iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: III Month of publication: March 2021

DOI: https://doi.org/10.22214/ijraset.2021.33305

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com

Camera Model Identification using Convolutional Neural Network

Teena Varma[1], Aditi Bane[2], Shraddha Bundge[3], Bethel Thomas[4]
[1, 2, 3, 4]Department of Computer Engineering & University of Mumbai, India

Abstract: Source camera identification is the procedure of finding out which camera model has been used to capture an image. In the past few years, there has been a high-speed growth of research interest in the field of forensics. In our present work, we have proposed a Deep Learning approach for identifying the camera model of ten cameras as a part of the Camera Model Identification Challenge organized by the Kaggle.com. Through this paper, we have presented a camera model identification method based on Convolutional Neural Network (CNN). In contrast to traditional methods, CNNs can spontaneously and concurrently extricate features and can be trained to classify during the learning process. Source camera identification is used in legal as well as security matters as a proof. As a correlation to previous task, researchers have suggested to utilize the artifacts that are present in the pipeline of camera to gather particular features manually and use them to differentiate between individual devices or camera models. Our task is implemented with 10 camera models from the dataset provided by the IEEE's Signal Processing Society with multi class CNN classifier.
Keywords: Camera Identification, Deep Learning, Convolutional Neural Network, Fully Connected Network.

I. INTRODUCTION

Digital forensics has gained a great observation from law imposition agencies and scholastic investigators in the previous decade. The availability and transference of digital images have been growing exceptionally due to powerful development in network and computing technologies. Due to wide accessibility of image editing software appliances, digital images can be easily edited, modified or distorted. Therefore, the dependability and solidity of digital images are cross-examined when used as verification in security and legal domains. Authentic forensic techniques are crucially required by law imposition agencies to reinstate the certainty to digital images. Over the last few years, results based on Deep Learning methods became an orthodox in computer vision. Issues in medical imagery, satellite imagery or any other imagery are victoriously tackled with these techniques, habitually subduing human performance in many pieces of works, including categorization, fragmentation, and apprehension. There are a hardly any causes why these methods are in demand. In the first place, they are flexible in a way, that expanding the model to work with unfamiliar models is an elementary job that does not need any notable forensic domain understanding or practice, making this problem engineering preferably than an experimental problem. Moreover, factual proof displays the accuracy of the models is rising when more training data is given, which permits taking help of the verity that extremely impressive number of images and videos are issued on the cyberspace. There are numerous distinct works where authors had used Deep Learning methods to forensics. Such as, in certain papers, authors used CNN algorithm to identify double JPEG Compression, while other authors operated CNN to extricate features and classifiers of SVM in addition to the extricated features

II. LITERATURE REVIEW

In the past decades, many methods have been put forward to decide which camera was used to capture the image. Almost all of the present approaches that do not consider metadata can be split up into two categories: hardware and software source camera detection. Hardware category considers features of camera hardware, like Charge Coupled Device (CCD) sensors and lens. Although, software approach works with Sensor Pattern Noise (SPN)and color filter array (CFA) interpolation artifacts. The present encouraging methods as state by Van Lanh T et al are as follows:

1) Lens characteristics
2) Noise pattern in digital cameras.

The initial method was proposed in the paper published by Choi et al. in 2006. The method is focused on lens radial distortion where the distortion parameters were used as characteristics for classification algorithm. This optical deviation takes place because of the use of the low-quality wide-angle lenses which have a low cost. Producers are carrying out various lens systems to remunerate for radial distortion, where they are influencing the pattern of the radial distortion. The major limitation of this approach is substituting to traditional lenses which can reduce the classification accuracy or manual zooming. In 2007, Van Lanh T et al. expanded this method and applied it to cameras of mobile phones. This is one of the rare methods that acquire outcomes on early detection stage like lenses. The next approach was first put forward by Luka et al. in 2006. Silicon wafers had numerous defects and different homogeneity which were used throughout the production of the sensors.

Consequently, pixels at various positions have a distinct sensitivity to light which shows a unique way to each camera pattern of noise which is contemplated as the major component of Pixel Non-Uniformity (PRNU). The authors strengthened the previous algorithm by removing the average whitened sensor pattern noise. The drawback of this method is the recommendation to use the smooth content pictures to take out appropriate and genuine noise-based fingerprint. In this approach, the authors use a feature extrication pipeline, consisting of edge extraction using the Laplace and canny operators, and combining them with the original images to extract Homogeneity, Entropy, Contrast and Correlation. The authors made use of SVM and other classifiers on top of these characteristics to achieve top accuracy outcomes on the Dresden Dataset. In the last few years, techniques of deep learning were also put in to the camera identification task. Deep Learning method has the ascendancy of working with immensely high-capacity models, having tens of millions of free parameters. The power of this method is that Neural Networks do not need manual feature extraction as the model is trained on the relevant features straight from the data. This makes the approach ascendable in two ways. First of all, you can simply expand your identification algorithm to a large set of cameras, attaching new models if required. Secondly, the quality of the features that are extracted extends with the quantity of the data that is being used for training.

III.PROPOSED METHOD

Lately, Deep learning with the help of Convolutional Neural Networks (CNNs) have accomplished immense interest in many fields. Representations of features and preformation of classification instinctively from the primary images can be achieved by using frameworks of deep learning. Convolutional Neural Networks (CNNs) have exhibited superior performances in several tasks of artificial intelligence like natural language processing and object recognition. The construction of a Convolutional Neural Network is composed of layers composed of neurons. The neurons get hold of the input values as well the computations are performed by the neurons and the neurons take the accountability to pass the result to next layer. Figure 1 depicts the prevalent structure of the CNN which also displays the resemblance with traditional machine learning approach. The layers of the CNNs are outlined in the further subsections.

A. Convolutional Layers & Classification Layers

The conventional layer comprises of three vital operations: convolution, activation function, and pooling. This output of the convolutional layer is called as a feature map, and also can be contemplated as a specific representation of a feature of the input image. Convolution formulation is: a l j = Pni=1 a l−1 i∗w l−1 ij + b l j, where ∗represents convolution, a l j is the j-th output map in layer l, w l−1 ij is convolutional kernel linking the i-th output map in layer l − 1 and the j-th output map in layer l, b l j is the training bias parameter for the j-th output map in layer l, and the number of feature maps as n, from layer l − 1. The activation function is put in application to each and every value of the filtered image. There are various types of the activation functions for example a sine function f(x) = sinus(x), an absolute function f(x) = |x|, or Rectified Linear Units (ReLU) function f(x) = max (0, x).

1) *Pooling is the Upcoming Major Step:* The pooling layer is frequently appended between two consecutive convolutional layers. Its role is to minimize the spatial size of the depiction and to reduce the number of parameters and computation in the network. Throughout the pooling, a highest value or an average value is computed. The final process done by the convolutional layer is normalization of the feature maps. The normalization is put on the feature maps with a view to get comparable values as output for each neuron. The classification layer consists of a SoftMax function and fully connected layers. In fully connected layer, neurons have all associations to all activations in the current previous layer. Matrix multiplication can be used to calculate the activations followed by a bias offset. The fully connected layer will calculate the scores of the class with the help of a SoftMax function. This way, Convolutional neural networks can be used to convert the primary image from pixel values to the final class scores.
2) *Learning Process:* When the learned features move through the fully connected layers, they will be provided to the top layer of the CNNs, where the classification is done by using SoftMax activation function. The Convolutional neural network is trained by the back-propagation algorithm. The weights and the bias can be later changed in the convolutional and fully connected layers because of the error propagation process. So, the classification output can be given back to lead the feature extraction instinctively and the learning process can be initiated successfully. The CNN architecture has thousands of parameters which may appear because of the overfitting problem. To reduce the overfitting the drop out technique can be used. It consists of setting the result of each hidden neuron with probability 0.5 to 0. The neurons which are terminated in this process do not put up to the forward pass and do not participate in the back propagation process. This technique gives a rise to robustness, since a neuron cannot hang on the presence of certain other neurons. It is, thus, imposed to adapt more robust features that are helpful in conjunction with many different random subsets of the other neurons.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue III Mar 2021- Available at www.ijraset.com

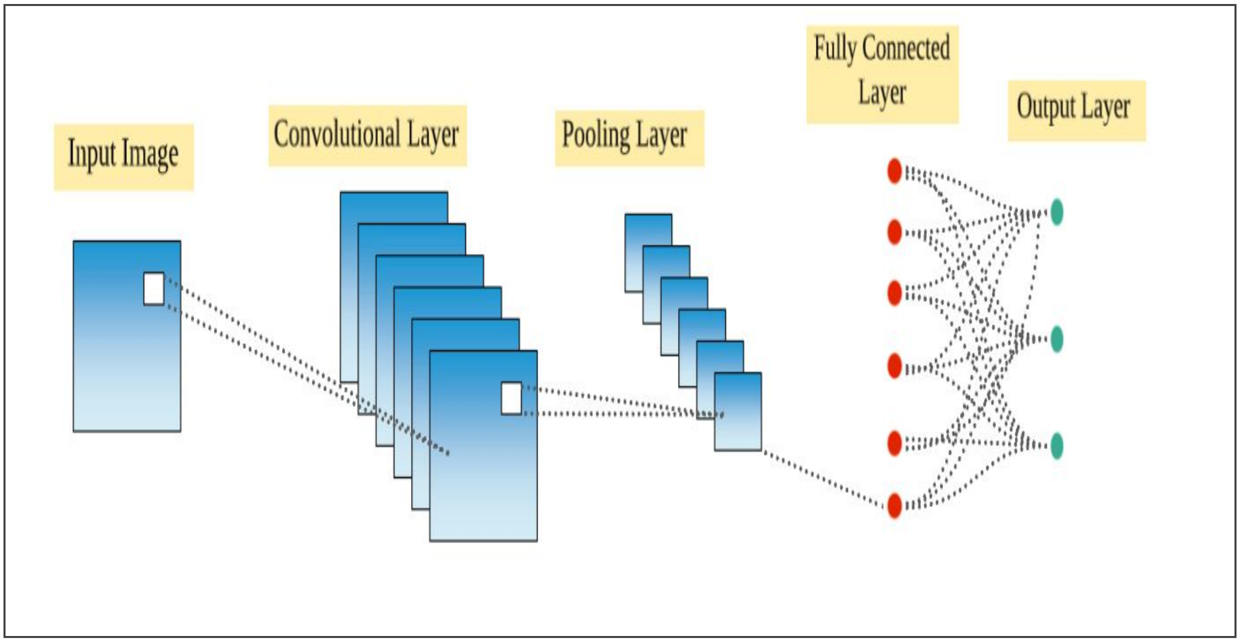


Fig 1. The Convolutional Neural Network Notion

B. Camera Model Identification Proposed Design in CNN

The structure of our model is shown in the Figure 2, where we have expressed the detailed settings of the architecture. The predominant layer is the filter layer, following the three convolutional layers from the first layer (Conv1) to the third layer (Conv3). Although the ultimate three layers are the fully-connected layers (FC1, FC2, FC3) which are used for the classification. The features of the CNN model are shown in the forthcoming subsections.

C. Filter Layer

The major way for denoising an image is to put a denoising filter on the image. For each and every image which is indicated by I, the residual noise is taken out by removing the denoised version of the image from the image alone as follows: N =I − F(I), (2) The denoised image is indicated by F(I), and F is a denoising filter. This filter will be utilized later in our experiments and applied on every color channel distinctly.

An additional denoising high-pass filter is applied on the input image I. Applying this kind of filter has much value in the proposed approach as it can subdue the intervention caused by edges of the image and textures with regard to get the image residual as follows: A = I ∗1 12 −1 2 −2 2 −1 2 −6 8 −6 2 −2 8 −12 8 −2 2 −6 8 −6 2 −1 2−2 2 −1. The result of this step will be provided to the CNN. In our observations, we have tested two types of filters as preprocessing. The first one is the high pass filter and the second one is the very popular wavelet based denoising filter.

D. Convolutions

Convolutional Neural Networks (CNN) is adjusted and transformed particularly to fit the requirements of model. The initial convolutional layer (Conv1) handles the residual image with 64 kernels of size 3 × 3. The size of the result produced by the feature maps is 126 × 126. Then the second convolutional layer (Conv2) takes the outcome of the first layer as its input. Convolutions is applied on it with kernels of size 3×3 and it generates feature maps of size 64×64. The third convolution layer puts on convolutions with 32 kernels of size 3 × 3.

The Rectified Linear Units (ReLUs) is an activation function which is stochastic and which is applied to the outcome of each and every convolutional layer. ReLUs is generally contemplated as the typical way to model the neurons output and it can lead a way to faster convergence with huge models trained on large datasets. The third convolutional layer comes after the max pooling operation with window size 3×3, which does operations on the feature map in the correlating with convolutional layer, and lead a way to the same number of feature map with decreasing spatial resolution.

E. Fully Connected Layers

The fully-connected layers (FC1) and (FC2) have 256, and 4096 neurons in them individually. ReLUs activation function is applied to the outcome of every fully connected layer. Each of the fully connected layers (FC1) and (FC2) are relinquished during the learning process. The output of final fully connected layer (FC3) is then provided to a SoftMax function.

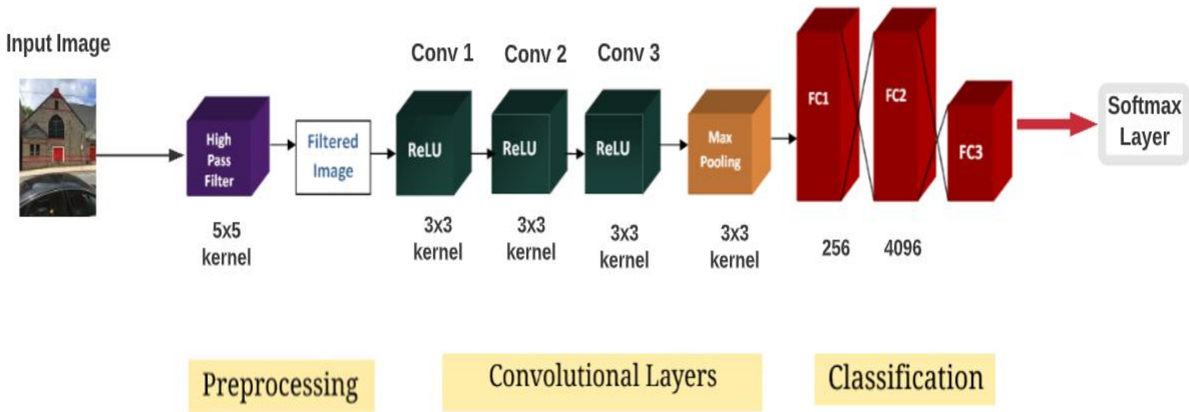


Fig 2. Flowchart of Convolutional Neural Network for Camera Model Identification

IV. CONCLUSIONS

In our paper, we assessed the effectiveness of utilizing CNN algorithm for source camera model identification based upon deep learning and convolutional neural networks. The endowment acts as a substantial challenge because it is fairly different from current common approaches for source camera model identification. Extensibility has also been assessed and the rise in the quantity of models reduces the accuracy somewhat significantly.

Adding a greater number of layers appears to be favorable and future work should traverse through substantial networks for example Microsoft's ResNet (which comprises over 150 layers). In the present work, we illustrated the implementation of the deep learning techniques trained on the large number of the data, which was scraped from the web. In circumstances, quality schedules of training, neural network formation and photo augmentations would help in making a model that displays exceptional performance in the camera model identification task. Depending on the presented model, the work of this project can be directly put in practice. In our approach, we accomplished filtering of data to circumvent the use of manipulated pictures during training that notably reduced the available imagery data. We have confidence that enormous number of the camera imagery data used for training may produce a better-quality model. Although, we did not carry out a straight juxtaposition of these two proposed models in our current work leaving it for further future research.

V. FUTURE WORK

The way forward to this approach contains polishing up the presently used methodology to a furthermore precise categorization by augmenting the previous set of features and inspecting and deciding upon correcting the issue of the geometrical transformations. Moreover, an additional unspecified class will also be one of the outlooks to control the models which were not in the train set.

The achievable goals:

We are planning to provide a function, where we are going to create web application using Flask, HTML, CSS, JS, etc. The future work will include using a database of a bigger volume having additional camera models, so as the usage of multiple devices of the same model.

The far-fetched goal:

Now this the one goal, we are not so sure about. It is the one goal that has failed in all the attempts by previous computer-generated camera model identification works.

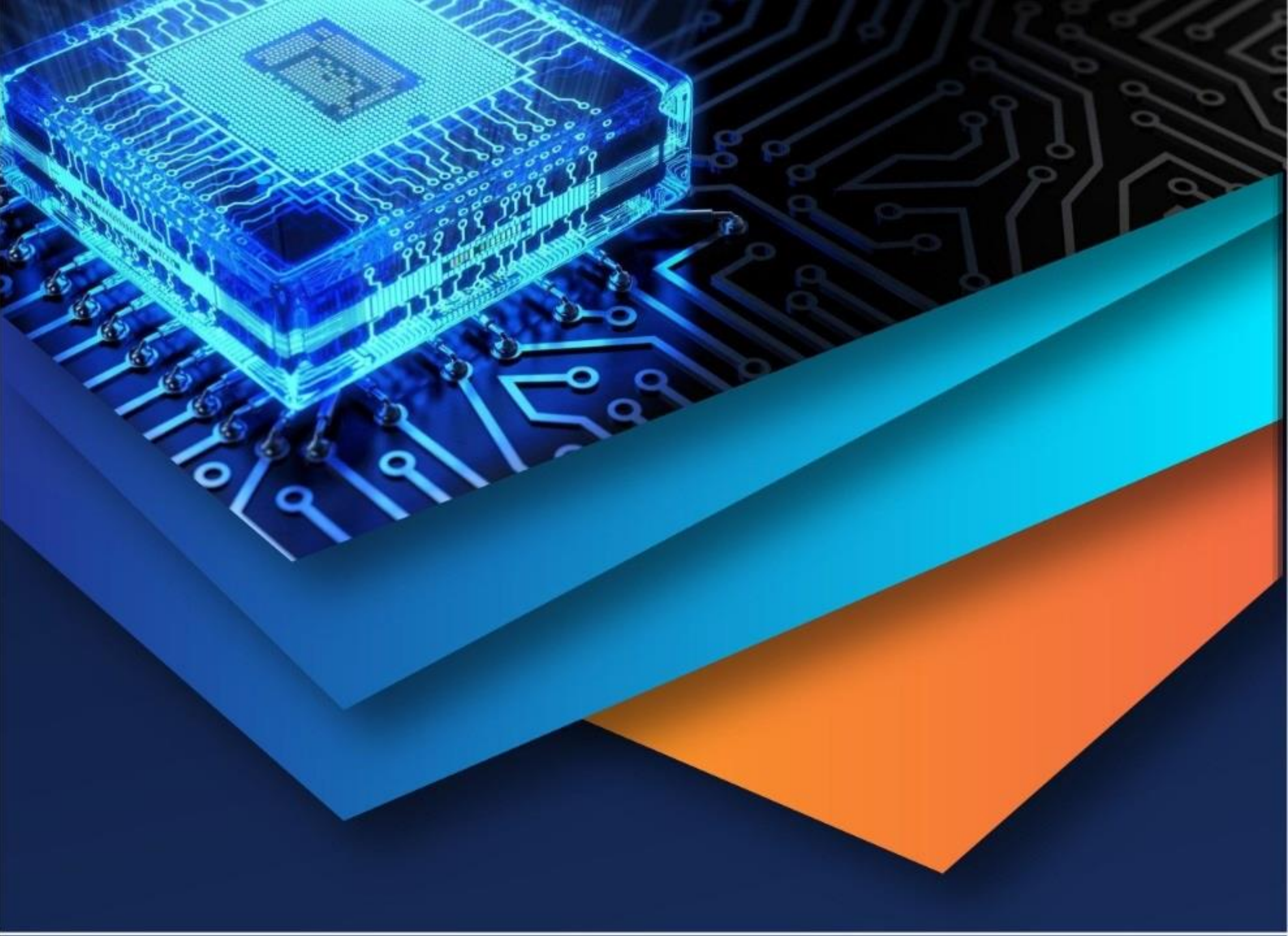
The main idea in this is adding a large number of features to permit intensifying the identification rate by providing powerful demographic tool. The target of this project is to try and implement a proper structure to the generated camera model identification.

VI. ACKNOWLEDGMENT

We sincerely thank our mentor and guide Prof. Teena Varma for her persistent guidance and assistance in our research. She gave us the golden opportunity to do this project on the topic Camera Model Identification using Convolutional Neural Network, which also helped us in doing a lot of research and we learned about so many new things. Without her tutelage, we wouldn't be able to complete this project. Special thanks to her for sharing her knowledge with us and guiding us throughout this project.

REFERENCES

[1] Artur Kuzin, Artur Fattakhov, Ilya Kibardin, Vladimir I. Iglovikov and Ruslan Dautov, "Camera Model Identification Using Convolutional Neural Networks" in IEEE International Conference on Big Data (Big Data),2018
[2] AmelTuama, Fr´Ed´EricComby And Marc Chaumont," Camera Model Identification with The Use of Deep Convolutional Neural Networks
[3] AmelTuama, Fr´Ed´EricCombyAnd Marc Chaumont," Camera Model Identification Based Machine Learning Approach With High Order Statistics Features" in 24th European Signal Processing Conference (Eusipco), 2016
[4] M. Kirchner and T. Gloe, "Forensic camera model identification," in T. Ho, S. Li, (eds.) Handbook of Digital Forensics of Multimedia Data and Devices. Wiley-IEEE Press, 2015.
[5] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," IEEE Transactions on Information Forensics and Security, vol. 1, no. 2, pp. 205–214, June 2006.
[6] K. Choi, E. Lam, and K. Wong, "Source camera identification using footprints from lens aberration," in Proc. SPIE, Digital Photography II, vol. 6069, no. 1, pp. 60 690J, 2006.
[7] A. E. Dirik, H. T. Sencar, and N. Memon, "Source camera identification based on sensor dust characteristics," in IEEE Workshop on Signal Processing Applications for Public Security and Forensics, SAFE '07, Washington, USA, pp. 1–6, 2007.
[8] S. Bayram, H. Sencar, and N. Memon, "Improvements on source camera model identification based on cfa interpolation," in Advances in Digital Forensics II, IFIP International Conference on Digital Forensics, Orlando Florida, pp. 289–299, 2006.
[9] M. Kharrazi, H. Sencar, and N. Memon, "Blind source camera identification," in IEEE International Conference on Image Processing ICIP. 2004., vol. 1, pp. 709–712, 2004.
[10] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind identification of source cell-phone model." IEEE Transactions on Information Forensics and Security, vol. 3, no. 3, pp. 553–566, 2008.
[11] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in Proc. of the 15th IEEE International Conference on Image Processing (ICIP), San Diego, California, October 12-15, pp. 1296–1299, 2008.
[12] T. Gloe, "Feature-based forensic camera model identification," Y.Q. Shi, S. Katzenbeisser, (eds.) Transactions on Data Hiding and Multimedia Security VIII. LNCS, vol. 7228, pp. 42–62, Springer, Heidelberg, 2012.
[13] G. Xu and Y. Q. Shi, "Camera model identification using local binary patterns," in Proc. IEEE Int Conference onMultimedia and Expo (ICME), Melborne, Australia, pp. 392–397, 2012.
[14] A. AbdulWahab, A. Ho, and S. Li, "Inter camera model image source identification with conditional probability features," in Proc. of the 3rd Image Electronics and Visual Computing Workshop, 2012.
[15] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of resdual-based local features for camera model identification," in New Trends in Image Analysis and Processing - ICIAP Workshop: BioFor, Genoa, Italy, September 7-8, pp. 11–18, 2015.



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY