Expert Systems With Applications 238 (2024) 121603

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hierarchical deep learning approach using fusion layer for Source Camera Model Identification based on video taken by smartphone

Younes Akbari [a,*], Somaya Al Maadeed [a], Omar Elharrouss [a], Najmath Ottakath [a], Fouad Khelifi [b]

[a] Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[b] Department of Computer and Information Sciences, Northumbria University, UK

ARTICLE INFO

Keywords:
Source model camera identification
Video
Hierarchical deep learning
Fusion layer
Joint sparse representation

ABSTRACT

Over the last decade, videos uploaded and shared through web-based multimedia platforms and mobile applications have proliferated worldwide. This is because cloud-based applications such as iCloud, YouTube, Facebook, Twitter, and WhatsApp offer affordable and secure environments for video storage and sharing. However, new challenges have emerged alarming forensic analysts and investigators since videos can be used to commit heinous crimes such as blackmail, fraud, and forgery. Source Camera Identification (SCI) has become of paramount importance in the field of image and video forensics. Camera model identification can also help identify the perpetrators or narrow down the search and can be used to enhance SCI systems. In this context, existing approaches such as the Photo Response Non-Uniformity (PRNU) based methods and machine learning techniques such as the support vector machine (SVM) and deep learning models are commonly used solutions. This work exploits these two categories of methods by exploring a hierarchical deep learning model for camera model identification based on smartphone videos. The PRNU features are extracted by CNN-based structures during the training process. Proposed six-stream networks are leveraged to extract both low-level and high-level features through the network. A fusion layer is created based on joint sparse representation using forward and backward functions defined for fusing the proposed six streams. The proposed approach has been implemented and evaluated through intensive experiments, and results showed successful camera model identification with a performance at the frame level reaching an average accuracy of 69.9% for the Daxing dataset and 81.6% for the QUFVD dataset.

1. Introduction

Mobile phones have been one of the most successful technologies ever introduced and adopted worldwide. This is because, unlike many other technologies, mobile phones have multiple uses and multiple purposes due to a variety of social and economic needs across different countries and regions (Tian et al., 2019). However, they can be used for malicious purposes. Smartphone devices provide critical information for forensic investigations and criminal prosecutions (Akbari et al., 2022; Tian et al., 2019). Forensic experts have been particularly intrigued by this subject in recent times due to the increasing number of crimes committed through videos. Medical, legal, and surveillance systems require reliable and authentic information to be shared through multiple sources. Investigations performed in cases of anomalous activities through these video sources or for the purpose of cataloging are required to be accurate.

Video source identification through existing techniques can be compromised through lossy compression, which can complicate forensic analysis. High compression rates can significantly damage the evidential traces and thus make it impossible or difficult to recover the traces of the data and its source (Ahmed, Khelifi, Lawgaly, & Bouridane, 2019; Kang, Li, Qu, & Huang, 2011; Lawgaly & Khelifi, 2016; López, Orozco, & Villalba, 2021). The forensic analysis of videos in this regard has been less explored (Akbari, Al-maadeed, Elharrouss et al., 2022). Although several methods have been successfully implemented on images, they cannot be directly applied to videos (Altinisik & Sencar, 2020; Iuliani, Fontani, Shullani, & Piva, 2019; Mandelli, Bestagini, Verdoliva, & Tubaro, 2019). Compression, stabilization, scaling, and cropping are further challenges identified for video source identification. Video identification algorithms can identify and distinguish the camera types by analyzing digital videos.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: younes.akbari@qu.edu.qa (Y. Akbari), s_alali@qu.edu.qa (S. Al Maadeed), elharrouss.omar@gmail.com (O. Elharrouss), no1912348@student.qu.edu.qa (N. Ottakath), fouad.khelifi@northumbria.ac.uk (F. Khelifi).

https://doi.org/10.1016/j.eswa.2023.121603
Received 20 April 2023; Received in revised form 11 September 2023; Accepted 11 September 2023
Available online 19 September 2023

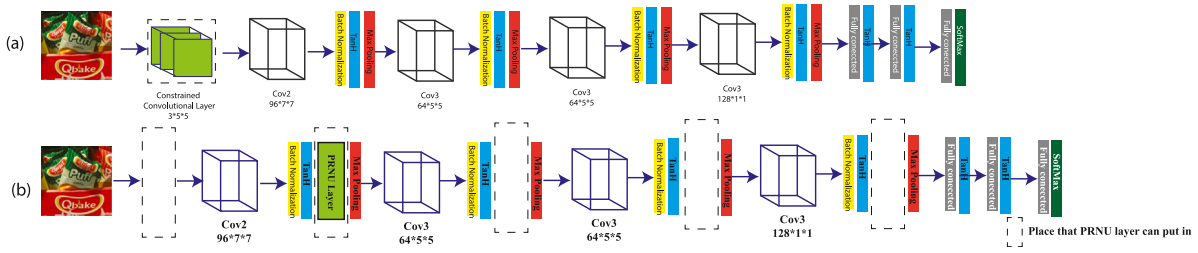

Fig. 1. Overview of (a) The CNN (MISLnet) presented in Bayar and Stamm (2018) with a constrained layer in the first layer of the network, (b) PRNU-Net presented in Akbari, Almaadeed, Al-Maadeed et al. (2022) with the PRNU layer of the network.

In general, there are two concepts in the field: Individual Source Camera Identification (ISCI) (Lawgaly, Khelifi, Bouridane, Al-Maaddeed, & Akbari, 2022a; Lawgaly, Khelifi, Bouridane, Al-maadeed, & Akbari, 2022b) and Source Camera Model Identification (SCMI) (Akbari, Al-maadeed, Al-Maadeed, Khelifi & Bouridane, 2022; Villalba, Orozco, López, & Castro, 2016). ISCI is able to distinguish between cameras of the same or different models, whereas SCMI is a subset of ISCI that distinguishes only a specific camera model from other models, but not between different cameras of the same model. SCMI can be a step toward improving source camera identification, and in some cases, model identification can be a sufficient step (Villalba et al., 2016).

The source of images and videos can typically be identified in two ways: by extracting a unique fingerprint from the images or videos, or by analyzing the metadata stored on the images or videos, which is also called the DNA of the video. Although (López, Luengo, Orozco, & Villalba, 2020) proved that the metadata and the internal elements of the video can be used for source video identification but metadata manipulation can easily take place which compromises the reliability of the approach. Extracting noise patterns built into the camera for source identification through unique patterns is one of the methods used. The PRNU (Chuang, Su, & Wu, 2011; Lawgaly et al., 2022a, 2022b; Mahalanobis, Kumar, & Casasent, 1987) in specific is the unique fingerprint of the camera, often referred to as residual noise or sensor pattern noise (SPN). The existence of PRNU is caused by the CCD (charge coupled device) or CMOS (Complementary Metal Oxide Semiconductor) converting input signal that is the light signal, to a digital signal. The PRNU generated is a low-level feature and is unique to each camera. Deep learning approaches are the other methods used for ISCI and SCMI. The fingerprint of the video is extracted during the training process.

The challenge in deep learning approach is amounted to the difficulty in separating the desirable noise (PRNU) from the video (Akbari, Al-maadeed, Elharrouss et al., 2022). This is typically solved by devising algorithms on this specific task such as exploring new architectures and new loss functions. A popular architecture for this purpose was MISLnet (Multimedia and Information Security Lab) (Bayar & Stamm, 2018) which is based on a constrained convolutional layer. As indicated in Fig. 1(a), a constrained convolutional layer is inserted at the beginning of a CNN that will execute the forensic tasks. Low-level characteristics are extracted as a result of the layer to conceal the image content. Despite the promising results of the method (Hosler et al., 2019; Timmerman, Bennabhaktula, Alegre, & Azzopardi, 2020), and because of the degree of sensitivity for camera model identification problem, an improvement in the field is always essential. Another method used in the field is based on adding a PRNU layer to the CNN (Akbari, Almaadeed, Al-Maadeed et al., 2022) as shown in Fig. 1(b). In this method, instead of the constrained convolutional layer, a new layer is defined that extracts the PRNU of each input. In the two models, low-level features are extracted by adding a new layer to the normal CNNs. The two models obtained promising results in the field. However, the layer was only added at one location in the models.

To improve the two structures, a hierarchical approach as shown in Fig. 2 is employed in our proposed method where high-level and low-level features are fused by defining a fusion layer before the Softmax layer. The input of the fusion layer is six streams (each stream is a network that begins with the input layer and its output is fed into the fusion layer) improving the results significantly. This is considered a multi-featured problem. The fusion layer is based on the joint sparse representation method. Fingerprint features are extracted using either a constrained convolutional layer or a PRNU layer placed in between the convolutional layers. Low-level feature representation is achieved using the constrained layer or the PRNU layer. Fusion of the features after fully connected layers is an added advantage where at-least two streams are required to be fused. Fingerprint information extracted from the frames by passing it through streams produce low level features that are extracted from consecutive layers which further produce, high, mid and low level features. For evaluation and optimization of the structure for better performance, the constrained layer or the PRNU layer is placed at different locations of the network and an ablation study with varying numbers of streams is performed. Forward function is based on the joint sparse representation method and the derivative loss identified is utilized for backpropagation with respect to the input data of the layer. Evaluating the approach requires that the frames are extracted from the video. Two datasets, Qatar University Forensic Video Database (QUFVD) (Akbari et al., 2022) and Daxing database (Tian et al., 2019) are evaluated on in this approach.

Following is a summary of the main contributions of this paper:

- *A CNN-based hierarchical structure:* For efficient multi-level feature extraction in source model camera identification, a CNN-based hierarchical structure is presented.
- *Exploring a fusion approach:* A sparse representation based fusion method that combines the extracted deep features.

The paper is structured as follows. Section 2 presents a review of available deep learning methods for source camera verification from videos and fusion methods. The new approach is presented in Section 3. Section 4, evaluates the proposed approach with the subsequent section concluding the work.

2. Related work

The classification of methods used for identifying the source of videos primarily involves two categories, namely PRNU and Deep Learning techniques. The following section details the deep learning based approaches as it is the prime network used in the proposed architecture. Sparse representation approach is also delved into as a fusion method and the state-of-the-art in these approaches are also explored.

2.1. Deep learning methods

Deep learning methods in the literature involve noise extraction using signal analysis and modifying the traditional convolutional network. Modification is performed by adding modified filters that target

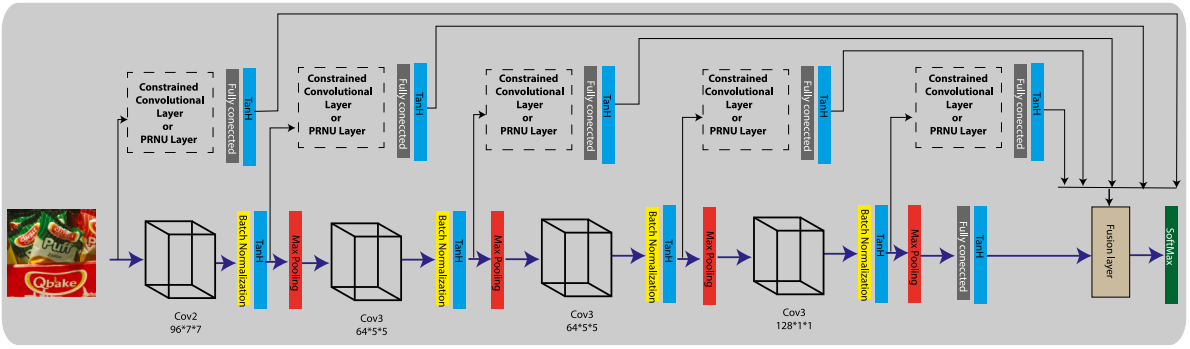

Fig. 2. Overview of Proposed hierarchical network structure in six streams with a defined new fusion layer.

specific regions and features of the image leading to an enhanced feature description of the video. The signal analysis method was employed in Kirchner and Johnson (2019) called SPN-CNN where a CNN based sensor pattern noise (SPN) was used to extract signals characterized by noise from a set of frames (Zhang, Zuo, Chen, Meng, & Zhang, 2017). This was used to extract a noise pattern which was tested on VISION database (Shullani, Fontani, Iuliani, Al Shaya, & Piva, 2017), outperforming the usual wavelet denoiser. The advantage of this method was that the input frame was I-frame which significantly improved the identification compared to the previous state-of-art.

With the base in MISLnet architecture several state-of-art have modified the structure using different representations of the network to achieve better convergence in the deep learning network. MISLnet for source camera identification using video frames to train the network was proposed in Hosler, Mayer, Bayar et al. (2019), Timmerman et al. (2020) with an extended version of Constrained convolutional layer in Bayar and Stamm (2018). A majority voting is performed on the frames to decide video level classifications in the network. The constrained convolutional layer is the initial layer in the network with kernel size 5 which is constructed in such a way that the relationships between pixels are independent of the content of the scene. This method was tested on the VISION database (Shullani et al., 2017). The results indicated that the deep learning model with the constrained layer showed a significant improvement compared to the setup without the constraint layer. The noted difference in them was the size of the frames and types of color modes available. While authors in Timmerman et al. (2020) used RGB color mode, Hosler, Mayer, Bayar et al. (2019) used gray-scale mode for source camera identification. Patch based analysis was performed in these approaches with a patch size of 480×800.

Further experimentation on the CNN based approach in Bayar and Stamm (2018) was performed in Mayer, Hosler, and Stamm (2020). The similarity network maps two input deep feature vectors to a 2D similarity vector. To achieve this, the authors follow a design of similarity network developed in Mayer and Stamm (2019). The fusion approach presented is based on the mean of the inactivated output layer from the similarity network which produces a decision for the whole video. While tested on SOCRatES dataset (Galdi, Hartung, & Dugelay, 2019) showed that the method improved compared to traditional methods such as Goljan, Fridrich, and Filler (2009). Fig. 1(a) shows the structure of the CNN used in the three studies proposed here, with a constrained convolutional layer added to the simple CNN.

2.2. Fusion methods

Multi-feature fusion enables the discovery and correlation of features across different views. It extracts complimentary and complete information for the given task. It can also identify similarity across different features through multi-feature fusion. Based on the existing state-of-art the types of multi-feature fusion methods are categorized as follows: multi-kernel learning, which learns from a predefined set of kernels (Li, Zhang, Lu, & Zhang, 2019; Shao, Liu, & Yu, 2016); subspace learning, where the aim is to identify a generalized linear subspace, which includes dimension normalization and subspace projection by maximizing the cumulative pairwise canonical correlation constructed from each pair of the resultant feature set (Kan, Shan, Zhang, Lao, & Chen, 2015; Wang, Arora, Livescu, & Bilmes, 2015); and Sparse representation approach, where an entity is represented in its minimum possible non-zero coefficients (Abavisani & Patel, 2018; Bahrampour, Nasrabadi, Ray, & Jenkins, 2015).

Joint sparse representation of the obtained features is utilized in our study and thus this review points its focus on state-of-art in this domain.

The task of approximating the least dictionary atoms utilizing a sparse representation matrix is the most popular approach existing in joint sparse representation as in Abavisani and Patel (2018), Akbari, Elharrouss and Al-Maadeed (2022), Bahrampour et al. (2015), Li et al. (2017). Another approach presented by Yang et al. Yang, Zhang, Zhang, and Wang (2012) was a relaxed collaborative representation (RCR) where different features represented a common coefficient. The sum of distance of coefficients from their average was identified to minimize the sparse codes.

Further, Yuan et al. in Yuan, Liu, and Yan (2012), produced a joint sparse representation for the multi-features (MTJSRC) using the l_1, l_2 norm. A high-dimensional data was successfully tested on this method. In Li, Zhang and Zhang (2017), similar and discriminative coefficients were obtained by multi-feature fusion resultant of a join discriminative collaborative representation (JDCR) approach. The joint feature extraction which aligns with multi-feature groups were introduced by a feature selection method in Gui, Tao, Sun, Luo, You, and Tang (2014) for the purpose of dimensionality reduction. A partial multi-view clustering (PVC) was another approach utilized in Li, Jiang, and Zhou (2014) where incomplete data was presented. The latent sub-space was learned from process of non-negative matrix factorization (NMF) (Lee & Seung, 1999). Applying sparse representation for multi-modal features, authors in Bahrampour et al. (2015), Li et al. (2017) utilized a sparse representation model based on dictionary learning. Health data, in-specific, diabetes mellitus and impaired glucose regulation problems were represented through specific and similar components extracted as multiple features using joint sparse representation producing desirable results as reported in Li, Zhang, Li, Wu and Zhang (2017). The review proves efficiency of join sparse representation for representing multiple features.

3. Hierarchical deep learning approach

Efficiency of multi-feature learning through joint sparse representation was utilized in the framework proposed, as shown in Fig. 2. The structure of the network proposed is presented in the figure. The succeeding subsection details structural components of the approach implemented in this paper.

3.1. Structure of the network

Although single-stream CNNs can produce good results when using lower-level data such as contours and edges, multiple streams can provide more useful information (Vo, Kim, Yang, & Lee, 2018). Evidence shows that results based on CNNs can be significantly improved when both low-level features and high-level features are present in multiple data streams (Vo et al., 2018). With multi-level feature representation taken into consideration, the constrained layer proposed by Bayar and Stamm (2018) (MISLnet architecture) and PRNU layer by Akbari, Almaadeed, Al-Maadeed et al. (2022) are placed at varied regions of the network. Constrained convolutional layer filters in MISLnet are created with the following limitations:

$$\begin{cases} \omega_{k_j}^{(1)}(0,0) = -1 \\ \sum_{p,q\neq 0} \omega_{k_j}^{(1)}(p,q) = 1 \end{cases} \tag{1}$$

where p and q show the entry in the pth row and qth column of the filter ω ($\omega(0,0)$ is the center of the filter ω) and $j = \{1,2,3\}$. $\omega_{k_j}^{(1)}$ denotes the jth kernel of the kth filter in the first layer of the network.

Also, PRNU-Net uses a defined layer based on PRNU. To define the layer, consider $B = \left\{ X_{(l)}^{(j)}, Y^j \right\}_{j=1}^{N}$ as training set with N samples. l is the position of the layer as shown in Fig. 1(b). For each input of the layer, let $X_{(l)}^{(j)} = \{x_1, x_2, \ldots, x_d\}$, where d is the dimension of the input of the layer. For the approach, for $l = \{1,2,3,4,5\}$, d can be $d = \{1, 96, 64, 64, 128\}$, respectively. Therefore, PRNU can be extracted from raw images (input layer) and feature maps of the convolutional layers. To obtain PRNU of the input of the layer as (Goljan et al., 2009):

$$x_i = O + OK + \Theta \tag{2}$$

Where O refers to the original input multimedia file, K represents the PRNU factor and Θ is a random noise factor. To estimate K, noise residual W of the input should be obtained using denoising filter F:

$$W_i = x_i - F(x_i) \tag{3}$$

Estimation of K is obtained by the following maximum likelihood estimator:

$$\hat{K}_i = \frac{W_i x_i}{(x_i)^2} \tag{4}$$

Where $\hat{K}_i$ is the output of the layer for input x_i.

Six streams introduce one of the two layers, the constrained layer or the PRNU layer. A constrained convolution or a PRNU layer, a fully connected layer, and a tanh activation function comprise the first stream. The resulting features are fed into the fusion layer in the form of a feature set. Following each convolutional layer, the remaining streams contain either the constrained layer or the PRNU layer. The final stream is either the non-constrained or the PRNU layer. The fusion layer receives the features of all the streams, resulting in a fused multi-feature space.

3.2. Fusion layer

The multiple streams approach produces low, mid and high-level features improving the results considerably in different domains (Vo et al., 2018). Simple fusion methods like concatenation and addition operations also produce significant results as shown in our results section. To further enhance the methodology, a multi-feature joint sparse representation is utilized. The traditional convolutional layers with a fully connected layer are considered as a mapping from sparse to dense. Joint sparse representation (dense to sparse) can be useful for fusion methods (Ahmad & Scheinkman, 2019; Yu & Gao, 2020). It is found to distinguish the feature space in multi-modal problems efficiently (Bahrampour et al., 2015; Cotter, Rao, Engan, & Kreutz-Delgado, 2005; Li & Zhang, 2021; Zhang, Zhang, Nasrabadi, & Huang, 2012). The implementation of the fusion method is referenced based on unsupervised multi-modal dictionary learning as presented in Bahrampour et al. (2015).

In order for the convergence of deep learning model through fusion layers, the forward and backward functions are defined. The process is detailed in the subsequent sections.

3.2.1. Forward function

Let $FC = [1, \ldots, S]$ is as a finite set with S streams (S can be at least two streams) and $ST_s = \left\{ X_{(s)}^{(j)}, Y^j \right\}_{j=1}^{N}, s \in FC$ is output of fully connected layers with N samples in each stream. A dictionary is used to represent each data stream. $D^s \in \mathbb{R}^{n^s \times d}$ (n^s and d are the number of samples and the number of dictionary atoms in stream s, respectively) by the method for addressing the fusion layer.

Therefore, we can have multi-stream dictionaries constructed by the collection of data extracted from different streams in the network. Based on the number of dictionary atoms, the collection of data extracted is selected. To obtain the optimized dictionary, the selection continues for all ST_s set. For ST_s, we can solve the ι_{12}-regularized reconstruction problem to obtain the optimal code sparse matrix $A^* \in \mathbb{R}^{d \times FC}$ (output of our fusion layer):

$$l(X, D) \doteq \min_{A[\alpha^1 \ldots \alpha^S]} \frac{1}{2} \sum_{s=1}^{S} \left\| X_s - D^s \alpha^s \right\|_{\iota_2}^2 + \lambda_1 \left\| A \right\|_{\iota_{12}} + \frac{\lambda_2}{2} \left\| A \right\|_F^2, \tag{5}$$

where λ_1 and λ_2 are the regularizing parameters. For the joint sparse optimization problem, the Frobenius norm $\|A\|_F$ term is added to obtain a unique solution (Bahrampour et al., 2015). Here, α^s is the sth-column of A which shows the sparse representation for the sth stream. The ι_2 norm of a vector $v \in \mathbb{R}^m$ and the ι_{12} norm of matrix $V \in \mathbb{R}^{m \times n}$ are defined as $\|v\|_{\iota 2} = (\sum_{j=1}^{m} \left| v_j \right|^2)^{1/2}$ and $\|V\|_{\iota 12} = \sum_{i=1}^{m} \left\| v_{i\rightarrow} \right\|_{\iota_2}$ ($v_{i\rightarrow}$ is the ith row of matrix), respectively. There are several algorithms to solve the optimization problem (Rakotomamonjy, 2011), such as the efficient multiplier method (ADMM) (Parikh, Boyd, et al., 2014), which can be used for finding A^*. Multi-stream dictionaries are obtained by the optimization problem:

$$D^{s*} = \underset{D^s \in \mathbb{D}}{\arg\min}\ E_{X_s}[l(X_s, D^s)],\ \forall s \in FC \tag{6}$$

assuming that X is drawn from a finite probability distribution $p(X)$, which is normally unknown, and $E_{X_s}[.]$ is the expectation operator with respect to the distribution $p(X)$ and the convex set $\mathbb{D}$ is defined as:

$$\mathbb{D}^s \doteq \left\{ D \in \mathbb{R}^{n^s \times d} \mid \left\| d_k \right\|_{\iota_2} \leq 1,\ \forall k = 1, \ldots, d \right\}. \tag{7}$$

It is assumed that data X_s are drawn from a finite (unknown) probability distribution $p(X_s)$. A classical projected stochastic gradient algorithm (Aharon & Elad, 2008) can be used to solve the above optimization problem, yielding a sequence of updates for each iteration:

$$D^s \leftarrow \Pi_{\mathbb{D}^s}[D^s - \rho_t \nabla_{D^s} l(X_s^t, D^s)], \tag{8}$$

where ρ_t and $\Pi_{\mathbb{D}}$ are the gradient step at time t, and the orthogonal projector on the set $\mathbb{D}$, respectively. As shown in Aharon and Elad (2008), Bottou (2010), for a decreasing sequence of ρ_t the algorithm converges to a stationary point. Due to the non-convexity of the optimization problem, it is not guaranteed that it converges to a global minimum (Bottou & Bousquet, 2007; Mairal, Bach, Ponce, & Sapiro, 2010). Nevertheless, practical applications have shown that a stationary point of this type is sufficient (Elad & Aharon, 2006; Mairal, Elad, & Sapiro, 2007).

3.2.2. Backpropagation

The forward function generates input and output for each layer, which is essential to the backpropagation of the fusion layer. With a user-defined layer, the output of the previous layer is fed to the forward function. The input size of the forward function and the output size of the backward function should be the same. The derivative of the loss with respect to the input data (ST_s) is:

$$\frac{\partial L}{\partial ST_s} = \frac{\partial L}{\partial f(ST_s)} \frac{\partial f(ST_s)}{\partial ST_s} \tag{9}$$

where $\frac{\partial L}{\partial f(ST_s)}$ is the resultant gradient propagated from the previous layer. As the both input and output of the forward function are used in backward propagation to derive the derivative of the activation, considering $\hat{A}^*$ as:

$$\hat{A}^* = E \odot ST_s \tag{10}$$

where $\hat{A}^*$ shows a vector that performs the operation to obtain fused features like (4), and $\odot$ is the element-wise product of the two vectors. Then, if we have $f(ST_s) = \hat{A}^*$, the derivation is:

$$\frac{\partial f(ST_s)}{\partial ST_s} = E \tag{11}$$

E is obtained after deriving $\hat{A}^*$ and ST_s through backpropagation operation. Utilizing the same size in backpropagation results in three options for the selection of the size of dictionary atoms in the fusion model. If the size of the ST_s for atoms is too small, it can result in an underdetermined system, whereas if it is too large, it can result in an overdetermined system. This resultant vector A^* should be calculated to be the same size as previous. The condition to stop the process is that the atoms equals the size of ST_s. The vector A^* with the same size as ST_s can also be achieved through interpolation and extrapolation methods. To elaborate on the problem, we consider the size of the atoms in the dictionary as equal to ST_s.

Algorithm 1 enumerates the learning schemes for forward function and backpropagation.

Algorithm 1 Training fusion method with forward and backward functions.

function FORWARD(ST_s, d)
  for j=1 to N
    Compute D^s using (6), (7), (8)
    Compute A^* using (5)
  end for
  return A^*
end function
function BACKWARD(A^*, ST_s, $\frac{\partial L}{\partial f(ST_s)}$)
  if $d > sizeof\ ST_s$
    Update A^* using methods to solve an underdetermined
    system or extrapolation methods
  else $d < sizeof\ ST_s$
    Update A^* using methods to solve an overdetermined
    system or interpolation methods
  end if
  Compute vector of fused features operation:
  $E = \hat{A}^* \oslash ST_s$
  ($\oslash$ is the element-wise division)
  Compute $\frac{\partial f(ST_s)}{\partial ST_s}$ using (8)
  return $\frac{\partial L}{\partial ST_s} = \frac{\partial L}{\partial f(ST_s)} E$
end function

3.3. Time complexity

Since our CNN architecture has six streams and each stream has fully connected layers followed by a fusion layer, and in one implementation, we have PRNU layer, the time complexity of the forward pass for the layers is:

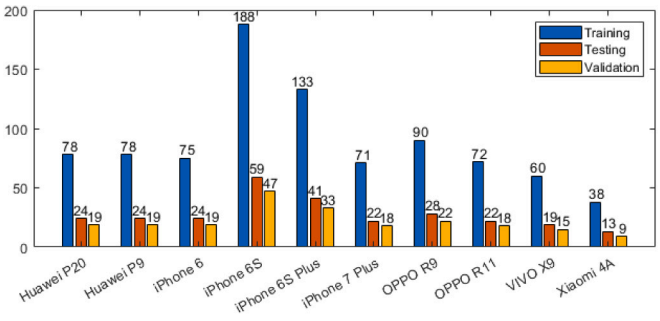

Fig. 3. The number of videos related to the training, testing, and validation sets for the Daxing database.

Convolutional Layers: O (N.K.K.C.H.W), where N is the number of input samples, K is the kernel size, C is the number of input channels, and H and W are the height and width of the feature map. The parameters are mentioned in Fig. 2.

Fully Connected Layers: O (N.L), where L is the number of neurons in the fully connected layer. The fully connected neurons based on the best result is 100.

PRNU layers: The time complexity of this layer is the number of input samples used for noise extraction (N) and the size of each input (M), which can be approximated as O (N.M).

Fusion Layer: Based on Algorithm 1, computing dictionary learning and fusion features should be considered. Dictionary learning involves iteratively updating the dictionary and the sparse representations of the input data. The time complexity for a single iteration can be approximated as O (T.N.D.K^2), where T, N, D, K are number of iterations, number of input samples, dimensionality of the data (size of the input), number of atoms in the dictionary, respectively. In our implementation,$T = 20$ (the same (Bahrampour et al., 2015)), K = 600 (total of output of neurons of fully connected layer of each stream where is 100).

During the backward pass (backpropagation), gradients are computed with respect to the network's parameters. The backward pass's time complexity is generally similar to the forward pass because it also involves propagating gradients through the layers. In our implementation for fusion layer, the process is done by using element-wise division operation for obtaining the gradient.

4. Databases

The databases available for source video identification are recordings captured from video cameras. There are two databases that offer the recording with smartphones, they are namely Daxing[1] (Tian et al., 2019) and QUFVD[2] (Akbari et al., 2022). Experimenting on methods that are based on smartphone databases that are recently developed can show its significance in the current domain. Comparing the results of Daxing and QUFVD with older databases such as VISION, has shown that the latest smartphone-based databases are more challenging and thus require extensive analysis and improvement (Akbari et al., 2022).

Daxing smartphone identification database includes both images and videos captured from different smartphones of different brands and models. The database contains data from 90 smartphones which are from 22 models and 5 brands. A total of 43400 images and 1400 videos are in the database. The number of training videos is small and may differ across the devices. This unbalanced data makes it

[1] https://github.com/xyhcn/Daxing

[2] https://www.dropbox.com/sh/nb543na9qq0wlaz/AAAc5N8ecjawk2KlVF8kfkrya?dl=0

Table 1
The results of the frame and video levels in terms of overall accuracy (%) in terms of QUFVD and Daxing databases.

Database	I-frame				Video			
	MISLnet	Ours	PRNU-Net	Ours	MISLnet	Ours	PRNU-Net	Ours
QUFVD	74.5	80.8	77.6	**81.6**	82.0	88.7	85.7	**89.4**
Daxing	62.7	67.6	64.6	**69.9**	66.7	73.2	69.6	**73.5**

Fig. 4. Sample frames from captured videos of the databases (QUFVD is on the first row, and Daxing is on the second row).

even more challenging for machine learning and deep learning models. Although there are several solutions to address data scarcity issues such as Self-Supervised Learning (SSL) (Doersch, Gupta, & Efros, 2015), Physics-Informed Neural Network (PINN) (Raissi, Perdikaris, & Karniadakis, 2019), and Deep Synthetic Minority Oversampling Technique (DeepSMOTE) (Dablain, Krawczyk, & Chawla, 2022) that explored in Alzubaidi et al. (2023), the impact of the number of input samples in the field was explored in Akbari et al. (2022). For a fair comparison, 10 models of smartphones having the most videos were used to train the deep learning model in this paper. The minimum number of videos for a class is 60, and the maximum number of videos for a class is 294, for a total of 1378 videos. Fig. 3 shows the number of videos related to the training, testing, and validation sets for the Daxing database. The QUFVD (Akbari et al., 2022) dataset consists of 6000 videos from the latest and prominent smartphones that are from 10 models. Each model has two devices having a total of 600 videos. The I-frame was extracted and a total of 76,531 frames were produced. Fig. 4 shows samples of the databases.

The extracted frames consists of intra-coded picture(I-frame), predictive coded picture (P-frame), and bi-predictive coded picture (B-frame). I-frames displayed exceptional results compared to the rest (Altinisik & Sencar, 2020; Hosler et al., 2019). The importance of the I-frame is evident in its wide use in literature.

5. Experiments

When evaluating the hierarchical network, various scenarios are considered. The problem is considered as a 10 class classification problem. The experimental configuration is separated into various scenarios that display the impact of each scenario on the outcome. Each scenario is related to a different configuration of the fusion layer and streams used in training. The proposed network is compared with MISLnet architecture (Bayar & Stamm, 2018) which was used in several state-of-art studies such as Hosler, Mayer, Bayar et al. (2019), Mayer et al. (2020), Timmerman et al. (2020) and PRNU-Net (Akbari, Almaadeed, Al-Maadeed et al., 2022). The implementation of this experimental setup was majorly based on Timmerman et al. (2020), which was used to identify the source camera. Training is performed using stochastic gradient descent (SGD). The batch size is 100 and the parameters for momentum and decay of the stochastic gradient descent are 0.95 and 0.0005. Train and test split was set to 80% and 20% respectively for both Daxing and QUFVD. Validation data was derived from the train data, which was 20%. The training was performed for 10 epochs in each experiment. I-frames of the videos are used to extract and evaluate the performance of the database. All the I-frames of the videos are used for training, validation and testing. To identify a video based on its I-frame, all of the I-frames present in the video must be included in the test set. The scores while classifying the database show that the highest probability of the class is in the CNN based model. A majority vote is utilized to decide whether all the frames belong to a certain video. A patch based training is performed where the patch size is set to 350 × 350. The fully connected neurons based on the best result is 100 for each stream that for the number of dictionary atoms is $d = 600$. A 64-bit operating system (Ubuntu 18) is used with MATLAB R2022a with Deep Learning Toolbox, a CPU E5-2650 v4 @ 2.20 GHz, 128.0 GB RAM, and four NVIDIA GTX TITAN X GPU. Implementation details and source code, and databases links are freely available at: https://github.com/YounesAkbari/Source-Camera-Model-Identification-.

5.1. Results and discussion

Model camera identification has been considered by deep learning methods. The methods improve over the traditional methods. The hierarchical approach is found to be more successful than the MISLnet and PRNU-Net for the problem in all device models for both datasets. The results obtained at the video level are significantly better for all device models for both databases. The results of QUFVD compared to Daxing show that QUFVD produces better results, possibly due to the number of videos the two databases may contain where QUFVD has 6000 videos and Daxing has 1378 videos. Furthermore, our results obtained from PRNU-Net demonstrate superior performance in comparison to our approach using MISLnet. The results are discussed in detail below.

Our results for the frame and video levels in terms of overall accuracy (%) for the QUFVD and Daxing databases are shown in Table 1. Our results for the frame level in terms of the confusion matrix are shown in Tables 2 and 3 for the QUFVD and Daxing databases. The results of the frame and video levels in terms of precision (%) for each smartphone model based on our approach, the MISLnet, and the PRNU-Net scenarios are listed in Tables 4 and 5. Tables 2 and 4 (QUFVD database) in terms of recall and precision, show that at the frame level, a few devices are difficult to classify such as Samsung Note9, Iphone8 plus, and Nokia 7.1. An extended analysis is required to be explored to identify the reason for this imbalance. The expected argument can be the resolution of videos or the imaging technology used. However, models with lesser resolution such as Y7 and Y9, are not the lowest-performing models. The biggest improvement in frame level is found in Xiaomi Redmi Note9 Pro which compared to MISLnet was about 9.9%. The overall improvement was at video level, which is also for Redmi Note9 pro, which compared to MISLnet produces about 15.1%. On the frame level, Nokia 5.4 achieved the best results with 89.2%. On the video level, Note9 achieved the best results with 97.8% precision.

As shown in Tables 3 and 5 (Daxing database), few devices were hard to identify such as the Huawei P9, iPhone 6, and iPhone 6S Plus at the frame level. The largest spike in improvement was Huawei p9 in comparison to MISLnet which was a 22% increase. An overall improvement was noted at the video level for all the devices. The best result was obtained for Xiaomi 4 A with a precision of 98%.

With this premise, Fig. 5 provides a more comprehensive picture of camera identification performance to check the quality of our approach compared to PRNU-Net which achieves better results by presenting the Receiver Operating Characteristic (ROC) curves in terms of the two databases. For plotting the ROC, the True Positive Ratio (TPR) also termed sensitivity, and False Positive Ratio (FPR) were calculated. The

Table 2
Confusion matrix of QUFVD database. The rows correspond to the predicted class (Output Class) and the columns correspond to the true class (Target Class). The column on the far right of the table is called precision and the row at the bottom of the table is called recall. Classes 1 to 10 are Samsung A50, Samsung Note9, Huawei Y7, Huawei Y9, iPhone 8 Plus, iPhone XS Max, Nokia 5.4, Nokia 7.1, Xiaomi Redmi Note8, Xiaomi Redmi Note9 Pro, respectively.

1221	48	15	15	30	0	39	53	7	37	83.3
31	1045	18	12	20	50	28	31	4	12	83.5
17	71	1459	108	59	11	52	85	17	65	75.0
36	35	75	1415	43	2	6	47	36	19	82.5
23	41	25	51	1146	57	37	57	25	35	76.5
15	54	2	7	37	1199	20	43	1	40	84.5
31	37	2	5	23	12	1324	19	25	6	89.2
12	79	10	9	67	10	13	1034	22	20	81.0
46	48	25	8	75	3	17	62	1437	44	81.4
15	89	10	13	94	29	14	22	4	1207	80.6
84.3	67.5	88.9	86.1	71.8	87.3	85.4	71.1	91.0	81.2	81.6

Table 3
Confusion matrix of Daxing database. Classes 1 to 10 are Huawei P20, Huawei P9, iPhone 6, iPhone 6S, iPhone 6S Plus, iPhone 7 Plus, OPPO R9, OPPO R11, VIVO X9, Xiaomi 4A, respectively.

192	0	21	3	15	2	0	0	16	0	77.1
3	110	17	28	2	0	11	7	13	1	57.3
0	3	159	31	41	4	1	1	0	0	66.2
2	20	83	418	82	37	26	20	11	9	59.0
12	49	2	55	207	35	4	11	25	25	48.7
10	16	12	40	31	134	10	1	6	1	51.3
0	18	7	21	2	10	207	2	1	0	77.2
2	14	11	1	6	1	3	146	0	0	79.3
0	3	3	7	1	0	2	0	116	1	87.2
0	0	0	1	1	0	0	0	2	102	96.2
86.9	47.2	50.5	69.1	53.4	60.1	78.4	77.7	61.1	73.4	69.9

Table 4
The results of the frame and video levels in terms of precision (%) for each smartphone model in terms of QUFVD.

Model	I-frame				Video			
	MISLnet	Ours	PRNU-Net	Ours	MISLnet	Ours	PRNU-Net	Ours
Samsung A50	72.8	82.1	75.0	**83.3**	73.3	**86.5**	77.2	86.2
Samsung Note9	78.7	**83.7**	78.8	83.5	95.8	97.5	95.8	**97.8**
Huawei Y7	68.0	74.6	71.5	**75.0**	84.2	88.5	86.0	**88.7**
Huawei Y9	76.9	**82.6**	77.3	82.5	86.7	95.8	91.6	**96.5**
iPhone 8 Plus	67.8	**76.9**	73.9	76.5	84.2	90.2	85.5	**92.1**
iPhone XS Max	76.8	83.0	79.2	**84.5**	68.3	75.5	74.9	**75.9**
Nokia 5.4	81.8	87.5	83.1	**89.2**	90.8	94.2	92.7	**95.0**
Nokia 7.1	75.5	80.2	80.6	**81.0**	90.0	93.6	92.2	**93.9**
Xiaomi Redmi Note8	75.8	81.6	**81.9**	81.4	80.8	85.1	84.2	**85.4**
Xiaomi Redmi Note9 Pro	66.4	76.3	75.0	**80.6**	65.8	80.9	77.3	**82.2**
Overall precision	74.0	80.8	77.6	**81.8**	82.0	88.7	85.7	**89.4**

Table 5
The results of the frame and video levels in terms of precision (%) for each smartphone model in terms of Daxing.

Model	I-frame				Video			
	MISLnet	Ours	PRNU-Net	Ours	MISLnet	Ours	PRNU-Net	Ours
Huawei P20	71.0	75.1	71.6	**77.1**	73.0	78.8	76.2	**80.1**
Huawei P9	32.3	54.8	35.4	**57.3**	35.5	**60.9**	56.5	60.2
iPhone 6	65.5	65.9	65.3	**66.2**	72.1	73.8	74.5	**75.1**
iPhone 6S	56.8	**59.2**	58.2	59.0	60.4	**63.9**	60.8	63.2
iPhone 6S Plus	44.4	46.9	**50.3**	48.7	51.0	**55.2**	50.9	55.0
iPhone 7 Plus	46.5	42.7	51.2	**51.3**	50.2	48.9	**52.1**	51.2
OPPO R9	59.8	73.3	60.1	**77.2**	63.9	**80.2**	64.5	79.0
OPPO R11	**80.6**	76.8	79.6	79.3	83.9	82.0	83.5	**84.6**
VIVO X9	82.2	85.6	83.2	**87.2**	86.0	90.3	86.9	**90.5**
Xiaomi 4A	87.5	95.2	91.2	**96.2**	90.9	**98.0**	91.0	96.5
Overall precision	62.7	67.6	64.6	**69.9**	66.7	73.2	69.6	**73.5**

TPR can be defined as the true positive predictions divided by the predictions. It evaluates that a predicted class is in the actual class. On the other hand false positive rate identifies the frames that are identified as not being in a particular class. Fig. 5 shows the TPR compared to FPR for two methods at different frame-level threshold for both the databases. All the models are shown to achieve a large Area Under

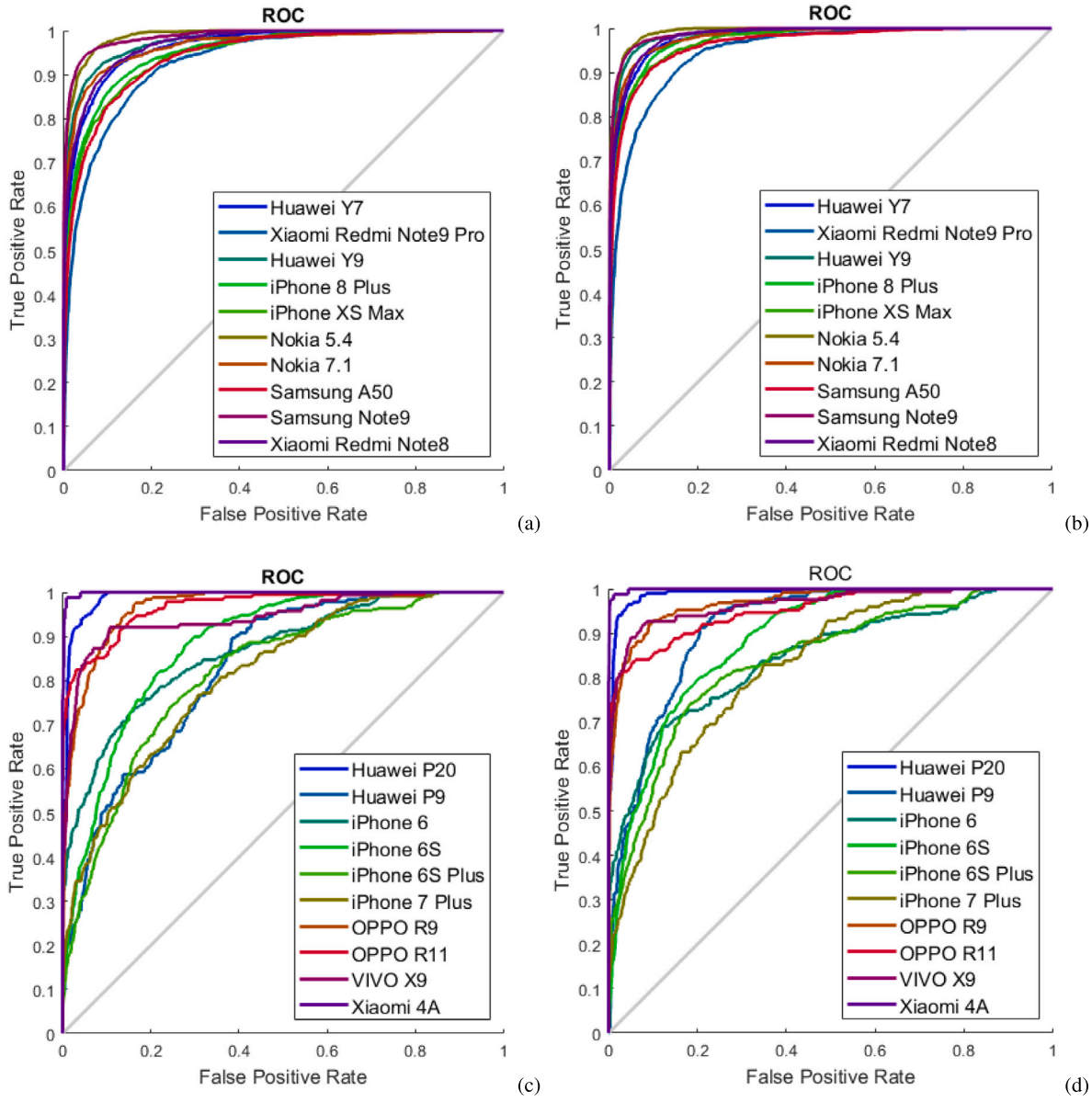

Fig. 5. True and false positive rates (ROC) obtained in terms of two databases (a) 10 classes with PRNU-Net (QUFVD) (b) 10 classes with our approach (QUFVD) (c) 10 classes with PRNU-Net (Daxing) (d) 10 classes with our approach (Daxing) (It can be zoomed to have better quality)

Curve (AUC) value compared to PRNU-Net for both the databases. A separate analysis for QUFVD, shows that the best performing class is Nokia 5.4 with AUC as 0.9898 for our approach compared to AUC as 0.9895 for PRNU-Net (Fig. 5(b)). In the Daxing database analysis, the best performing class was Xiaomi 4 A with AUC as 0.9955 for the proposed approach (Fig. 5(c)) and for the PRNU-Net (Fig. 5(d)) approach it was 0.9919.

Although QUFVD and Daxing have a common device that is iPhone 8 Plus, the device was eliminated from the evaluation of the Daxing database due to its low number of videos which may hinder the performance of the deep learning models. To test the feasibility of this approach across database, a separate evaluation was performed on Daxing database with Iphone8 plus in the test set using the model trained on QUFVD (our approach based on PRNU-Net). It produced frame-level accuracy of 80.5% and video-level accuracy of 85.0%. This shows that the model trained can be generalized and used in real-world scenarios for model video identification.

5.1.1. *Impact of the fusion layer*

The impact of replacing the concatenation and addition layers with the fusion layer is explored in this section. Concatenation is performed along a specific dimension and an addition layer adds inputs element wise in multiple neural network layers. Table 6 shows the result for both databases on all the fusion-based methods and shows that all the three fusion methods improve the results obtained by PRNU-Net. It is worth noting that we conducted the subsequent experiment based on our approach with PRNU-Net, as it yielded better results compared to MISLnet in our initial experiment. This also shows that our fusion layer performs better than concatenation and addition.

5.1.2. *Impact of the streams*

In the evaluation, we determine how many streams are sufficient to achieve a promising result. We consider 2 to 6 streams for both databases. In the upper part of our architecture (see Fig. 2), we limit the number of streams to two for a two stream approach (the first stream is the top stream in our approach). We add three streams for three stream approach and repeat the same pattern for 4 and 5 streams. The results

Table 6
The results of the impact of the fusion layer based on the frame and video levels in terms of accuracy (%).

Fusion methods	Daxing		QUFVD	
	Frame	Video	Frame	Video
Add	64.9	70.0	78.1	86.0
Concatenation	65.5	70.2	76.9	85.2
Our approach	**69.9**	**73.5**	**81.6**	**89.4**

Table 7
The results of the impact of the streams based on the frame and video levels in terms of accuracy (%).

# of Streams	Daxing		QUFVD	
	Frame	Video	Frame	Video
2	50.5	55.0	63.2	66.3
3	59.3	62.0	73.1	80.0
4	67.3	72.6	78.9	87.4
5	68.1	73.0	79.6	88.0
6	**69.9**	**73.5**	**81.6**	**89.4**

are shown in Table 7. As can be seen from the table, the best results are obtained when we have all the streams. However, the improvement is not significant and the system with 4 streams can also be considered.

6. Conclusion

In this paper, Low, mid, and high-level features were obtained from smartphone videos through a hierarchical deep neural network. A joint sparse representation method was implemented to fuse the extracted features. An unsupervised multi-modal dictionary learning was used for this approach. The features extracted from the fusion layer in the forward function and derivative loss is computed in the backward propagation. The evaluation of this method was performed on two databases, QUFVD and Daxing, which contain 10 popular models of devices. The total number of videos used for evaluation were 6000 original videos for QUFVD and 1378 videos for Daxing. The approach has proven to be better compared to MISLnet and PRNU-Net architecture producing better results. The impact of the fusion layer was analyzed in comparison to concatenation and additional layers. The results showed that the fusion layer generally improves the results and in particular our fusion layer outperforms the other two fusion layers. Next, we investigated the impact of the number of streams. Our results showed that the best results were obtained with the six-stream approach. Although good accuracy has been achieved, there is still room for improvement in future work.

In our future work, other deep learning architectures will be evaluated with the proposed fusion layer. It would also be worth considering a sequential frame analysis rather than single frame analysis. The impact of the number of dictionary atoms (less or larger than the input data) should be explored further. Finally, the proposed approach could be applied in different computer vision problems given the significant improvements obtained for camera model identification.

CRediT authorship contribution statement

Younes Akbari: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Visualization. **Somaya Al Maadeed:** Supervision, Conceptualization, Writing – review & editing, Project administration, Funding acquisition. **Omar Elharrouss:** Conceptualization, Writing – review & editing, Visualization. **Najmath Ottakath:** Conceptualization, Writing – review & editing, Visualization. **Fouad Khelifi:** Supervision, Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This publication was made possible by NPRP, Qatar grant # NPRP12S-0312-190332 from Qatar National Research Fund (a member of Qatar Foundation). The statement made herein are solely the responsibility of the authors.

References

Abavisani, M., & Patel, V. M. (2018). Multimodal sparse and low-rank subspace clustering. *Information Fusion*, *39*, 168–177.
Aharon, M., & Elad, M. (2008). Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, *1*(3), 228–247.
Ahmad, S., & Scheinkman, L. (2019). How can we be so dense? The benefits of using highly sparse representations. arXiv preprint arXiv:1903.11257.
Ahmed, F., Khelifi, F., Lawgaly, A., & Bouridane, A. (2019). Comparative analysis of a deep convolutional neural network for source camera identification. In *2019 IEEE 12th international conference on global security, safety and sustainability (ICGS3)* (pp. 1–6). IEEE.
Akbari, Y., Al-Maadeed, S., Al-Maadeed, N., Al-Ali, A., Khelifi, F., Lawgaly, A., et al. (2022). A new forensic video database for source smartphone identification: Description and analysis. *IEEE Access*, *10*, 20080–20091.
Akbari, Y., Al-maadeed, S., Elharrouss, O., Khelifi, F., Lawgaly, A., & Bouridane, A. (2022). Digital forensic analysis for source video identification: A survey. *Forensic Science International: Digital Investigation*, *41*, Article 301390.
Akbari, Y., Almaadeed, N., Al-Maadeed, S., Khelifi, F., & Bouridane, A. (2022). PRNU-net: a deep learning approach for source camera model identification based on videos taken with smartphone. In *2022 26th international conference on pattern recognition (ICPR)* (pp. 599–605). IEEE.
Akbari, Y., Elharrouss, O., & Al-Maadeed, S. (2022). Feature fusion based on joint sparse representations and wavelets for multiview classification. *Pattern Analysis and Applications*, 1–9.
Altinisik, E., & Sencar, H. T. (2020). Source camera verification for strongly stabilized videos. *IEEE Transactions on Information Forensics and Security*, *16*, 643–657.
Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A., Al-dabbagh, B. S. N., et al. (2023). A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, *10*(1), 46.
Bahrampour, S., Nasrabadi, N. M., Ray, A., & Jenkins, W. K. (2015). Multimodal task-driven dictionary learning for image classification. *IEEE Transactions on Image Processing*, *25*(1), 24–38.
Bayar, B., & Stamm, M. C. (2018). Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2691–2706.
Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177–186). Springer.
Bottou, L., & Bousquet, O. (2007). The tradeoffs of large scale learning. *Vol. 20*, In *Advances in neural information processing systems*.
Chuang, W.-H., Su, H., & Wu, M. (2011). Exploring compression effects for improved source camera identification using strongly compressed video. In *2011 18th IEEE international conference on image processing* (pp. 1953–1956). IEEE.
Cotter, S. F., Rao, B. D., Engan, K., & Kreutz-Delgado, K. (2005). Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, *53*(7), 2477–2488.
Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*.
Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision* (pp. 1422–1430).
Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, *15*(12), 3736–3745.
Galdi, C., Hartung, F., & Dugelay, J.-L. (2019). SOCRatES: A database of realistic data for source camera recognition on smartphones. In *ICPRAM* (pp. 648–655).
Goljan, M., Fridrich, J., & Filler, T. (2009). Large scale test of sensor fingerprint camera identification. *Vol. 7254*, In *Media forensics and security*. International Society for Optics and Photonics, Article 72540I.

Gui, J., Tao, D., Sun, Z., Luo, Y., You, X., & Tang, Y. Y. (2014). Group sparse multiview patch alignment framework with view consistency for image classification. *IEEE Transactions on Image Processing*, *23*(7), 3126–3137.

Hosler, B., Mayer, O., Bayar, B., Zhao, X., Chen, C., Shackleford, J. A., et al. (2019). A video camera model identification system using deep learning and fusion. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 8271–8275). IEEE.

Hosler, B. C., Zhao, X., Mayer, O., Chen, C., Shackleford, J. A., & Stamm, M. C. (2019). The video authentication and camera identification database: A new database for video forensics. *IEEE Access*, *7*, 76937–76948.

Iuliani, M., Fontani, M., Shullani, D., & Piva, A. (2019). Hybrid reference-based video source identification. *Sensors*, *19*(3), 649.

Kan, M., Shan, S., Zhang, H., Lao, S., & Chen, X. (2015). Multi-view discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(1), 188–194.

Kang, X., Li, Y., Qu, Z., & Huang, J. (2011). Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, *7*(2), 393–402.

Kirchner, M., & Johnson, C. (2019). Spn-cnn: Boosting sensor-based source camera attribution with deep learning. In *2019 IEEE international workshop on information forensics and security (WIFS)* (pp. 1–6). IEEE.

Lawgaly, A., & Khelifi, F. (2016). Sensor pattern noise estimation based on improved locally adaptive DCT filtering and weighted averaging for source camera identification and verification. *IEEE Transactions on Information Forensics and Security*, *12*(2), 392–404.

Lawgaly, A., Khelifi, F., Bouridane, A., Al-Maaddeed, S., & Akbari, Y. (2022a). PRNU estimation based on weighted averaging for source smartphone video identification. *Vol. 1*, In *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 75–80). IEEE.

Lawgaly, A., Khelifi, F., Bouridane, A., Al-maadeed, S., & Akbari, Y. (2022b). Three dimensional denoising filter for effective source smartphone video identification and verification. In *2022 7th international conference on machine learning technologies (ICMLT)* (pp. 124–130).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788.

Li, S.-Y., Jiang, Y., & Zhou, Z.-H. (2014). Partial multi-view clustering. In *Twenty-eighth AAAI conference on artificial intelligence*.

Li, B., Yuan, C., Xiong, W., Hu, W., Peng, H., Ding, X., et al. (2017). Multi-view multi-instance learning based on joint sparse representation and multi-view dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2554–2560.

Li, S., & Zhang, B. (2021). Joint discriminative sparse coding for robust hand-based multimodal recognition. *IEEE Transactions on Information Forensics and Security*, *16*, 3186–3198.

Li, J., Zhang, D., Li, Y., Wu, J., & Zhang, B. (2017). Joint similar and specific learning for diabetes mellitus and impaired glucose regulation detection. *Information Sciences*, *384*, 191–204.

Li, J., Zhang, B., Lu, G., & Zhang, D. (2019). Generative multi-view and multi-feature learning for classification. *Information Fusion*, *45*, 215–226.

Li, J., Zhang, B., & Zhang, D. (2017). Joint discriminative and collaborative representation for fatty liver disease diagnosis. *Expert Systems with Applications*, *89*, 31–40.

López, R. R., Luengo, E. A., Orozco, A. L. S., & Villalba, L. J. G. (2020). Digital video source identification based on container's structure analysis. *IEEE Access*, *8*, 36363–36375.

López, R. R., Orozco, A. L. S., & Villalba, L. J. G. (2021). Compression effects and scene details on the source camera identification of digital videos. *Expert Systems with Applications*, *170*, Article 114515.

Mahalanobis, A., Kumar, B. V., & Casasent, D. (1987). Minimum average correlation energy filters. *Applied Optics*, *26*(17), 3633–3640.

Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, *11*(1).

Mairal, J., Elad, M., & Sapiro, G. (2007). Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, *17*(1), 53–69.

Mandelli, S., Bestagini, P., Verdoliva, L., & Tubaro, S. (2019). Facing device attribution problem for stabilized video sequences. *IEEE Transactions on Information Forensics and Security*, *15*, 14–27.

Mayer, O., Hosler, B., & Stamm, M. C. (2020). Open set video camera model verification. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2962–2966). IEEE.

Mayer, O., & Stamm, M. C. (2019). Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, *15*, 1331–1346.

Parikh, N., Boyd, S., et al. (2014). Proximal algorithms. *Foundations and Trends® in Optimization*, *1*(3), 127–239.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.

Rakotomamonjy, A. (2011). Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Processing*, *91*(7), 1505–1526.

Shao, L., Liu, L., & Yu, M. (2016). Kernelized multiview projection for robust action recognition. *International Journal of Computer Vision*, *118*(2), 115–129.

Shullani, D., Fontani, M., Iuliani, M., Al Shaya, O., & Piva, A. (2017). VISION: a video and image dataset for source identification. *EURASIP Journal on Information Security*, *2017*(1), 1–16.

Tian, H., Xiao, Y., Cao, G., Zhang, Y., Xu, Z., & Zhao, Y. (2019). Daxing smartphone identification dataset. *IEEE Access*, *7*, 101046–101053.

Timmerman, D., Bennabhaktula, S., Alegre, E., & Azzopardi, G. (2020). Video camera identification from sensor pattern noise with a constrained ConvNet. arXiv preprint arXiv:2012.06277.

Villalba, L. J. G., Orozco, A. L. S., López, R. R., & Castro, J. H. (2016). Identification of smartphone brand and model via forensic video analysis. *Expert Systems with Applications*, *55*, 59–69.

Vo, Q. N., Kim, S. H., Yang, H. J., & Lee, G. (2018). Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, *74*, 568–586.

Wang, W., Arora, R., Livescu, K., & Bilmes, J. (2015). On deep multi-view representation learning. In *International conference on machine learning* (pp. 1083–1092).

Yang, M., Zhang, L., Zhang, D., & Wang, S. (2012). Relaxed collaborative representation for pattern classification. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2224–2231). IEEE.

Yu, Z., & Gao, S. (2020). Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1949–1958).

Yuan, X.-T., Liu, X., & Yan, S. (2012). Visual classification with multitask joint sparse representation. *IEEE Transactions on Image Processing*, *21*(10), 4349–4360.

Zhang, H., Zhang, Y., Nasrabadi, N. M., & Huang, T. S. (2012). Joint-structured-sparsity-based classification for multiple-measurement transient acoustic signals. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *42*(6), 1586–1598.

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, *26*(7), 3142–3155.