IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 948~960
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp948-960

Source printer identification using convolutional neural network and transfer learning approach

Naglaa F. El Abady[1], Hala H. Zayed[1,2], and Mohamed Taha[1]
[1]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt
[2]School of Information Technology and Computer Science (ITCS), Nile University, Giza, Egypt

Article Info

Article history:

Received Jan 20, 2023
Revised May 27, 2023
Accepted Jun 3, 2023

Keywords:

Convolution neural network
Document forgery
Source printer identification
Transfer learning
Two-dimensional convolution neural network

ABSTRACT

In recent years, Source printer identification has become increasingly important for detecting forged documents. A printer's distinguishing feature is its fingerprints. Each printer has a unique collection of fingerprints on every printed page. A model for identifying the source printer and classifying the questioned document into one of the printer classes is provided by source printer identification. A paper proposes a new approach that trains three different approaches on the dataset to choose the more accurate model for determining the printer's source. In the first, some pre-trained models are used as feature extractors, and support vector machine (SVM) is used to classify the generated features. In the second, we construct a two-dimensional convolutional neural network (2D-CNN) to address the source printer identification (SPI) problem. Instead of SoftMax, 2D-CNN is employed for feature extractors and SVM as a classifier. This approach obtains 93.75% 98.5% accuracy for 2D-CNN-SVM in the experiments. The SVM classifier enhanced the 2D-CNN accuracy by roughly 5% over the initial configuration. Finally, we adjusted 13 already-pre-trained CNN architectures using the dataset. Among the 13 pre-trained CNN models, DarkNet-19 has the greatest accuracy of 99.2 %. On the same dataset, the suggested approaches achieve well in terms of classification accuracy than the other recently released algorithms.

Corresponding Author:

Naglaa F. El Abady
Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University
Benha Mansoura Road, next to Holding Company for Water Supply and Sanitation Benha
Qalyubia Governorate, Egypt
Email: naglaa.fathy@fci.bu.edu.eg

1. INTRODUCTION

The process of examining and analyzing digital evidence to determine a crime's occurrence is known as digital forensics. To locate, collect, and analyze digital evidence, digital forensics requires specialized tools and processes [1]. The use of digital documents has exploded in the recent decade. These digital papers could include images of official contracts, bills, checks, and so on. The goal of these digital papers is to eliminate the need for paper. Compared to a hard copy, maintaining a digital document is also simpler, cheaper, and more effective, but safety is a problem.

Personal computers, scanners, and printers can produce forged documents such as certificates, agreements, identity cards, and lottery tickets. Modern printers have such high resolution that it is difficult for normal persons to differentiate forged documents from real ones. Traditional approaches use chemical techniques to detect forgeries in printed documents [2], [3]. These procedures need laboratory tools and a specialist to evaluate the samples. Additionally, these methods take a long time and risk damaging the printed paper. Digital techniques, in contrast, use a reference scanner to turn printed papers into their digital equivalent. Using digital approaches for source printer identification, it is possible to distinguish between documents printed on various printers. Since all the analysis is done digitally, it is quicker and more automatic.

Extrinsic (active) and intrinsic (passive) are the two main digital methods for identifying source printers. Prior research typically used the following techniques to extract statistical features from printed documents: Discrete wavelet transform (DWT), local binary pattern (LBP), key printer noise features (KPNF), gray-level co-occurrence matrix (GLCM), speeded up robust features (SURF), orientated FAST rotated and BRIEF (ORB), Histogram of gradient (HOG) and spatial filters [4]–[10], [11]–[16]. Subsequently, the forensic classification system adopts support vector machine (SVM), random forest (RF), and ensemble techniques. However, the tasks involving feature extraction, feature selection, and classification, as indicated in the preceding statement, demand much professional human participation. Additionally, to obtain results that may be generalized, the entire procedure must be repeated several times using a random selection of training and testing samples. Moreover, machine learning methods have fundamentally altered the research area for both academia and industry. A branch of artificial intelligence is machine learning (AI) [7]. The general goal of machine learning is to identify the structure of data and fit it into models that are both useful and understandable.

Deep learning needs a large number of datasets to give more accuracy. In this research, we solve the problem of the large dataset by using transfer learning. The transfer learning convolutional neural network (CNN) was used to save time for creating a model from scratch, training, and small datasets. Transfer learning CNN is divided into two types: the first used a pre-trained model for feature extraction, and the result was classified using machine learning classification techniques. The second used a pre-trained model but changed the last layers. We also used 2D-CNN models with SoftMax or SVM classifiers to determine the printer's source. The significant contributions of this work include the following,

- Up to our experience, this is the first approach using CNN for source printer identification (SPI) of full printed text documents without dividing the document into letters, words, or patches.
- Multiple pre-trained CNN models are modified to increase the efficiency of detecting forgery in documents based on the SVM classifier instead of SoftMax.
- We successfully developed a pre-processing stage that included histogram equalisation and gamma correction, which significantly improved the model's performance and accuracy.
- The proposed approach performs better in classification accuracy than the other recently published algorithms on the same dataset.

The following sections make up the entire paper. While section 3 focuses on the background, section 2 provides a short description of the related work for classifying the source printer of a printed text document. Section 4 contains a description of the specifics of our proposed approach. The efficiency of the proposed approach is investigated using a detailed series of experiments. The proposed approach's description and outcomes have been explored in section 5. Finally, conclusions from this effort are presented in section 6.

2. RELATED WORK

Document manipulation can be detected using a variety of approaches. The bulk of these approaches detects the printer's source to identify the printer types employed in the printing process. The issue of source printer (SP) classification has received a lot of attention in the last decade [17]. This section will go over the most popular ways for authenticating a document and ensuring that it was printed by a legal printer. Mikkilineni *et al.* [4], proposed a printer identification procedure based on an SVM classifier. They investigated the impact of font size, font type, paper type, and printer age. When the variables font size, paper type, and printer age are constant, their printer identification technique works. Jung *et al.* [5], introduced a novel colour laser printer forensic algorithm. It uses an SVM classifier and noisy texture analysis. To estimate invisible noises, the Wiener filter and the 2D DWT filter are used. The GLCM is then applied to the noise texture. The machine classifier is trained and validated using data from 384 statistical features. For brand, toner, and model recognition, the presented technique achieves 99.3%, 97.4%, and 88.7% accuracy, respectively. The authors in [6], described a method for identifying document forgeries based on distortion mutation of geometric parameters (DMGP) with translation and rotation distortion parameters. This technique can be used to examine papers in both English and Chinese. It can analyze papers based on their individual characters. It is resistant to JPEG compression and performs well with low-resolution documents. The GLCM and DWT were used to extract texture features from the Chinese printed source to evaluate the effects of different output devices [7]. The feature selection techniques are utilized to select the most appropriate feature subset and an SVM to determine the source model of the documents. The overall experimental results achieve an identification rate of 98.64%, 1.27% higher than the GLCM method that was previously used.

The result in [8], image processing techniques and data exploration techniques are used to calculate several significant statistical features, such as the Spatial filters, LBP, Wiener filter, GLCM, Gabor filter, DWT, Haralick, and segmentation-based fractal texture analysis (SFTA) features. The LBP approach yields the best rate of identification. It is said to be better than other approaches in many ways. Tsai described a method for examining the link between printed Chinese characters and digital printers in [9], The feature selection decision fusion and SVM-based classification are utilized. The most significant features are chosen methodically from the GLCM, DWT, spatial filters, Wiener filter, and Gabor filter. When compared to other methodologies, the GLCM method has the highest identification accuracy rate. The word in [10], a collection of characteristics for describing geometric distortions at the text-line level is presented. Experiments on 14 printers discovered that the proposed approach outperforms the present state-of-the-art geometric distortion method. When working with a limited training size, it provides significantly higher accuracy. The average classification accuracy of a classifier trained with one page, one printer, one font, three different fonts, and 14 printers was 98.85%. A proposed technique in [11], used all printed letters at the same time to find the originating printer from scanned images of printed documents. An individual classifier is used to classify all printed letters as well as local texture pattern-based features. The method was tested on a publicly available dataset of 10 printers as well as a new dataset of 18 printers scanned at 600 and 300 dpi resolution and created in four different fonts. The reseacher of [12], used a passive technique to identify the document source printer. Key printer noise features (KPNF), speeded up robust features (SURF), erientated FAST rotated, and BRIEF are some feature extraction methodologies that have been used. For the classification job, three classification processes are considered: k-nearest-neighbor (k-NN), random forest, and decision tree. These three classification techniques received the most votes. The best accuracy of 95.1% was obtained by combining ORB, KPNF, and SURF with an RF classifier and an adaptive boosting technique.

Printer identification using GLCM is presented in [13]. A feature vector is created by extracting a set of features from each character for each letter "e" in the document. Following that, a 5-nearest-neighbor (5-NN) classifier is used to classify each feature vector. With training, this approach is unaffected by font type or size, although cross-font and cross-size testing yielded mixed results. A separate 5-NN classifier block for each character would be required to classify a document using all its characters, not only "e"s. The classifier becomes more complex as a result. Techniques for colour and picture documents produced by inkjet printers must also be researched. A text-independent method for adequately describing source printers using deep visual Features has been implemented in [14]. Using transfer learning on a pre-trained CNN, the system detected 1200 documents from 20 dissimilar (13) laser and (7) inkjet printers. Anselmo investigated systems that can examine discriminant-printing patterns directly from existing data in [15]. This helped him to eliminate any prior assumptions about the printing artifact that distinguish each printer. The experimental results proved that the system is robust to noisy data and outperforms its counterparts. A unique approach based on the SURF, oriented FAST rotated, and BRIEF feature descriptors is proposed in [16]. Classification was achieved through the use of Random Forest, Naive Bayes, k-NN, and other classifier combinations. The model was able to correctly classify the questioned documents and assign them to the appropriate printer. Using a combination of Naive Bayes, k-NN, Random Forest classifiers, a basic majority voting system, and adaptive boosting algorithms, the accuracy was 86.5%. Based on source printer identification (SPI), a text-independent technique for identifying document forgeries is proposed [18]. The top, middle, and bottom parts of the image are divided. There are two feature extraction algorithms used: HOG and LBP. For printer identification, classification methods such as decision trees, k-NN, SVM, random forests, bagging, and boosting are taken into consideration. The best classification accuracy of 96% is attained with the AdaBoost classifier.

3. BACKGROUND

This section introduces convolutional neural networks (CNN) and various types of CNNs, along with their details. It also covers transfer learning and its advantages. Additionally, this section explains the details of support vector machines (SVM) and the reasons for using them.

3.1. Convolution neural network (CNN)

CNN is an artificial neural network suggested by Lecun *et al.* [19]. It has become one of the best-known images and speech recognition tools. Its common weights network structure is similar to the simulation's existing biological neural network. This feature can also decrease the network model's complexity and the number of parameters. CNN can directly use the image as the input, avoiding the traditional identification method of complex feature extraction and data reconstruction. Table 1 shows details of different types of CNN models. Convolutional, activation, pooling, and fully connected layers make up most of a typical CNN, as seen in Figure 1. To extract features from the input images, convolutional layers of CNN are mostly utilized. Activation layers are used to enable nonlinear mapping, which enhances feature maps' capacity for expression.

Typically, as the images are down sampled, the pooling layer reduces the dimension of the features. The fully connected layers are typically put at the top, near the output layers. Fully connected layers are employed as the classifier and final output layers for classification tasks [20].

Table 1. Different types of CNN models

Name Network	Year	Depth	Input size	Parameters
AlexNet [21]	2012	8	227*227*3	61M
VGGNet-16 [22]	2014	16	224*224*3	138M
VGGNet-19		19		144M
GoogleNet [23]	2014	22	224*224*3	7M
ResNet-18 [24]	2016	18	227*227*3	11.7M
ResNet-50		50		25.6M
ResNet-101		101		44.6M
Inceptionv3 [25]	2016	48	299*229*3	23.9M
Squeeze Net [26]	2016	18	227*227*3	1.24M
XceptionNet [27]	2017	71	299*299*3	22.9M
DarkNet-19 [28]	2017	19	256*256*3	20.8M
DarkNet-53		53		41.6M
ShuffleNet [29]	2018	50	224*224*3	1.4M

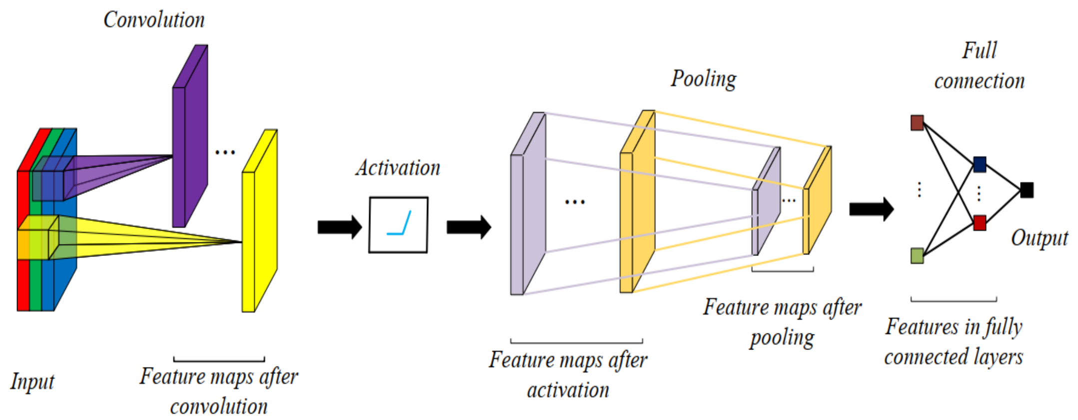


Figure 1. The general architecture of CNN [20]

3.2. Transfer learning (TL)

Deep learning's success is also strongly connected to large amounts of data, which means that a lack of training data can seriously affect the performance of deep learning models. Transfer learning was introduced to resolve this problem. Transfer learning (TL) is often employed when a new dataset is smaller than the previous dataset used to train the learned model [30]. Although it has many advantages, the key ones include reducing training time, improving neural network performance, and not requiring a lot of data. Transfer learning can be done in two ways: either through feature extraction or fine-tuning. When executing feature extraction, the pre-trained network is viewed as an arbitrary feature extractor. This allows the input image to propagate forward, stopping at the pre-specified layer, taking the outputs of that layer as our features, then applying machine learning classifiers such as SVM. In contrast to feature extraction, when fine-tuning is carried out, a new fully connected head is constructed and added to the base architecture. The new fully connected (FC) layer head is randomly set. (Just like any other layer in a new network) and connected to the body of the original network. Using fine-tuning, we may use networks that have already been trained to distinguish classes that weren't initially trained. Figure 2 depicts the transfer learning fine-tuning process.

3.3. Support vector machine (SVM)

The SVM classifier is commonly used in machine learning algorithms for binary classification [5]. SVM is a crucial machine learning method that is primarily employed for multi-class classification problems. Regardless of their dimensional spaces, it has concentrated on classifying features that have been retrieved beforehand. The best linear decision surface is produced by minimizing training feature vectors. The feature vectors are designed in the high-dimensional feature space. The SVM determines the extreme margin hyperplane to partition the various class feature vectors. The performance of the classification is enhanced if the margin between the vectors is large.

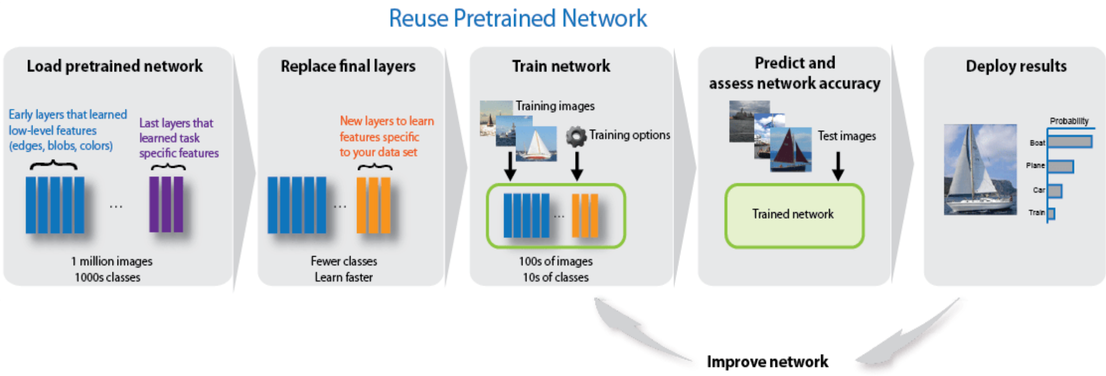


Figure 2. The transfer learning fine-tuning process

4. THE PROPOSED ALGORITHM

Every printer has its own set of fingerprints. On every printed page, there are some fingerprints left by the printer. These fingerprints are a printer's distinguishing feature. This research provides an algorithm for identifying the source printer and categorizing the questioned document into one of the printer types. Deep learning requires a large database to achieve higher accuracy. Therefore, most current research divides scanned documents into characters, words, or patches to enlarge datasets. Up to our knowledge, the proposed algorithm is the first research to apply deep learning with limited datasets and without dividing the scanned document into characters, words, or patches utilizing CNN models' transfer learning. Three different CNN approaches are employed to determine the types of printers with small datasets in this research. First, a transfer learning of pre-trained CNN models for feature extractions is employed. It classifies the result using an SVM classifier. Second, the 2D-CNN is used to generate a trained model from scratch on a given dataset [31], [32]. Then the pre-trained model is used for feature extraction and classification with SVM. The third approach fine-tunes the pre-trained CNN models by using transfer learning, which means that a new fully connected layer replaces the last fully connected or learnable layer of a CNN model with the number of classes in the datasets.

4.1. Pre-processing

The pre-processing phase is utilized in the training and testing phases. For the pre-processing step, there are three methods: Histogram equalization (HE), gamma correction, and resizing. Histogram equalization (HE) [33] helps normalize image grey-scale values and improve brightness discrimination between foreground and background images. The primary objective of the HE is to generate an image with a consistent distribution throughout the entire brightness scale by using the cumulative density function of the image as a transfer function [34]. The histogram function is written as (1),

$$H(f) = \frac{C(f) - C(f)_{min}}{(W \times H) - C(f)_{min}} \times (G_L - 1) \tag{1}$$

where, $H(f)$ denotes the histogram function of the image, $C(f)$ indicates the cumulative function, $C(f)_{min}$ signifies the least non-zero value of the cumulative distribution function, W × H gives the image's number of pixels, and GL defines the number of grey levels utilized. Gamma correction is a nonlinear process used to manage an image's overall brightness. By translating the values of the input intensity image to new values, it improves the image's contrast. The Gamma is derived by (2),

$$N_i = G(\alpha) \times (O_i)^{\frac{1}{\gamma}} \tag{2}$$

where, N_i indicates the new intensity value, Oi represents the old intensity value, G(α) identifies the gray stretch parameter utilized to linearly scale the outcome on the image of [0, 255], and λ signifies the positive constant. Gamma can be any value between 0, and infinite mapping is linear when Gamma is 1 (the default value). Gamma is weighted toward higher (brighter) output values when it is less than 1. The mapping is weighted toward lower (darker) output values if Gamma is greater than 1 as shown in Figure 3. Finally, the input image is resized to match each model's input size because each CNN model has an input size. The histogram equalization and gamma correction of the image are shown in Figure 4.

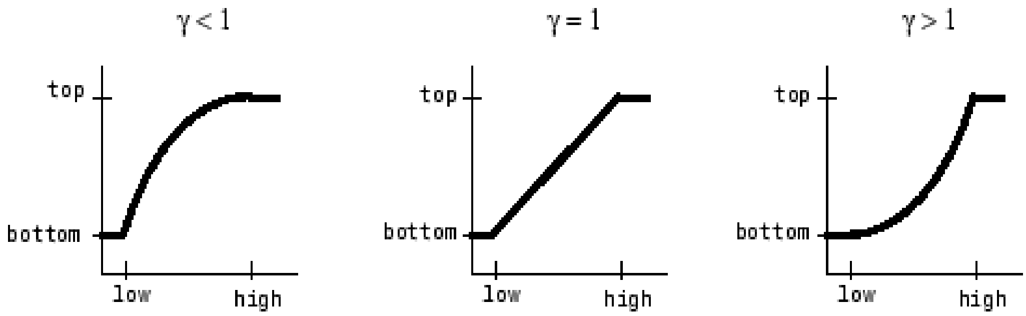
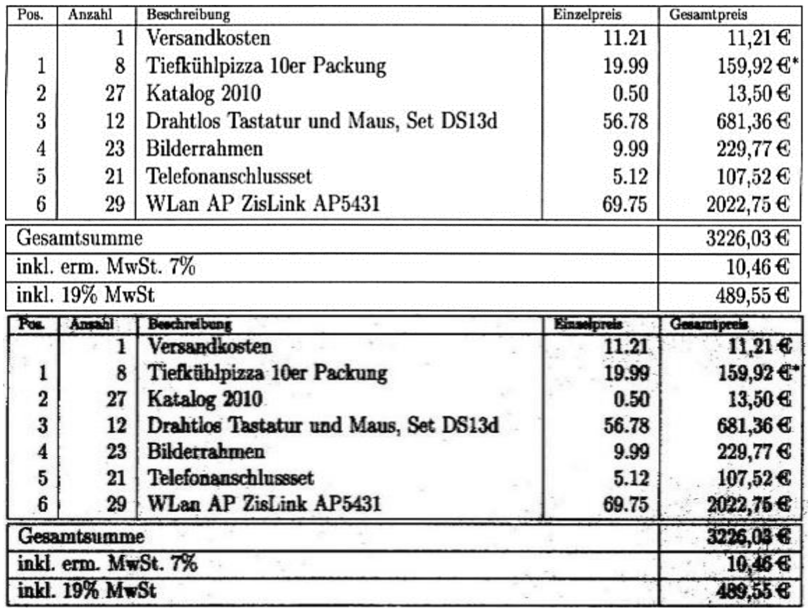


Figure 3. Plots showing three different gamma correction settings



(a)

Pos.	Anzahl	Beschreibung	Einzelpreis	Gesamtpreis
	1	Versandkosten	11.21	11,21 €
1	8	Tiefkühlpizza 10er Packung	19.99	159,92 €*
2	27	Katalog 2010	0.50	13,50 €
3	12	Drahtlos Tastatur und Maus, Set DS13d	56.78	681,36 €
4	23	Bilderrahmen	9.99	229,77 €
5	21	Telefonanschlussset	5.12	107,52 €
6	29	WLan AP ZisLink AP5431	69.75	2022,75 €
Gesamtsumme				3226,03 €
inkl. erm. MwSt. 7%				10,46 €
inkl. 19% MwSt				489,55 €

(b)

Pos.	Anzahl	Beschreibung	Einzelpreis	Gesamtpreis
	1	Versandkosten	11.21	11,21 €
1	8	Tiefkühlpizza 10er Packung	19.99	159,92 €*
2	27	Katalog 2010	0.50	13,50 €
3	12	Drahtlos Tastatur und Maus, Set DS13d	56.78	681,36 €
4	23	Bilderrahmen	9.99	229,77 €
5	21	Telefonanschlussset	5.12	107,52 €
6	29	WLan AP ZisLink AP5431	69.75	2022,75 €
Gesamtsumme				3226,03 €
inkl. erm. MwSt. 7%				10,46 €
inkl. 19% MwSt				489,55 €

Figure 4. Comparison of (a) original image and (b) output image after applying histogram equalization and gamma correction

4.2. Using transfer learning via feature extraction and SVM classifier

This section proposes an approach for identifying the source printer. Instead of constructing and training a deep learning model from scratch, we used transfer learning to transfer the learning abilities to our application. This approach extracted features using 13 well-known pre-trained CNN models (AlexNet, VGG-16, VGG-19, GoogleNet, DarkNet-19, DarkNet-53, ResNet-18, ResNet-50, ResNet101, SqueezeNet, XceptionNet, shuffleNet, inceptionv3). SVM is used to classify the resulting features. The model that uses transfer learning via feature extraction to extract features and classify them using an SVM classifier is shown in Figure 5.

4.3. Source printer identification based on the 2D-CNN model

This section proposes the 2D-CNN approach for identifying source printers. The method was split into two parts: feature extraction using a 2D-CNN model and a source printer predictor using an SVM. Figure 6 depicts the identification procedures: all the scanned documents of the datasets are pre-processed, as described in section 4.1. After all the documents have been pre-processed, the next step is applying the 2D-CNN model. A multi-layer neural network is made up of various combinations of convolutional, rectified linear unit (ReLU), and pooling layers. Train the system using samples, and it will learn the features of images from various printers. SVM classifiers replaced SoftMax. As a result, the SVM is trained using the outputs from the preceding layer (Layer 9). The extracted features and the source printer are then identified from the testing images. After many experiments, the parameters listed in Table 2 were obtained. These parameters were used to modify the proposed 2D-CNN architecture.

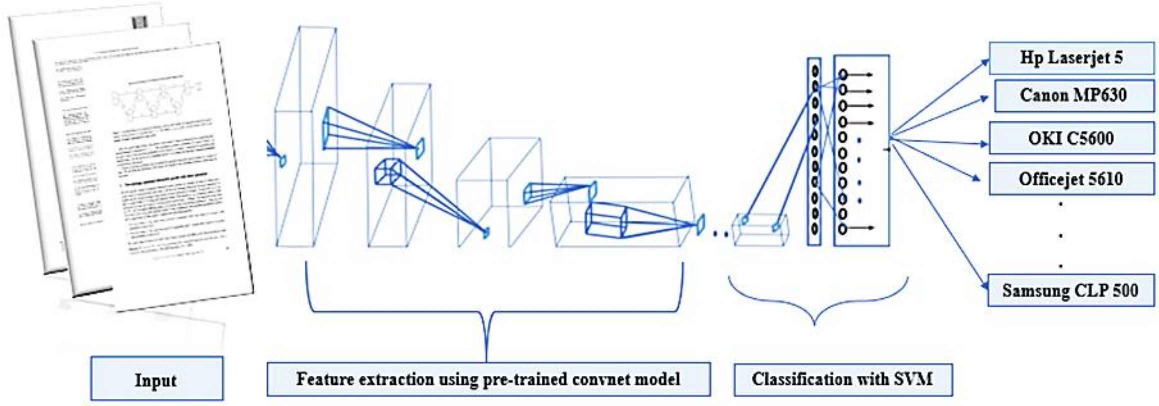


Figure 5. Using the transfer learning approach via extracting features and classifying using an SVM classifier

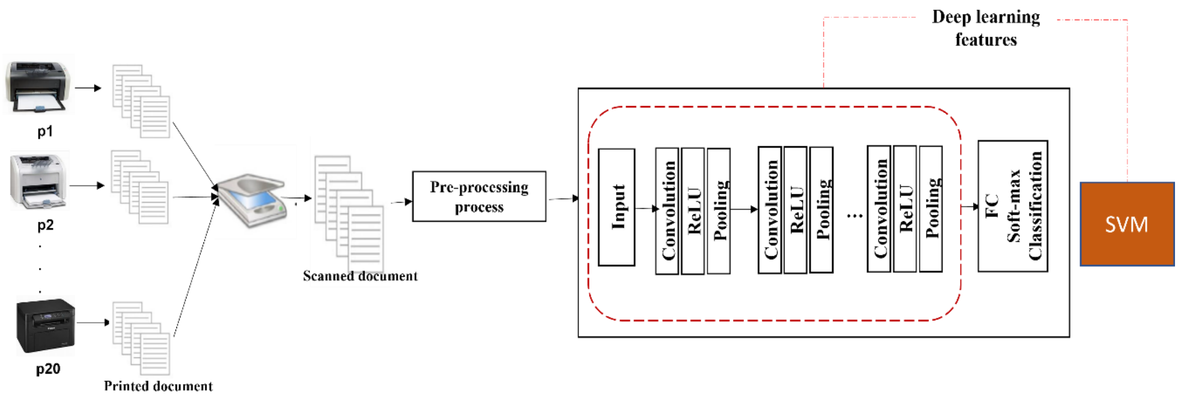


Figure 6. The architecture of the proposed 2D-CNN

Table 2. 2D-CNN design model architecture

Layer	Type	Filters	Kernel Size	Kernel stride	Padding	Output Size
1	Convolution1 /RELU	32	5 × 5	1	2	256 ×256 ×32
2	Maxpool_1	-	2 ×2	2	0	128 ×128 ×32
3	Convolution 2/RELU	32	5×5	1	2	128×128×32
4	Maxpool_2	-	2×2	2	0	64×64×32
5	Convolution 3/RELU	64	5×5	1	2	64×64×64
6	Maxpool_3	-	2×2	2	0	32×32×64
7	Convolution 4/RELU	64	5×5	1	2	32×32×64
8	Average pool_1		2×2	2	0	16×16×64
9	Fully Connected (FC)	-	-	-	-	1×1×20
10	SoftMax	-	-	-	-	1×1×20
11	Class Output	-	-	-	-	1×1×20

4.4. Using transfer learning via fine-tuning

Transfer learning, which has been used in many fields, allows for the transfer of information from one domain to another [35]. Fine-tuning is a process for training a new ConvNet by transferring the weights of pre-trained ConvNets (AlexNet, VGG-16, VGG-19, GoogleNet, DarkNet-19, DarkNet-53, ResNet-18, ResNet-50, ResNet101, SqueezeNet, XceptionNet, shuffleNet, inceptionv3), and it has been effectively employed in a variety of applications. All pre-trained ConvNets' last FC layer neurons are changed to match the number of classes in the current classification task. Transfer learning via fine-tuning We're working on a 20-class classification task. We reduce the output layer of networks from 1,000 to 20 to fit our dataset of 20 printer classes. The learning and classification layers that exchange in each pre-trained CNN is shown in Table 3. Stochastic Gradient Descent (SGD) is applied to fine-tune the network utilizing error backpropagation with a tiny learning rate of 0.0001. Figure 7 depicts a model that employs transfer learning via fine-tuning.

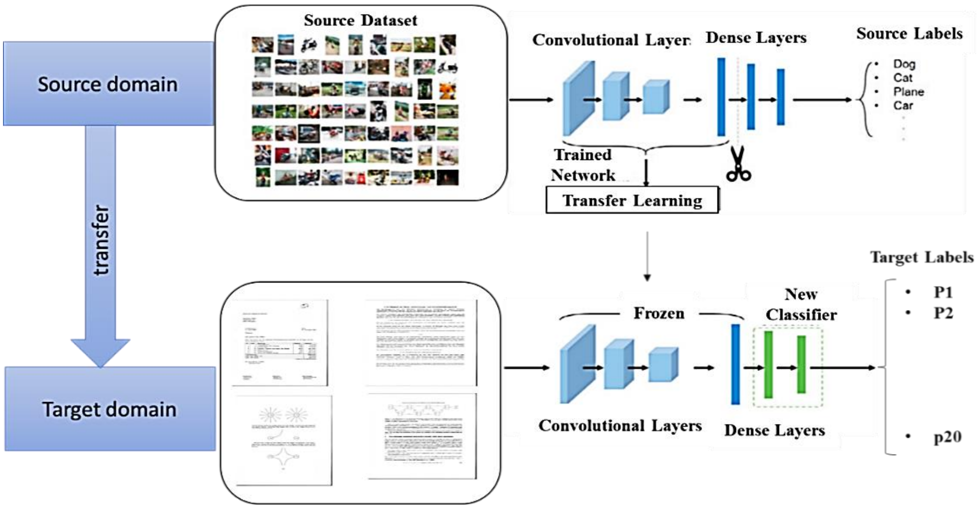


Figure 7. Using the transfer learning approach via fine-tuning

Table3. The learning and classification layers that are exchanged in each pre-trained CNN

Network	Learning layer replaced	ClassLayer replaced
AlexNet	Fc8	Output
ResNet-18	Fc1000	Classification Layer predictions
ResNet-50	Fc1000	ClassificationLayer_fc1000
ResNet-101	Fc1000	ClassificationLayer_predictions
GoogleNet	loss3-classifier	Output
VGG-16	Fc8	Output
VGG-19	Fc8	Output
DarkNet-19	conv19	Output
DarkNet-53	conv53	Output
SqueezeNet	conv10	ClassificationLayer_predictions
XceptionNet	Predictions	ClassificationLayer_predictions
Inceptionv3	Predictions	ClassificationLayer_predictions
ShuffleNet	node_202	ClassificationLayer_node_203

5. EXPERIMENTAL SETUP AND DATASET

The proposed approaches were implemented in MATLAB R2021b and tested on a DELL PC with the following configuration: Intel(R) Core (TM) i7-11800H@2.30 GHz, 6 GHz GPU, 16 GB RAM, 64-bit Windows 11. Several experiments were performed to evaluate the performance of the proposed approaches. The datasets used to train and test the suggested approaches are described in section 5.1. Section 5.2 describes the experiment's setup, whereas section 5.3 describes the evaluation measures. The fourth subsection introduces a discussion of the results. Finally, a comparison of the three approaches to other methods is given and comparison with other methods.

5.1. Dataset's description

Khanna *et al.*'s public dataset [36] is used to test the proposed approach. The documents in this dataset were printed on 13 laser printers and 7 inkjet printers. Each printer is given a total of 60 documents to consider. A printer's documents are all one- of-a-kind. The dataset contains documents from three categories: contracts, invoices, and scientific papers. The printer model's datasets used in this study are listed in Table 4.

5.2. Evaluation metrics

The performance of the proposed algorithm is assessed using a variety of evaluation metrics, including confusion matrix, accuracy, recall, precision, and F-measure metrics [37]–[39]. The confusion matrix is a cross table that records the number of occurrences between two rates, the actual classification, and the predicted classification. The rows show the actual classification, whilst the columns provide a model prediction. The classes are mentioned in the same order in the rows as in the columns. As a result, the main diagonal is where the correctly classified items are placed. Table 5 displays metrics for classification evaluations, where TP: true positive, FN: false negative, FP: false positive, and TN: true negative.

Table 4. Printer models in the dataset

Dataset	Printer type	Printer	# Documents	Abbreviations
01	Ink	Officejet 5610	60	P_1
02	Laser	Samsung CLP 500	60	P_2
03	Laser	Ricoh Aficio MPC2550	60	P_3
04	Laser	HP LaserJet 4050	60	P_4
05	Laser	OKI C5600	60	P_5
06	Laser	HP LaserJet 2200dtn	60	P_6
08	Laser	Ricoh Afico Mp6001	60	P_7
11	Ink	Epson Stylus Dx 7400	60	P_8
13	Ink	Unknown	60	P_9
19	Laser	HP Color LaserJet 4650dn	60	P_10
20	Laser	Nashuatec DSC 38 Aficio	60	P_11
21	Laser	Canon LBP7750 cdb	60	P_12
22	Ink	Canon MX850	60	P_13
23	Ink	Canon MP630	60	P_14
24	Laser	Canon iR C2620	60	P_15
26	Ink	Canon MP64D	60	P_16
31	Laser	Hp Laserjet 4350 o.4250	60	P_17
32	Ink	Unknown	60	P_18
49	Laser	Hp Laserjet 5	60	P_19
50	Laser	Epson Aculaser C1100	60	P_20

Table 5. Metrics for classification evaluations

Metrix	Formula	Evaluation Focus
$Precision$	$\frac{TP}{TP+FP}$	Evaluate the positive patterns that are exactly calculated from the total predicted patterns in a positive class.
$Recall$	$\frac{TP}{TP+FN}$	Determine the fraction of positive patterns that are perfectly classified
$F-measure$	$\frac{2 \times recall \times precision}{recall + precision}$	Denotes the harmonic mean between recall and precision values
$Accuracy$	$\frac{TP+TN}{TP+TN+FN+FP}$	Estimates the proportion of exact predictions to all cases that were believed.

5.3. Experimental setup

This section describes the experimental setting and examines the outcomes. Three experiments are carried out. Evaluating the performance using 2D-CNN with SoftMax and 2D-CNN with SVM, evaluating the performance using pre-trained models as feature extraction, and evaluating the performance using pre-trained models as fine-tuning.

5.3.1. Performance of transfer learning via feature extraction

As stated in section 4.2, we use pre-trained CNN as feature extractors in our dataset. When pre-trained CNN are used as feature extractors, classification is done with an SVM. Table 6 summarizes the accuracy of feature extraction using pre-trained CNN. VGG-19 reports a maximum classification rate of 83.6%. Table 7 shows the performance of all pre-trained CNN models.

Table 6. The accuracy of feature extraction and classification using pre-trained CNN and SVM before and after pre-processing

Network	Accuracy before pre-processing	Accuracy after pre-processing
AlexNet	58.3%	72.8%
ResNet18	43%	72.2%
ResNet50	58.1%	82.8%
ResNet101	59.2%	76.7%
GoogleNet	28.3%	62.5%
VGG-16	68.3%	83.3%
VGG-19	69.2%	**83.6%**
DarkNet-19	69.4%	66.1%
DarkNet-53	59.2%	78.6%
SqueezeNet	44.7%	73.1%
XceptionNet	47.5%	69.2%
Inceptionv3	46.4%	70.8%
ShuffleNet	43.9%	69.2%

Table 7. Performance of pre-trained CNN models as feature extractors

Model	Accuracy	Recall	Precision	F1_score
AlexNet	72.8%	72.8%	75%	73.3%
ResNet-18	72.2%	72.2%	74.5%	72.4%
ResNet-50	82.8%	82.8%	83.4%	82.8%
ResNet-101	76.7%	76.7%	78.5%	76.7%
GoogleNet	62.5%	62.5%	65.3%	62.8%
VGG-16	83.3%	83.3%	84.3%	83.4%
VGG-19	**83.6%**	**83.6%**	**84.2%**	**83.7%**
DarkNet-19	66.1%	66.1%	67.8%	66.5%
DarkNet-53	78.6%	78.6%	80.4%	78.9%
SqueezeNet	73.1%	73.1%	74.7%	73.2%
ShuffleNet	69.2%	69.2%	70.4%	69.3%
Inceptionv3	70.8%	70.8%	72.8%	71.2%
Xception	69.2%	69.2%	70.3%	69.3%

5.3.2. Performance using 2D-CNN with SoftMax and 2D-CNN with SVM

The network architecture comprises four different depths to measure performance: 1, 2, 3, and 4 convolutional layers, with each convolutional layer provided with a ReLU layer and a pooling layer. For comparison, 7-layer, 10-layer, 13-layer, and 16-layer neural network models were created. The data sets are divided into 70% training and 30% testing. The network architecture is as follows: The 2D-CNN was trained up to 20 epochs with a batch size of 10 samples. The root mean square propagation (RMSProp) optimizer with a learning rate of 0.0001 was used. Each scanned document of samples is submitted for training and classification to the 7, 10, 13, and 16-layer CNN networks (details in Table 2), with the results displayed in Table 8. Results show that for 2D-CNN with SoftMax and 2D-CNN with SVM, 10-layer CNNs have the highest accuracy rate. The table also indicates that 2D-CNN with SVM is more accurate than 2D-CNN with SoftMax.

Table 8. Performance of 2D_CNN with SoftMax and SVM

Model		Accuracy	Recall	Precision	F1_score
2D-CNNwith SoftMax	1_conv (7 layers)	84.2%	84.2%	87.1%	93.3%
	2_conv (10 layers)	93.8%	93.8%	94.2%	93.7%
	3_conv (13 layers)	90.8%	90.8%	91.3%	90.9%
	4_conv (16 layers)	92.1%	92.1%	92.9%	91.9%
2D-CNN with SVM	1_conv (7 layers)	97.5%	97.5%	97.6%	97.5%
	2_conv (10 layers)	98.8%	98.8%	98.9%	98.8%
	3_conv (13 layers)	98.8%	98.8%	98.9%	98.7%
	4_conv (16 layers)	98.3%	98.3%	98.5%	98.4%

5.3.3. Performance of transfer learning via fine-tuning

As stated in section 4.4, we fine-tune pre-trained CNN models in our dataset. This method relied on deep transfer learning CNN architectures to transfer learning weights, which reduced training time, mathematical calculations, and hardware resource usage. As shown in Table 3, the pre-trained CNN models were modified in the last fully connected (The learning) layer and classification layers to meet the number of classes in the dataset, which consists of 20 classes. The accuracy of fine-tuning pre-trained CNN models is summarized in Table 9. The maximum classification rate reported by DarkNet-19 is 99.2 %. The performance of pre-trained CNN models as fine-tuning is shown in Table 10.

Table 9. The accuracy of pre-trained CNN via fine-tuning before and after pre-processing

Network	Accuracy before pre-processing	Accuracy after pre-processing
AlexNet	80.1%	90%
ResnNet18	87.5%	95.8%
ResNet50	91.7%	96.7%
ResNet101	89.2%	95.8%
GoogleNet	82.5%	90.8%
VGG-16	95.8%	96.7%
VGG-19	95%	96.7%
DarkNet-19	91.7%	99.2%
DarkNet -53	93.3%	97.5%
SqueezeNet	86.7%	95%
XceptionNet	91.7%	98.3%
Inceptionv3	90.8%	98.3%
ShuffleNet	90%	94.2%

Table 10. Performance of pre-trained CNN models as fine-tuning

Model	Accuracy	Recall	Precision	F1_score
AlexNet	90%	90%	91.3%	89.6%
ResNet-18	95.8%	95.8%	96.4%	95.7%
ResNet-50	96.7%	96.7%	97.1%	96.6%
ResNet-101	95.8%	95.8%	96.3%	96.6%
GoogleNet	90.8%	90.8%	92.3%	90.8%
VGG-16	97.5%	97.5%	97.7%	97.5%
VGG-19	97.5%	97.5%	97.9%	97.5%
DarkNet-19	99.2%	99.2%	99.3%	99.2%
DarkNet-53	95.5%	95.5%	97.7%	97.5%
SqueezeNet	95%	95%	95.4%	95%
ShuffleNet	94.2%	94.2%	95.2%	93%
Inceptionv3	98.3%	98.3%	98.6%	98.3%
Xception	98.3%	98.3%	98.6%	98.3%

5.3.4. Discussion

A new approach is proposed in this research to determine the printer's source. Although much research on source printer identification has been proposed, they have all been analysed using distinct datasets and experimental setups. As previously mentioned, many researchers use isolated characters in a text-dependent framework for experimental purposes. This is the first paper to use CNNs to identify the source printer without segmenting the document into characters, words, or patches and with small datasets. For VGG-19, 69.2% and 83.6% recognition rates were achieved using pre-trained CNN models as feature extraction before and after pre-processing, respectively. Recognition rates of 93.8% and 98.8% were achieved using the (2conv) 2D-CNN approach with SoftMax and SVM classifiers, respectively. The SVM classifier enhanced the 2D-CNN accuracy by roughly 5% over the initial configuration. DarkNet-19 achieved recognition rates of 91.7% and 99.2% utilizing pre-trained CNN models as fine-tuning before and after pre-processing, respectively. When fine-tuning DarkNet-19, the approach achieved an accuracy rate of 99.2% in several testing, as shown in Table 10.

5.3.5. Comparison with other techniques

This section compares the results of three models with some recent algorithms, including Elkasrawi *et al.* [37], CNN [14], KPNF+SURF+ORB [12], SURF and ORB with AdaBoost [16] and HOG+LBP with AdaBoost [18]. Comparison with related work on the dataset of 20 printers is highlighted in Figure 8. The proposed approaches using 2D-CNN with SVM transfer learning via fine-tuning DarkNet-19 outperform the previous five algorithms reported in the literature.

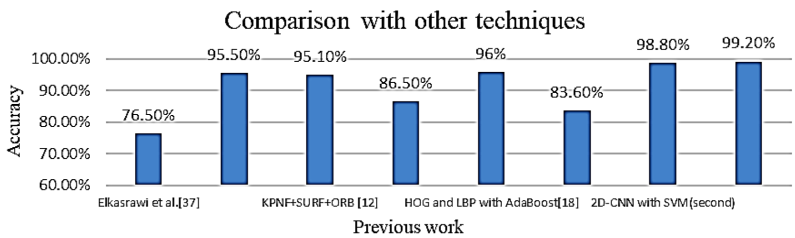


Figure 8. Comparison of classification accuracies between the three proposed approaches and existing methods

6. CONCLUSION

Based on source printer identification (SPI), this research proposes a new approach for identifying document forgeries. This approach trains three different models on the data set to choose the more accurate model. In the first model, a transfer learning for 13 pre-trained CNN models is used. These models are used as feature extractors, and SVM is used as a classifier. In this model, VGG-19 with SVM gives the best result with 83.6.5 % accuracy. In the second model, 2D-CNN was used to train from scratch. Then, this trained model was utilized for feature extraction rather than SoftMax. We used SVM as a classifier. The model achieved 98.5% accuracy for 2D-CNN and 93.75% accuracy for 2D-CNN-SVM in the studies. We tried the same 13 pre-trained models in the third model, but we fine-tuned them by retraining each model and modifying the last fully connected (The learning) layer. The fine-tuned DarkNet-19 gives the maximum classification rate of 99.2%. We conclude that transfer learning through fine-tuning is more accurate than the other two models based on the results of the three approaches. The second and third approaches' accuracy is better than five recently published algorithms that used the same dataset.

REFERENCES

[1] J. A. Lewis, "Forensic document examination defined," in *Forensic Document Examination*, Elsevier, 2014, pp. 35–44.

[2] A. Braz, M. López-López, and C. García-Ruiz, "Raman spectroscopy for forensic analysis of inks in questioned documents," *Forensic Science International*, vol. 232, no. 1–3, pp. 206–212, Oct. 2013, doi: 10.1016/j.forsciint.2013.07.017.

[3] P. C. Chu, B. Y. Cai, Y. K. Tsoi, R. Yuen, K. S. Y. Leung, and N. H. Cheung, "Forensic analysis of laser printed ink by X-ray fluorescence and laser-excited plume fluorescence," *Analytical Chemistry*, vol. 85, no. 9, pp. 4311–4315, Apr. 2013, doi: 10.1021/ac400378q.

[4] A. K. Mikkilineni *et al.*, "Printer forensics using SVM techniques," *International Conference on Digital Printing Technologies*, vol. 21, no. 1, pp. 223–226, Jan. 2005, doi: 10.2352/issn.2169-4451.2005.21.1.art00064_1.

[5] J. H. Choi, H. Y. Lee, and H. K. Lee, "Color laser printer forensic based on noisy feature and support vector machine classifier," *Multimedia Tools and Applications*, vol. 67, no. 2, pp. 363–382, Jul. 2013, doi: 10.1007/s11042-011-0835-9.

[6] S. Shang, N. Memon, and X. Kong, "Detecting documents forged by printing and copying," *Eurasip Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–13, Sep. 2014, doi: 10.1186/1687-6180-2014-140.

[7] M. J. Tsai, J. S. Yin, I. Yuadi, and J. Liu, "Digital forensics of printed source identification for Chinese characters," *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 2129–2155, Sep. 2014, doi: 10.1007/s11042-013-1642-2.

[8] M. J. Tsai and I. Yuadi, "Digital forensics of microscopic images for printed source identification," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 8729–8758, May 2018, doi: 10.1007/s11042-017-4771-1.

[9] M. J. Tsai, C. L. Hsu, J. S. Yin, and I. Yuadi, "Digital forensics for printed character source identification," in *Proceedings - IEEE International Conference on Multimedia and Expo*, Jul. 2016, vol. 2016-Augus, doi: 10.1109/ICME.2016.7552892.

[10] H. Jain, S. Joshi, G. Gupta, and N. Khanna, "Passive classification of source printer using text-line-level geometric distortion signatures from scanned images of printed documents," *Multimedia Tools and Applications*, vol. 79, no. 11–12, pp. 7377–7400, Dec. 2020, doi: 10.1007/s11042-019-08508-x.

[11] S. Joshi and N. Khanna, "Single classifier-based passive system for source printer classification using local texture features," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1603–1614, Jul. 2018, doi: 10.1109/TIFS.2017.2779441.

[12] S. Gupta and M. Kumar, "Forensic document examination system using boosting and bagging methodologies," *Soft Computing*, vol. 24, no. 7, pp. 5409–5426, Aug. 2020, doi: 10.1007/s00500-019-04297-5.

[13] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. C. Chiu, J. P. Allebach, and E. J. Delp III, "Printer identification based on graylevel co-occurrence features for security and forensic applications," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, Mar. 2005, vol. 5681, p. 430, doi: 10.1117/12.593796.

[14] M. Bibi, A. Hamid, M. Moetesum, and I. Siddiqi, "Document forgery detection using printer source identification-a text-independent approach," in *2019 International Conference on Document Analysis and Recognition Workshops, ICDARW 2019*, Sep. 2019, vol. 2019-Janua, pp. 7–12, doi: 10.1109/ICDARW.2019.70134.

[15] A. Ferreira *et al.*, "Data-driven feature characterization techniques for laser printer attribution," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1860–1873, Aug. 2017, doi: 10.1109/TIFS.2017.2692722.

[16] M. Kumar, S. Gupta, and N. Mohan, "A computational approach for printed document forensics using SURF and ORB features," *Soft Computing*, vol. 24, no. 17, pp. 13197–13208, Jan. 2020, doi: 10.1007/s00500-020-04733-x.

[17] P. J. Chiang *et al.*, "Printer and scanner forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 72–83, Mar. 2009, doi: 10.1109/MSP.2008.931082.

[18] N. F. E. Abady, M. Taha, and H. H. Zayed, "Text-independent algorithm for source printer identification based on ensemble learning," *Computers, Materials and Continua*, vol. 73, no. 1, pp. 1417–1436, 2022, doi: 10.32604/cmc.2022.028044.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[20] X. Yu, J. Wang, Q. Q. Hong, R. Teku, S. H. Wang, and Y. D. Zhang, "Transfer learning for medical images analyses: A survey," *Neurocomputing*, vol. 489, pp. 230–254, Jun. 2022, doi: 10.1016/j.neucom.2021.08.159.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 2, pp. 1097–1105, 2012.

[22] K. S. and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations*, 2015, doi: 10.48550/arXiv.1409.1556.

[23] C. Szegedy *et al.*, "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[26] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[28] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.

[29] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 6848–6856, doi: 10.1109/CVPR.2018.00716.

[30] D. Larsen-Freeman, "Transfer of learning transformed," *Language Learning*, vol. 63, no. SUPPL. 1, pp. 107–129, Feb. 2013, doi: 10.1111/j.1467-9922.2012.00740.x.

[31] M. J. Tsai, Y. H. Tao, and I. Yuadi, "Deep learning for printed document source identification," *Signal Processing: Image Communication*, vol. 70, pp. 184–198, Feb. 2019, doi: 10.1016/j.image.2018.09.006.

[32] C. B. Rabah, G. Coatrieux, and R. Abdelfattah, "Automatic source scanner identification using 1D convolutional neural network,"

Multimedia Tools and Applications, vol. 81, no. 16, pp. 22789–22806, Jun. 2022, doi: 10.1007/s11042-021-10973-2.
[33] W. K. Pratt, "Digital image processing," *European Journal of Engineering Education*, vol. 19, no. 3, p. 377, Jan. 1994, doi: 10.1080/03043799408928319.
[34] Anjani Suputri Devi D and Satyanarayana Ch, "An efficient facial emotion recognition system using novel deep learning neural network-regression activation classifier," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 17543–17568, Feb. 2021, doi: 10.1007/s11042-021-10547-2.
[35] W. Tao, M. Al-Amin, H. Chen, M. C. Leu, Z. Yin, and R. Qin, "Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing," *Procedia Manufacturing*, vol. 48, pp. 926–931, 2020, doi: 10.1016/j.promfg.2020.05.131.
[36] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification using sensor pattern noise," in *Security, Steganography, and Watermarking of Multimedia Contents IX*, Feb. 2007, vol. 6505, p. 65051K, doi: 10.1117/12.705837.
[37] S. Elkasrawi and F. Shafait, "Printer identification using supervised learning for document forgery detection," in *Proceedings - 11th IAPR International Workshop on Document Analysis Systems, DAS 2014*, Apr. 2014, pp. 146–150, doi: 10.1109/DAS.2014.48.
[38] Y. Zhang, J. Goh, L. L. Win, and V. Thing, "Image region forgery detection: A deep learning approach," *Cryptology and Information Security Series*, vol. 14, pp. 1–11, 2016, doi: 10.3233/978-1-61499-617-0-1.
[39] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020.

BIOGRAPHIES OF AUTHORS

Naglaa F. El Abady received the M.Sc. degree in computer science at the Faculty of Computers and Artificial intelligence, Benha University, Egypt, in 2015. She is currently an assistant lecturer at the computer science department, Faculty of Computers and Artificial intelligence, Benha University, Egypt. Currently, she is working on his Ph.D. degree. Her research interest's concern: digital forensics (document forgery detection), security (cryptography), and image processing. She can be contacted at email: naglaa.fathy@fci.bu.edu.eg.

Hala H. Zayed received her B.Sc. in electrical engineering (with an honor degree) in 1985, her M.Sc. in 1989, and her Ph.D. in 1995 from Benha University in electronics engineering. She is now a professor at the Information Technology and Computer Science faculty, Nile University, Egypt. Her research areas are computer vision, biometrics, machine learning, and image processing. She can be contacted at email: hala.zayed@fci.bu.edu.eg or hhelmy@nu.edu.eg.

Mohamed Taha is an Associate Professor at Benha University, Faculty of Computers and Artificial intelligence, Computer Science Department, Egypt. He received his M.Sc. degree and his Ph.D. degree in computer science at Ain Shams University, Egypt, in February 2009 and July 2015. He is the founder and coordinator of the "Networking and Mobile Technologies" program, Faculty of Computers and Artificial Intelligence, Benha University. His research interest's concern: computer vision (object tracking-video surveillance systems), digital forensics (image forgery detection–document forgery detection-fake currency detection), image processing (OCR), computer networks (routing protocols-security), augmented reality, cloud computing, and data mining (association rules mining-knowledge discovery). Taha has contributed more than 20+ technical papers in international journals and conferences. He can be contacted at email: mohamed.taha@fci.bu.edu.eg.