# A VIDEO CAMERA MODEL IDENTIFICATION SYSTEM USING DEEP LEARNING AND FUSION

*B. Hosler, O. Mayer, B. Bayar, X. Zhao, C. Chen, J. A. Shackleford, and M. C. Stamm*

Dept. of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA

## ABSTRACT

While significant work has been conducted to perform source camera model identification for images, little work has been done specifically for video camera model identification. This is problematic because different forensic traces may be left in digital images and videos captured by the same camera. As our experiments in this paper will show, a system trained to perform camera model identification for images yields unacceptably low performance when given video frames from the same cameras. To overcome this problem, new systems for identifying a videos source must be developed. In this paper, we propose a deep learning based system for determining the source camera model that captured a digital video. To do this, we use a convolutional neural network to produce camera model identification scores for small patches taken from video frames. These patches are chosen by a patch selection system that obtains patches from several appropriate frames temporally distributed throughout the video. Forensic information obtained by the CNN is provided to a fusion system, which combines it to produce a single, more accurate identification result. Through a series of experiments, we evaluate several system design choices and show that our system can achieve 95.9% video camera model identification accuracy.

***Index Terms***— Deep learning, convolutional neural networks, multimedia forensics, camera model identification

## 1. INTRODUCTION

Multimedia signals, such as images and videos, are used in a variety of scenarios where their origin and integrity are of critical importance. Common examples of this are criminal investigations, legal proceedings, news reporting, and strategic decision making by both governmental and business organizations. As a result, determining information about the source of multimedia signals is an important task in multimedia forensics [1]. Forensic techniques capable of identifying information about a signal's source can be used to verify a multimedia file's origin or to identify source inconsistencies within the multimedia file that could be indicative of a forgery [1, 2].

Significant research has been done to develop forensic algorithms to determine the model and manufacturer of an image's source camera. These algorithms utilize traces left by a wide variety of physical and algorithmic components in a camera's processing pipeline. Forensic camera model algorithms have been designed that leverage traces left by demosaicing [3, 4, 5, 6, 7, 8], sensor noise and other noise statistics [9, 10], JPEG header information [11], and prediction residuals [12, 8]. More recently, approaches that use convolutional neural networks to identify an image's source camera model have gained significant attention [13, 14, 15, 16]. These algorithms have led to recent advances in the detection and localization of image splicing using source camera model inconsistencies [17, 18] and open set camera model identification [19, 20].

While substantial work has been done to determine the model and manufacturer of an image's source camera, little to no work has been done to perform this same task for digital videos [2]. The majority of existing work has focused on using sensor fingerprints to identify a video's specific source device [21, 22, 23, 24, 25, 26]. This is problematic because, as we experimentally demonstrate in this paper, a camera model identification algorithm trained on images is likely unable to accurately determine the source of videos from the same set of camera models! This is likely because a camera employs different algorithmic components in its processing pipeline for images and videos (such as different compression algorithms, different demosaicing and/or post-processing algorithms that produce different image vs. video frame sizes, etc.), thus leaving behind different forensic traces. Since the amount of video content has grown dramatically, with the video sharing site Youtube alone containing over 1 billion hours of video uploads [27], it is critically important for forensic researchers to develop source camera model identification approaches that can operate on video.

When creating a video camera model identification system, researchers must confront several issues that they do not face for image-based systems. For example, do different frame types used during video coding affect forensic traces used for camera model identification? If so, how should this be taken into account when designing and training a video-based identification system? Many image-based systems are able to make reliable source model decisions on the basis of a single $M \times M$ image patch. Is this possible in videos or will the accuracy obtained using a single patch be insufficiently low? If forensic information from multiple patches is needed, where in a video should these patches be gathered from? Because of the size of digital videos, it is computationally expensive to use all patches in a video. How many patches are needed to achieve acceptable identification performance? Additionally, what is a good fusion strategy to combine information from multiple patches?

In this paper, we propose a new deep learning based system to perform source camera model identification for digital videos. Our proposed system operates by using a CNN to extract information about a video's source camera model from a set of patches extracted from different video frames. These patches are chosen by a patch selection system that obtains patches from several appropriate frames

This material is based upon work supported by the National Science Foundation under Grant No. 1553610. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Research was sponsored by the Army Research Office and the Defense Forensics Science Center and was accomplished under Cooperative Agreement Number W911NF-17-2-0087. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, the Defense Forensics Science Center, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

temporally distributed throughout the video. Forensic information obtained by the CNN is then provided to a fusion system, which combines it to produce a single, more accurate identification result.

Through a series of experiments, we determine optimal design parameters for each system component and demonstrate that our proposed video camera model identification system can achieve an average accuracy of $95.9\%$ when evaluated on a database of videos from 20 different camera models. Additionally, we experimentally demonstrate that 1) video camera model identification is more accurate when performed on I-frames than P-frames, 2) fusing unnormalized CNN class activations for each patch outperforms other strategies such as majority voting, and 3) significantly higher accuracy is achieved when patches are selected from multiple, temporally distributed frames as opposed to from within a single frame.

## 2. PROBLEM FORMULATION

In this paper, we propose a system that analyzes a digital video and determines the model and manufacturer of the camera that captured it. To accomplish this, we make use only of the video itself and the forensic traces intrinsically embedded in it by its source camera. We do not rely on metadata or any other source of extrinsic information.

For the purposes of this work, we assume videos are captured by a camera model in a set of known camera models $\mathcal{M}$. Furthermore, we assume that all videos are directly encoded by the camera using H.264, MPEG-4, or a similar video coding standard. Videos are not edited, recompressed, transcoded, or otherwise post-processed.

While little research has been dedicated to identifying the source camera model for digital videos, camera model identification is a well studied problem for images. Techniques that perform camera model identification for images make use of traces left by several components within a camera's image processing pipeline including traces left by algorithmic components such as demosaicing [4, 6, 3], JPEG compression [11], and white balancing [28], or by physical components such as a camera's sensor [9], CFA [6, 3], lens [29], etc.

Since most cameras are capable of capturing both images and videos, it may seem reasonable to identify a video's source camera model using a forensic technique developed for images and trained on images from the same set $\mathcal{M}$ of possible camera models. Unfortunately, this approach is very likely to fail, most probably due to important differences between the image and video processing pipelines within the same camera.

To demonstrate this, we conducted an experiment in which we used an image-based camera model identification system to determine the source camera model of both images and video frames captured by the same set of devices. To perform this experiment, we used the CNN-based camera model identification system proposed by Bayar and Stamm in [15]. This system was trained on 400,000 patches of size $256 \times 256$ extracted from images captured by 10 different camera models[1]. It was then used to identify the source camera model of $256 \times 256$ pixel patches taken from a separate testing set of images as well as from video I-frames and P-frames captured by the same set of camera models.

The results of this experiment are shown in Table 1. While the camera model identification accuracy obtained for image patches is high, the accuracies obtained for patches from video I- and P-frames are unacceptably low. This suggests that image-based camera model classifiers, and their accompanying feature extractors, cannot be directly transferred to video.

**Table 1**. Single patch source camera model classification accuracy for 10 camera models, with a classifier trained on image patches.

| Input Patch Type | Accuracy |
|---|---|
| Image | 98.20% |
| Video (I-frame) | 7.54% |
| Video (P-frame) | 4.16% |

The reason for this lack of transferability is likely due to differences between the image and video processing pipelines within the same camera. One important difference can be seen in the difference between the dimensions of an image and a video frame captured by the same camera. Though the same sensor is used to produce both, images from a camera are substantially larger than video frames, suggesting that sensor readings undergo different processing during the image and video frame formation process. Another important difference lies in how images and videos are encoded. While images are typically JPEG compressed, modern video coding schemes, including H.264 and MPEG-4, take advantage of temporal redundancies to reduce the size of a video for storage or transmission. This can result in different compression traces and can have different effects on other forensic traces left by the camera.

Because video coding plays a factor in our camera model identification system, we briefly review common components of modern video compression. To exploit temporal redundancy, video frames are assigned one of three different frame types; I-frames, P-frames, and B-frames. Intraframes, or I-frames, are encoded independently of any other frame using a process similar to JPEG compression.

The remaining frames are designated as either a predicted-frame (P-frame), or a bidirectional-frame (B-frame). P-frames are predicted from a previous I or P frame, known as an anchor frame. To encode a predicted frame, blocks in the anchor frame are moved to recreate an approximation of the desired frame. The movement of these blocks, and the difference between the predicted frame and current frame are both encoded so that the frame can be recovered. A similar process is used for B-frames, but with the use of previous and future anchor frames. B-frames are not available in all profiles of many video codecs, so they are not considered in this work.

## 3. PROPOSED SYSTEM

To perform video camera model identification, we propose a new deep learning based system. Our proposed system obtains local information about the source camera model from appropriate locations throughout a video, then combines this to produce a single camera model decision.

Local information is gathered by using a CNN to extract camera model information from non-overlapping $M \times M$ pixel patches taken from several frames throughout a video. This local information is then passed to a fusion system where it is combined to produce a single camera model decision $\hat{m}$. Since examining all patches throughout a video is computationally prohibitive, we employ a patch selection system to identify a small number of patches $N$ that are provided to the CNN for analysis. This system chooses $F$ frames suitable for patch extraction, the selects a $N/F$ patches from each frame that are passed to the CNN. Detailed information about each of these three subsystems (our CNN, fusion, and patch selection systems) is provided below.

### 3.1. Camera Model ID CNN

To extract local camera model information, we use a modified version of the MISLnet CNN architecture proposed by Bayar and

[1] Camera models used for this experiment were: Canon SX530, Canon SX610, Canon SL1, Fujifilm XP80, Nikon S33, Nikon S7000, Samsung S5, Samsung S7, Olympus TG-860, Panasonic FZ2000.

Stamm [30]. MISLnet is composed of a constrained convolutional layer of three filters, followed by 5 convolution blocks (convolution, batch normalization, activation, then max pooling), then two fully-connected blocks (matrix multiplication followed by activation), and finally, a fully-connected layer. Full implementation details can be found in [30].

To adapt this CNN for video camera model identification, we make several modifications to its architecture. First, we choose an input patch size of $256 \times 256$. To adapt to color patches, we remove the constraint on the first convolutional layer, and increase the number of first layer filters from 3 to 6. We then assign each neuron in the last fully connected layer to a camera model in $\mathcal{M}$.

To train our CNN, we use a softmax layer on these neurons, and minimize the cross-entropy loss between this softmax and a one-hot vector representing the true class of the training patch. To perform fusion, we remove this softmax, and use the vector of output neurons $\boldsymbol{\phi} \in \mathbb{R}^{|\mathcal{M}|}$, where $\phi_k$ corresponds to neuron value associated with the $k^{th}$ camera model. While the softmax activation is important for training, it is constrained to being non-negative, and of normalized magnitude. These properties can be limiting when comparing and fusing multiple patches. We find that $\boldsymbol{\phi}$ is a much more expressive representation in the context of fusion.

In training this CNN, we experimented with the relationship between frame compression scheme and patch-level accuracy. Specifically, we will show that training on only I-frames or only P-frames reduces the model's generality and accuracy. We find the best results when training on a balance of I-frames and P-frames. By ensuring scene diversity in our training videos, we are able to treat temporally separated frames as distinct training samples. This allows us to increase the data volume while minimizing the risk of redundancy.

### 3.2. Fusion

While the CNN described in Section 3.1 could be used to perform camera model identification on the basis of a single patch, the accuracy of single-patch decisions is not high enough to reliably classify a video. To overcome this, we propose increasing our system's camera model identification accuracy by fusing information that the CNN has extracted from multiple patches within a video.

Our fusion system accepts as inputs the unnormalized neuron activation vector $\boldsymbol{\phi}^n$, where $\boldsymbol{\phi}^n$ denotes the activation vector from the $n^{th}$ patch. It combines these activation vectors and chooses a camera model $\hat{m} \in \mathcal{M}$ according to the fusion rule

$$\hat{m} = \arg\max_{k} \sum_{n=1}^{N} f(\boldsymbol{\phi}^n, k), \tag{1}$$

where $f(\cdot)$ is a voting function that adjusts the way in which each activation vector contributes to the decision.

In this work, we consider three different voting functions. The first voting function is given by the softmax function

$$f_1(\boldsymbol{\phi}, k) = \frac{e^{\phi_k}}{\sum_\ell e^{\phi_\ell}}. \tag{2}$$

This voting scheme is such that each patch is alloted one vote in total, which can be divided across all classes. This allows each class to receive a fraction of a vote, with the condition that the sum across all classes is equal to one. Each patch is equally weighted, but patches that are indecisive spread their vote across multiple classes.

The second voting function is given by

$$f_2(\boldsymbol{\phi}, k) = \mathbb{1}(\arg\max_{\ell}(\phi_\ell) = k), \tag{3}$$

where $\mathbb{1}(\cdot)$ is the indicator function. This function allows each patch to vote for only one class, with all votes counting the same. With this scheme, fusion becomes a simple majority vote across all patches.

The third voting function we consider,

$$f_3(\boldsymbol{\phi}, k) = \phi_k, \tag{4}$$

selects the $k^{th}$ unnormalized activation from $\boldsymbol{\phi}$. This allows each patch to vote unequally for all camera models. Each patch may vote for each class, positively or negatively, with as much weight as it wants. With this voting scheme, patches can have negative weight against classes they want to reject, and are able to indicate how confident they are in one or more class with respect to other patches.

### 3.3. Patch Selection

While intuition suggests that more patches will result in a more accurate decision, and therefore a better system, the system's computational cost is improved when using fewer patches. Furthermore, to be useful to investigators, our system must be able to operate on short videos with a limited number of available patches. In light of this, our selection system uses a limited number of patches for fusion.

We observe that the accuracy of our CNN is higher when evaluating I-frames than when evaluating P-frames, so our patch selection process begins with selecting only the I-frames from a video. From these, we select $F$ frames at random. These frames are divided into non-overlapping patches, and from these patches, $P$ are chosen from each frame. Activations are extracted from these $F \times P = N$ patches, using our CNN. These activations are then fused, and a decision is made according to (1).

Our experiments suggest that activations of patches within the same frame are more correlated than those that are temporally separated. As a result, we find that for a constant $N$, accuracy increases as $F$ increases. Therefore, $F$ should be maximized, limited by the scene redundancy.

## 4. EXPERIMENTAL RESULTS

We performed a series of experiments in order to determine optimal design choices, and to evaluate our system's performance. We 1) evaluated the effect that frame type has on our feature extractor, during both training and classification, 2) compared fusion functions, and their relative effect on performance, and 3) evaluated the effects of patch quantity and frame diversity to determine an appropriate patch selection strategy.

To do this, we created a database of videos from 20 camera models. We collected over 250 videos, from each of these camera models[2]. Each video is approximately 5 seconds in duration. Videos were captured using each camera's default settings, in a variety of lighting and scene conditions. All videos were encoded by their source camera according to the H.264 or MPEG-4 video encoding standard, and no video was edited or re-encoded.

We divided the videos from each camera model into separate training sets and testing sets, resulting in 5,962 total training videos and 650 total testing videos. A database of 320,880 I-frame patches was created from the set of training videos by selecting three I-frames from each video, dividing those frames into $256 \times 256$ patches, and labeling each patch with the true camera model. This process was repeated to create a dataset of 35,280 I-frame patches

[2] Camera models used for these experiments are: iPhone 8+, Asus Zenfone 3 Laser, Canon SL1, Canon T6i, Canon SX530, Canon SX610, Fujifilm XP80, Google Pixel 1, Google Pixel 2, Huawei Honor 6x, Kodak Ektra, LG Q6, LG X Charge, Moto G5+, Nikon s33, Olympus TG-860, Samsung J7 Pro, S7, Samsung J5-6, Sony Xperia L1

**Table 2**. CNN accuracy evaluating I-frames and P-frames conditioned on training set.

| Training frame type | I | P | I+P |
|---|---|---|---|
| I-frame Accuracy | 73.5% | 42.9% | 75.3% |
| P-frame Accuracy | 54.5% | 58.9% | 70.3% |

**Table 3**. Single frame accuracy of tested fusion techniques

| Fusion function | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| Accuracy | 91.0% | 92.1% | 92.6% |

from the set of testing videos, and two more datasets comprising P-frame patches, with 320,880 and 35,280 patches from the training and testing sets respectively.

We used the following parameters when training all of our CNNs. Networks are trained for 26 epochs on batches of 40 patches with a learning rate of 0.001 halved every two epochs. Training is performed by Tensorflow on an Nvidia Gforce GTX 1080Ti GPU.

**Experiment 1:** To determine the effect that frame type has on our feature extractor, we compared the use of different frame types in both training and single-patch classification. We trained our network three times, creating three different models, for comparison. One model was trained on the dataset of I-frames described above, another on the dataset of P-frames, also described above. The third model was trained using both the I-frame and the P-frame datasets. Each model was then used to classify patches in the I-frame testing set, and patches in the P-frame testing set.

The results of this experiment are shown in Table 2, which enumerates the accuracy of each CNN on each frame type. From these results, we can see that the greatest accuracy across both frame types is achieved by the model trained using both I- and P-frame datasets. This suggests that the diversity of training frame type helps the CNN learn better forensic feature extractors. We also note that this model achieved 5% greater accuracy when classifying I-frames as compared to P-frames. This shows that to achieve higher accuracy, our system should only utilize I-frames for extracting source camera model information. Finally, we note that the highest single-patch identification accuracy that our CNN achieved was 75.3%. While our frame type analysis has helped us identify conditions that maximize our CNNs performance, these results suggests that relying on a single patch identification system alone (as is often done for digital images) is unlikely to yield strong enough performance for video.

**Experiment 2:** Next, we conducted an experiment to evaluate the performance of the different voting functions used in our fusion approach. To determine which voting scheme was best suited for video camera model identification, we performed frame-level fusion in this experiment using a variety of different voting schemes. We compared the three different fusion functions, $f_1, f_2$, and $f_3$, as described in Section 3.2.

Using the CNN trained on both I and P frames, as described in the previous experiment, activations were collected from all 28 non-overlapping patch in one randomly selected I-frame from each of the 650 videos in the testing set. These activations were fused using each candidate voting function with our fusion system. Table 3 shows the frame-level fusion accuracy using each proposed voting function.

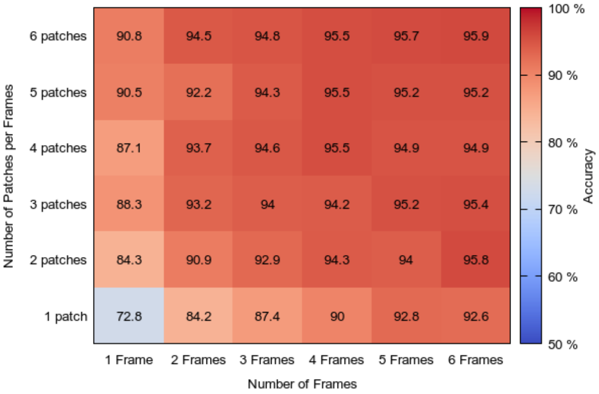

**Fig. 1**. Whole video classification accuracy when fusing $P$ patches from each of $F$ frames.

In Table 3 we see that function $f_3$, which allows each patch to vote for each class with any weight, performs frame-level fusion with 92.6% classification accuracy, and is the voting scheme which best combines the information present in the activation vectors. This suggests that forcing all patches to have the same number of votes is worse than allowing each patch to vote according to it's own relative confidence. We note that our chosen approach, $f_3$, outperforms the commonly used majority vote strategy, described by $f_2$.

**Experiment 3:** With a CNN and fusion technique fixed, we use patch selection to increase the quality of our system's video-level decision. To investigate the benefits of our patch selection process, we evaluated our system's accuracy in relation to the temporal diversity of the selected patches.

For each video in our testing database, we selected $F$ I-frames and divided them into patches. $P$ of these patches were then selected from each frame, and analyzed using our patch CNN. The activations from these patches were then fused to make a decision, using $f_3$ for $f$ in (1). Fig. 4 shows the accuracy for various $P$ and $F$ values. For example, our system correctly classified 95.8% of patches when fusing 2 patches from each of 6 frames.

Most notably, Fig. 4 shows that using a constant number of total patches and increasing the number of frames tends to result in improving accuracies. Using 6 patches from 1 frame ($P = 1,\ F = 6$) leads to 90.8% accuracy, but using 2 or more frames, while keeping the number of total patches constant, increases the classification accuracy by roughly 2%. This figure also shows that with the fusion of only 8 patches ($P = 2,\ F = 4$), we achieve over 94% accuracy. In contrast, using all 28 available non-overlapping patches from a single frame resulted in only 92.6% accuracy. These results suggest that temporally diverse information is important for performing accurate camera model identification. Over 95% accuracy is achieved using as few as 12 patches with enough temporal frame diversity.

The results of this experiment show that with proper frame and patch selection, a video's source camera model can be identified with high accuracy.

## 5. CONCLUSION

In this work, we proposed a method for performing video camera model identification. The proposed approach consists of a CNN that outputs identification scores from small patches in a video, and a strategy to fuse scores from multiple patches in an intelligent manner. We performed a series experiments that demonstrate the proposed approach achieves high accuracy (>95%) on a set of 20 camera models. Experiments also show that the choice of frame type for both training and analysis has significant impact on system performance, as well as the choice of fusion and patch selection strategies.

## 6. REFERENCES

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, 2012.

[3] A. Swaminathan, M. Wu, and K. J. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 91–106, Mar. 2007.

[4] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 899–910, Dec. 2009.

[5] X. Zhao and M. C. Stamm, "Computationally efficient demosaicing filter estimation for forensic camera model identification," in *IEEE International Conference on Image Processing (ICIP)*, Sept. 2016, pp. 151–155.

[6] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *IEEE International Conference on Image Processing 2005*, Sept. 2005, vol. 3, pp. III–69.

[7] S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro, "Demosaicing strategy identification via eigenalgorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2659–2663.

[8] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov. 2015, pp. 1–6.

[9] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *IEEE International Conference on Image Processing*, Oct. 2008, pp. 1296–1299.

[10] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 250–263, Jan. 2014.

[11] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from jpeg headers," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, Sept. 2011.

[12] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "A study of co-occurrence based local features for camera model identification," *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 4765–4781, 2017.

[13] L. Bondi, L. Baroffio, D. Gera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, Mar. 2017.

[14] B. Bayar and M. C. Stamm, "Design principles of convolutional neural networks for multimedia forensics," in *IS&T Symposium on Electronic Imaging (EI) - Media Watermarking, Security, and Forensics - Special Session on Deep Learning for Multimedia Security*, San Francisco, CA, Feb. 2017, pp. 77–86.

[15] B. Bayar and M. C. Stamm, "Augmented convolutional feature maps for robust cnn-based camera model identification," in *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, Sept. 2017, pp. 4098–4102.

[16] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2016, pp. 1–6.

[17] L. Bondi, S. Lameri, D. Guera, P. Bestagini, E. J. Delp, S. Tubaro, et al., "Tampering detection and localization through clustering of camera-based cnn features.," in *CVPR Workshops*, 2017, pp. 1855–1864.

[18] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2016, pp. 1–6.

[19] B. Bayar and M. C. Stamm, "Towards open set camera model identification using a deep learning framework," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 2018.

[20] O. Mayer and M. C. Stamm, "Learned forensic source similarity for unknown camera models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 2018.

[21] K. Kurosawa, K. Kuroki, and N. Saitoh, "Ccd fingerprint method-identification of a video camera from videotaped images," in *International Conference on Image Processing*, Oct. 1999, vol. 3, pp. 537–540.

[22] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Source digital camcorder identification using sensor photo response non-uniformity," in *Security, Steganography, and Watermarking of Multimedia Contents IX*. International Society for Optics and Photonics, 2007, vol. 6505, p. 65051G.

[23] W. Chuang, H. Su, and M. Wu, "Exploring compression effects for improved source camera identification using strongly compressed video," in *IEEE International Conference on Image Processing*, Sept. 2011, pp. 1953–1956.

[24] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution using stabilized video," in *IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.

[25] M. Goljan, M. Chen, P. Comesaña, and J. Fridrich, "Effect of compression on sensor-fingerprint based camera identification," *Electronic Imaging*, vol. 2016, no. 8, pp. 1–10, 2016.

[26] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, pp. 15, 2017.

[27] Youtube Official Blog, "You know whats cool? A billion hours," Feb. 2017, https://youtube.googleblog.com/2017/02/you-know-whats-cool-billion-hours.html.

[28] A. Swaminathan, M. Wu, and K. J. R. Liu, "Optimization of input pattern for semi non-intrusive component forensics of digital cameras," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2007, vol. 2, pp. II–225–II–228.

[29] T. Gloe, "Feature-based forensic camera model identification," in *Transactions on data hiding and multimedia security VIII*, pp. 42–62. Springer, 2012.

[30] B. Bayar and M. C. Stamm, "Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2691–2706, Nov. 2018.