug. 2017, pp. 1725–1731.

[8] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: learning the weight of feature interactions via attention networks," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, ser. IJCAI'17. Melbourne, Australia: AAAI Press, Aug. 2017, pp. 3119–3125.

[9] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Jul. 2018, pp. 1754–1763,

arXiv:1803.05170 [cs].

[10] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks," in Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Nov. 2019, pp. 1161–1170, arXiv:1810.11921 [cs].

[11] Y. Yu, Z. Wang, and B. Yuan, "An Input-aware Factorization Machine for Sparse Prediction," pp. 1466–1472, 2019. [Online]. Available: https://www.ijcai.org/proceedings/2019/203

[12] J. Pan, J. Xu, A. L. Ruiz, W. Zhao, S. Pan, Y. Sun, and Q. Lu, "Field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising," in Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18, 2018, pp. 1349–1357, arXiv:1806.03514 [cs, stat].

[13] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A Deep Matrix Factorization Method for Learning Attribute Representations," IEEE Transactions on Pattern Analysis and Machine Intel ligence, vol. 39, no. 3, pp. 417–429, Mar. 2017, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[14] R. Lara-Cabrera, A. Gonzalez-Prieto, and F. Ortega, "Deep Matrix Factorization Approach for Collaborative Filtering Recommender Systems," Applied Sciences, vol. 10, no. 14, p. 4926, Jan. 2020, number: 14 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/10/14/4926

[15] A. Ez-zahout, H. Gueddah, A. Nasry, R. Madani, and F. Omary, "A hybrid big data movies recommendation model based k nearest neighbors and matrix factorization," Indonesian Journal of Electrical Engineering and Computer Science, vol. 26, no. 1, pp. 434–441, Apr. 2022, number: 1. [Online]. Available: https://ijeecs.iaescore.com/index.php/IJEECS/article/view/26415

[16] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in Proceedings of the fifth ACM conference on Recommender systems, ser. RecSys '11. New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 301–304.

[17] R. Madani and A. Ez-zahout, "A Review-based Context-Aware Recommender Systems: Using Custom NER and Factorization Machines," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 3, 2022, number:3 Publisher: The Science and Information (SAI) Organization Limited. [Online]. Available: https://thesai.org/Publications/ViewPaper?Volume=13&Issue=3&Code=IJACSA&SerialNo=65

[18] M. Casillo, B. B. Gupta, M. Lombardi, A. Lorusso, D. Santaniello, and C. Valentino, "Context Aware Recommender Systems: A Novel Approach Based on Matrix Factorization and Contextual Bias," Electronics, vol. 11, no. 7, p. 1003, Jan. 2022, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/11/7/1003

[19] S.-Y. Jeong and Y.-K. Kim, "Deep Learning-Based Context-Aware Recommender System Considering Contextual Features," Applied Sciences, vol. 12, no. 1, p. 45, Jan. 2022, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/12/1/45

[20] A. Sattar and D. Bacciu, "Graph Neural Network for Context Aware Recommendation," Neural Processing Letters, vol. 55, no. 5, pp. 5357–5376, Oct. 2023. [Online]. Available: https: //doi.org/10.1007/s11063-022-10917-3

[21] S.-L. Vu and Q.-H. Le, "A Deep Learning Based Approach for Context-Aware Multi-Criteria Recommender Systems," Computer Systems Science and Engineering, vol. 44, no. 1, pp. 471–483, 2023. [Online]. Available: https://www.techscience.com/csse/v44n1/48080

[22] H. Vaghari, M. H. Aghdam, and H. Emami, "Diarec: Dynamic Intention-Aware Recommendation

with Attention-Based Context Aware Item Attributes Modeling," Journal of Artificial Intelligence and Soft Computing Research, vol. 14, no. 2, pp. 171–189,Mar. 2024. [Online]. Available: https://www.sciendo.com/article/10.2478/jaiscr-2024-0010

[23] A. D. Vecchia, B. Oliboni, and E. Quintarelli, "ICARE: the princi ples of Explainable AI in a Context-aware Recommendation APP."

[24] T. Dilekh, S. Benharzallah, A. Mokeddem, and S. Kerdoudi,"Dynamic Context-Aware Recommender System for Home Automation Through Synergistic Unsupervised and Supervised Learning Algorithms," Acta Informatica Pragensia, vol. 13, no. 1, pp. 38–61, Apr. 2024, publisher: Acta Informatica Pragensia. [Online]. Available: https://doi.org/10.18267/j.aip.228

[25] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and Scalable Response Prediction for Display Advertising," ACM Transactions on Intelligent Systems and Technology, vol. 5, no. 4, pp. 61:1–61:34, Dec. 2015. [Online]. Available: https://doi.org/10.1145/2532128

[26] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: estimating the click-through rate for new ads," in Proceedings of the 16th international conference on World Wide Web, ser. WWW '07. New York, NY, USA: Association for Computing Machinery, May 2007, pp. 521–530.

[27] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J.Lin, "Training and Testing Low-degree Polynomial Data Mappings via Linear SVM," The Journal of Machine Learning Research, vol. 11, pp. 1471–1490, Aug. 2010.

[28] Y. Zheng, B. Mobasher, and R. Burke, "CARSKit: A Java-Based Context-Aware Recommendation Engine," in 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Nov. 2015, pp. 1668–1671, iSSN: 2375-9259.

[29] W. McKinney et al., "Data structures for statistical computing in python," in Proceedings of the 9th Python in Science Conference, vol. 445. Austin, TX, 2010, pp. 51–56.

[30] I. M. A. Jawarneh, P. Bellavista, A. Corradi, L. Foschini, R. Montanari, J. Berrocal, and J. M. Murillo, "A Pre-Filtering Approach for Incorporating Contextual Information Into Deep Learning Based Recommender Systems," IEEE Access, vol. 8, pp. 40 485–40 498, 2020, conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/document/9004579

[31] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: a system for large-scale machine learning," in Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, ser. OSDI'16. USA: USENIX Association, Nov. 2016, pp. 265–283.

[32] W. Shen, "Deepctr: Easy-to-use,modular and extendible package of deep-learning based ctr models," https://github.com/shenweichen/ deepctr, 2017.

Abderrahmane EZ-ZAHOUT is currently an assistant professor of computer science at department of computer science/faculty of sciences at Mohammed V University. His researches are in fields of computer sciences, digital systems, big data and computer vision. Recently, He works on intelligent systems. He has served as invited reviewer for many journals. Besides, he is also involved in NGOs, student associations, and managing nonprofit foundation. He can be contacted at email: abderrahmane.ezzahout@um5.ac.ma.

Fouzia Omary is the leader IPSS team research and currently a full professor of computer science at department of computer science/ENS, Mohammed V University. She is a full-time researcher at Mohammed V University and she is the chair of the international conference on cybersecurity and blockchain and also the leader of the national network of blockchain and cryptocurrency. she can be contacted at email: omary@fsr.ac.ma.

Rabie MADANI received the master's degree in Big Data in 2018 from Mohammed V University, Rabat, Morocco and he is currently a Ph.D. student at the faculty of sciences, Mohammed V University in computer science. He is Head of IT at the Institute of Africans Studies in Rabat. His research domains include artificial intelligence, big data, recommender systems and text mining. has authored and coauthored many publications in international journals and conferences. He is currently a member of the Intelligent Processing Systems and Security team. He can be contacted at email: rabie.madani@um5.ac.ma.

Abdelhaq Chedmi earned his master's degree in Big Data from Mohammed V University in Rabat, Morocco. Presently, he is a Ph.D. student in computer science at the Faculty of Sciences, Mohammed V University. Alongside his academic pursuits, he holds the position of Head of IT at the Institute of Africans Studies in Rabat. His research focuses on artificial intelligence, big data, recommender systems, and text mining. Abdelhaq has contributed to several publications in international journals and conferences, establishing himself as a valuable member of the Intelligent Processing Systems and Security team. Abdelhaq_chedmi@um5.ac.ma.

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).
2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);
3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);
4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Note